PERCEPTUALLY DRIVEN METHODS FOR MULTIMODAL INTERACTIVE GRAPHICS SYSTEMS

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Computer Science)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Yunxiang Zhang

May 2026

PERCEPTUALLY DRIVEN METHODS FOR MULTIMODAL INTERACTIVE GRAPHICS SYSTEMS

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Computer Science)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Yunxiang Zhang

May 2026

Approved:



Department Chair Signature

April 27, 2026

Date

Approved by the Guidance Committee:

Major: PhD Computer Science

Qi Sun
Associate Professor
NYU Tandon School of Engineering

Date

April 23, 2026

Paul M. Torrens
Professor
NYU Tandon School of Engineering

Date

April 24, 2026

Gordon Wetzstein
Associate Professor
Stanford University

Date

April 27, 2026

Anton Kaplanyan
Vice President of Graphics Research
Intel Corporation

Date

April 27, 2026

Copies of this dissertation are obtainable from

UMI Dissertation Publishing

ProQuest CSA

789 E. Eisenhower Parkway

P.O. Box 1346

Ann Arbor, MI 48106-1346

Vita

Yunxiang Zhang was born in Chongqing, China. He received his B.S. degree in Information Engineering from Shanghai Jiao Tong University in 2017 and his Diplôme d'Ingénieur, with a specialization in Computer Science, from École Polytechnique in 2018. He subsequently earned his M.Eng. degree in Electronics and Communication Engineering from Shanghai Jiao Tong University in 2020 and his M.Phil. degree in Information Engineering from The Chinese University of Hong Kong in 2022.

In September 2022, he enrolled in the Ph.D. program in Computer Science at the NYU Tandon School of Engineering, where he conducted the research presented in this dissertation. His doctoral research lies at the intersection of computer graphics, virtual and augmented reality, human-computer interaction, and generative AI, with a focus on perceptually driven methods for multimodal interactive graphics systems. During his doctoral studies, he published 12 journal and conference papers, including five first-authored papers. He received a Best Paper Award and a Best Presentation Award at SAP 2024, as well as a Deborah Rosenthal, MD Award at NYU. He also conducted research internships and collaborations with Intel Graphics Research, Adobe Research, Meta Reality Labs, and Sony Interactive Entertainment.

His doctoral research was partially supported by the National Science Foundation under Grant Nos. 2225861 and 2232817, as well as by an Intel-sponsored research program and a Sony Focused Research Award.

List of Publications

WACV 2026 Perceptually Guided 3DGS Streaming and Rendering for Mixed Reality
Yunxiang Zhang*, *Sai Harsha Mupparaju**, *Kenneth Chen, Jenna Kang, Xinyu Zhang, Maito Omori, Kazuyuki Arimatsu, Qi Sun*

EI 2026 Perceptual Impact of Peak Luminance and Contrast in Direct View HDR Display
*Kenneth Chen**, ***Yunxiang Zhang****, *Qi Sun, Alexandre Chapiro*

SIGGRAPH 2025 Image-GS: Content-Adaptive Image Representation via 2D Gaussians
Yunxiang Zhang*, *Bingxuan Li**, *Alexandr Kuznetsov, Akshay Jindal, Stavros Diolatzis, Kenneth Chen, Anton Sochenov, Anton Kaplanyan, Qi Sun*

Optics Express 2025 Novel View Synthesis for 3D Computer-Generated Holograms Using Deep Neural Fields
Kenneth Chen, Anzhou Wen, ***Yunxiang Zhang****, Praneeth Chakravarthula, Qi Sun*

SIGGRAPH E-Tech 2025 Physically Grounded Monocular Depth via Nanophotonic Wavefront Prompting
*Bingxuan Li**, *Jiahao Wu**, *Yuan Xu**, *Zezheng Zhu,* ***Yunxiang Zhang****, Kenneth Chen, Yanqi Liang, Nanfang Yu, Qi Sun*

SAP 2024 GazeFusion: Saliency-guided Image Generation
Yunxiang Zhang*, Nan Wu, Connor Lin, Gordon Wetzstein, Qi Sun*

TVCG 2024 Measuring and Predicting Multisensory Reaction Latency: A Probabilistic Model for Visual-Auditory Integration
Xi Peng, ***Yunxiang Zhang****, Daniel Jiménez Navarro, Ana Serrano, Karol Myszkowski, Qi Sun*

ISMAR 2024 May the Force Be with You: Dexterous Finger Force-Aware VR Interface
*Fengze Zhang**, ***Yunxiang Zhang****, *Xi Peng, Sky Achitoff, Paul M. Torrens, Qi Sun*

ISMAR 2024 Toward User-Aware Interactive Virtual Agents: Generative Multi-Modal Avatar Behaviors in VR
Bhasura Gunawardhana, ***Yunxiang Zhang****, Qi Sun, Zhigang Deng*

SIGGRAPH 2024 Accelerating Saccadic Response through Spatial and Temporal Cross-Modal Misalignments
Daniel Jiménez Navarro, Xi Peng, ***Yunxiang Zhang****, Karol Myszkowski, Hans-Peter Seidel, Qi Sun, Ana Serrano*

SIGGRAPH 2023 Toward Optimized VR/AR Ergonomics: Modeling and Predicting User Neck Muscle Contraction
Yunxiang Zhang*, Kenneth Chen, Qi Sun*

SIGGRAPH Asia 2022 Force-Aware Interface via Electromyography for Natural VR/AR Interaction
Yunxiang Zhang*, Benjamin Liang, Boyuan Chen, Paul M. Torrens, S. Farokh Atashzar, Dahua Lin, Qi Sun*

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Qi Sun. Your mentorship has been instrumental in shaping my research path and intellectual growth. Thank you for teaching me to approach research holistically—to look beyond immediate technical details and consider the broader impact and significance of my work.

I am deeply grateful to Alex K., Akshay, Anton S., and Anton K. for guiding me through my first industry research experience at Intel—an experience that played an important role in shaping my interests and confidence as a researcher. I also sincerely thank Paul, Gordon, Stavros, and Kazuyuki for their insightful questions and thoughtful challenges during our collaboration. Your feedback pushed me to think more critically and helped me grow in ways I will carry forward. My heartfelt thanks go to Cuong and Long for their guidance during my time at Adobe Research—your mentorship has been invaluable. I am equally grateful to Alex C., Pat, Thomas, Yuta, and Ajit for their support and mentorship at Meta Reality Labs. Working with and learning from all of you was truly one of the most rewarding and enjoyable experiences of my Ph.D.

To the past and present members of the NYU Immersive Computing Lab, thank you for your friendship, collaboration, and the many stimulating discussions. I am especially thankful to my co-authors—Kenny, Xi, Fengze, Bingxuan, and Harsha—whose contributions were essential to the success of our work together.

To my friends Yongle, Siyu, Tian, and Xiangjiang, thank you for your companionship and encouragement—the moments we shared made this long journey much lighter.

Finally, to my parents and my wife, thank you for everything. Your unwavering love, understanding, and support have been my foundation throughout this journey. In both the best and most difficult times, you were always there for me. I could not have reached this milestone without you.

To my family, whose unwavering love and support carried me through this journey.

ABSTRACT

PERCEPTUALLY DRIVEN METHODS FOR MULTIMODAL INTERACTIVE GRAPHICS SYSTEMS

by

Yunxiang Zhang

Advisor: Prof. Qi Sun, Ph.D.

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Computer Science)

May 2026

Recent advances in sensing, computing, and rendering technologies are rapidly transforming how people interact with graphics systems. Modern interactive platforms such as extended reality (XR) devices increasingly expose rich real-time signals about their users, including eye gaze, body motion, and muscle activity, while emerging computational methods such as neural representations and diffusion models substantially expand the capabilities of visual rendering and content generation. Together, these trends create a timely opportunity to design human-in-the-loop interactive graphics systems that are not only more capable computationally, but also more

closely aligned with the perceptual and behavioral characteristics of their users, improving both system performance and user experience.

This dissertation investigates perceptually driven methods for multimodal interactive graphics systems, with an emphasis on incorporating perceptual and behavioral knowledge into modern graphics pipelines and interfaces to enable more natural, efficient, and human-centered interaction. Several closely related topics are explored, including perception-driven 2D and 3D representations for improving rendering quality and efficiency, saliency-guided content generation for attention-aware interaction design, modeling visual-auditory reaction performance for multimodal content design, force-aware interfaces for more natural, dexterous, and physically grounded interaction, and muscular workload characterization for ergonomically informed immersive experiences.

Taken together, these contributions demonstrate that human perceptual and behavioral knowledge plays a central role across the major components of multimodal interactive graphics systems, from representation and rendering to interaction and content generation. By treating such knowledge as design principles rather than solely as evaluation targets, this dissertation advances a human-centered perspective on graphics and XR systems and highlights new directions for building multimodal interactive technologies that are more efficient, intuitive, and responsive to their users.

Table of Contents

Vita iv

Acknowledgements vi

Abstract viii

List of Figures xvii

List of Tables xviii

1 Introduction **1**

2 Background and Related Work **6**

2.1 Visual Content Representation, Rendering, and Generation 6

2.2 Human Perception and Immersive Interaction 12

2.3 Human-Centered Computer Graphics 16

3 Content-Adaptive Image Representation **21**

3.1 Representing Images as 2D Gaussians 25

3.2 Image and Texture Compression Performance 29

3.3 Semantics-Aware Image Compression 42

3.4 Joint Image Compression and Restoration 47

3.5 Discussion 52

4 Perceptually Guided 3D Gaussian Splatting **54**

4.1 A Continuous Level-of-Detail Representation for 3DGS . . . 57
4.2 Mapping Perceptual Quality to 3DGS Level of Detail . . . 58
4.3 An Edge-Cloud Collaborative Rendering Framework for 3DGS . . . 60
4.4 User Study: Perceptual Quality Evaluation in Mixed Reality . . . 65
4.5 Discussion . . . 69

5 Saliency-Guided Image Generation 71
5.1 A Saliency-Guided Diffusion Model . . . 74
5.2 User Study: Evaluating Visual Attention in Generated Images . . . 80
5.3 Model-Based Saliency Analysis . . . 84
5.4 Interactive Design of Saliency Guidance . . . 87
5.5 Image Generation with Attention Suppression . . . 89
5.6 Display-Adaptive Image Generation . . . 91
5.7 Discussion . . . 93

6 Visual-Auditory Perception and Reaction 96
6.1 Pilot Study: Measuring Visual-Auditory Reaction Latency . . . 99
6.2 A Computational Model for Visual-Auditory Reaction Latency . . . 108
6.3 Model Accuracy and Generalization . . . 113
6.4 Visual-Auditory Reaction Latency in Immersive Environments . . . 115
6.5 Assessment of Visual-Auditory Perceptibility . . . 120
6.6 Layout Design for Visual-Auditory Content . . . 122
6.7 Video Playback Speed Recommendations . . . 124
6.8 Discussion . . . 126

7 Force-Aware Interfaces for XR Interaction 129
7.1 Human Hand and Forearm Biomechanics . . . 133

7.2 Joint sEMG and Force Data Collection 135
7.3 ForceSense1: An Autoencoder Model for sEMG-to-Force Decoding 138
7.4 Finger-Level Force Decoding with ForceSense1 141
7.5 User Study: Enhancing Material Perception in Virtual Environments 150
7.6 User Study: Ubiquitous Control via Finger Tapping 156
7.7 ForceSense2: A Transformer Model for sEMG-to-Force Decoding 159
7.8 User Study: Force-Driven Interaction with Virtual Objects 162
7.9 User Study: Multimodal Task Learning 166
7.10 Discussion 175

8 Toward Ergonomically Friendly XR Experiences 177

8.1 Quantifying Neck Muscle Contraction during Immersive Viewing 181
8.2 Predicting Neck Muscle Contraction from Head Movements 186
8.3 Evaluating Muscle Contraction Level Prediction Performance 195
8.4 Reducing Neck Muscle Discomfort via Content Layout Optimization 198
8.5 Discussion 203

9 Conclusion 205

References 208

List of Figures

3.1 Content-adaptive image representation with Image-GS. 23
3.2 Image-GS optimization pipeline. 28
3.3 Rate-distortion curves for image compression experiments. 31
3.4 Rate-distortion curves for image compression experiments. 31
3.5 Qualitative comparison under low-bitrate regime. 32
3.6 Qualitative comparison against conventional and neural image representations I. 33
3.7 Qualitative comparison against conventional and neural image representations II. 34
3.8 Rate-distortion curves for image compression experiments on the CLIC2020 benchmark. 35
3.9 Qualitative results on the CLIC2020 benchmark I. 36
3.10 Qualitative results on the CLIC2020 benchmark II. 37
3.11 Image-GS's rate-distortion trade-off through progressive optimization I. 39
3.12 Image-GS's rate-distortion trade-off through progressive optimization II. 40
3.13 System performance for image compression. 41
3.14 Rate-distortion curves for texture compression experiments. 41
3.15 Qualitative comparison against industry-standard GPU texture compression algorithms I. 42

3.16 Qualitative comparison against industry-standard GPU texture compression algorithms II. . . . 43
3.17 Qualitative comparison against industry-standard GPU texture compression algorithms III. . . . 44
3.18 Qualitative comparison against industry-standard GPU texture compression algorithms IV. . . . 45
3.19 Semantics-aware image compression for machine vision. . . . 46
3.20 Joint image compression and restoration I. . . . 48
3.21 Joint image compression and restoration II. . . . 49
3.22 Joint image compression and restoration III. . . . 50

4.1 Perceptually guided continuous LOD modulation across eye gaze positions and compute budget levels. . . . 59
4.2 Perceptually guided 3DGS streaming and rendering framework. . . . 61
4.3 Quantitative evaluation of different LOD modulation approaches. . . . 62
4.4 Visual comparison of different LOD modulation approaches. . . . 63
4.5 MR user study results. . . . 67
4.6 Per-fixation MR user study results. . . . 68

5.1 Attention-controllable image generation with saliency guidance. . . . 72
5.2 Saliency-guided image generation with GazeFusion. . . . 75
5.3 Qualitative results of saliency-guided image generation with GazeFusion I. . . . 76
5.4 Qualitative results of saliency-guided image generation with GazeFusion II. . . . 77
5.5 Ablation study on subtle changes in saliency guidance. . . . 78
5.6 Qualitative results of saliency-guided video generation with GazeFusion. . . . 79
5.7 Eye-tracked user study setup. . . . 81
5.8 Eye-tracked user study results. . . . 83

5.9 Interactive design of saliency guidance. 88
5.10 Attention suppression with GazeFusion. 89
5.11 Absolute attention suppression with GazeFusion. 90
5.12 Relative attention suppression with GazeFusion. 91
5.13 Display-adaptive image generation. 92

6.1 Predicting reaction latency under multisensory integration and changing eccentricity. 98
6.2 Pilot study experimental setup. 100
6.3 Pilot study stimuli and task. 101
6.4 Preliminary study: visual-auditory reaction latency results. 102
6.5 Pilot study: visual-auditory reaction latency effects. 105
6.6 Pilot study: visual-auditory reaction latency histograms. 106
6.7 3D visualization of the visual-auditory integrated DDM. 112
6.8 Cross-validation results and comparison with prior work. 114
6.9 Validation study scenes and target cues. 117
6.10 Validation study results vs model predictions. 119
6.11 Assessing target-missing probability during video playback. 121
6.12 Visual-auditory AR content layout design. 123
6.13 Video playback speed recommendations. 126

7.1 Force-aware XR interface via sEMG-based force decoding. 131
7.2 Force-driven virtual interactions with ForceSense2. 132
7.3 Illustration of forearm muscles responsible for finger flexion and extension. . . . 133
7.4 Experimental setup for synchronized sEMG and force data collection. 135
7.5 Hand–object interaction types used to construct the sEMG–force dataset. 137
7.6 Model architecture of ForceSense1. 139
7.7 Quantitative force decoding evaluation of the user-independent ForceSense1 model. 143

7.8 Quantitative force decoding evaluation of user-specific ForceSense1 models. . . . 147
7.9 Data-efficient calibration of user-specific models via transfer learning. 148
7.10 Comparison between predicted and ground-truth finger-level forces. 149
7.11 Stimuli used in the XR material perception user study. 152
7.12 XR material perception user study results. 154
7.13 Ubiquitous control via finger tapping. 158
7.14 Workflow of ForceSense2. 161
7.15 Force-driven interaction with virtual objects via ForceSense2. 164
7.16 Demonstration of multimodal task learning with ForceSense2. 168
7.17 Quantitative evaluation of multimodal task learning with ForceSense2. 171
7.18 Subjective evaluation of multimodal task learning with ForceSense2 through a post-study survey. 172
7.19 Qualitative results from the calligraphy study. 174

8.1 Predicting neck muscle discomfort level during immersive viewing. 179
8.2 Human neck muscle anatomy and pilot study experimental setup. 181
8.3 sEMG data processing to obtain normalized MCL. 183
8.4 MCL during stationary viewing. 184
8.5 ΔMCL during dynamic viewing. 185
8.6 Qualitative MCL estimation results. 189
8.7 MCLNet architecture and example MCL estimation results. 190
8.8 Predicting head motion trajectories from starting and ending head poses. 191
8.9 Our sEMG-to-MCL data processing pipeline. 193
8.10 Performance of MCLNet in estimating neck MCL from completed head motion trajectories. 196

8.11 Performance of MCLNet +TrajectoryNet in predicting neck MCL using target head poses only. 197

8.12 User study stimuli and results. 199

8.13 User study condition generation algorithm. 201

List of Tables

3.1 Ablation study on the design choices of Image-GS. 35

5.1 Eye-tracked user study results. 82

5.2 Model-based saliency analysis. 85

6.1 Varjo Aero VR headset specifications. 100

6.2 Sony WH-1000XM5 headphone specifications. 101

6.3 Generalization to unseen users. 113

6.4 Generalization to natural scenes. 118

7.1 Ablation study on the loss function and temporal smoothing. 142

7.2 Force decoding performance comparison with vision-based methods. 145

7.3 Ubiquitous control via finger tapping. 157

7.4 sEMG-to-force decoding performance comparison. 161

7.5 One-way ANOVA on the interaction method. 170

7.6 The 17 questions used in the post-study survey. 173

8.1 User study 2IFC results. 202

Chapter 1

Introduction

Interactive graphics systems are increasingly expected to do more than display visually plausible content at high speed. In immersive and intelligent media, they must decide how visual information should be represented, where limited computation should be allocated, how multimodal content should respond to user actions, and how interaction can remain intuitive and comfortable over time. These demands are especially pronounced in extended reality (XR), where users do not merely observe rendered output but continuously look, listen, move, and act within a computational loop.

This shift makes it impossible to treat users as an afterthought. A graphics system may be optimized numerically yet still fail to serve its users well if it allocates resources to perceptually unimportant regions, ignores how attention is distributed, overlooks multimodal cueing, or relies on interaction mechanisms that feel physically unnatural. What is needed, therefore, are methods that reason not only about data, models, and hardware, but also about the human users whose perception and behavior ultimately determine whether a system is effective.

Vision remains central to graphics. Visual sensitivity varies substantially across the field of view and depends on eccentricity, contrast, and spatial structure [1–3]. In immersive displays, perceived quality is shaped by viewing conditions and the non-uniform sensitivity of the human visual system rather than by pixel-wise fidelity alone [4–6]. Visual attention is likewise selective

and structured: what people notice depends on both low-level image features and higher-level semantic factors [7–9]. Together, these observations suggest that representations, rendering strategies, and generative controls should be designed with perceptual priorities in mind rather than relying solely on machine-oriented objectives.

Interactive graphics in immersive settings, however, is not purely visual. Human responses in immersive environments emerge from multisensory integration across modalities, such as vision, audition, and haptics [10–12]. In XR, users also interact through their bodies: they move, exert forces, and sustain physical effort over time. As a result, questions of dexterous interaction, tactile realism, and ergonomic efficiency become part of the graphics problem itself rather than auxiliary usability concerns [13]. Multimodal interactive graphics systems should therefore be understood as settings in which perception, action, and computation are tightly coupled.

In this dissertation, we explore perceptually driven methods for multimodal interactive graphics systems. Our central argument is that human perceptual and behavioral knowledge should serve as design principles throughout the graphics pipeline, not merely as external tools for post-hoc evaluation. Under this view, human-centered signals can inform how visual data should be represented, how rendering resources should be allocated, how generated content can be controlled, how multimodal cues should be designed, and how immersive interfaces can be made more natural, efficient, and comfortable.

We develop this argument through six technical chapters organized as a deliberate progression from visual representation, rendering, and generation to multimodal cueing, embodied interaction, and ergonomic efficiency. The first part of this progression focuses on visual content. It begins with how images and textures can be represented more adaptively, then turns to how rendered 3D content can be perceptually prioritized in immersive displays, and finally examines how generative models can be guided to produce intended visual attention. The second part expands beyond vision alone, first by modeling how visual and auditory cues jointly shape reaction performance, then by studying how force and haptics support more natural embodied interaction, and finally

by considering muscular workload and ergonomic efficiency as broader constraints in immersive graphics system design. Taken together, these chapters frame multimodal interactive graphics as a closed loop between computation and human response.

Chapter 3: Content-Adaptive Image Representation. We begin with 2D visual representation. Conventional pixel grids are convenient for storage, rasterization, and machine processing, but they do not by themselves reflect the highly non-uniform structure of visual content or the varying perceptual importance of image regions and features. In Chapter 3, we study a content-adaptive image representation based on 2D Gaussians, using flexible primitives to model structures of different scales, orientations, and colors. Within the broader dissertation, this chapter establishes an important starting point: representations are not merely neutral data containers, but design choices that can better align with downstream perceptual priorities.

Chapter 4: Perceptually Guided 3D Gaussian Splatting. We then move from 2D representation to 3D rendering. In Chapter 4, we study a perceptually guided 3D Gaussian Splatting (3DGS) framework that adaptively allocates rendering budget according to predicted visual sensitivity across the field of view. Compared with purely geometry- or compute-driven rendering strategies, this chapter makes the human connection more explicit: rendering quality should be adjusted based on where differences are most likely to matter perceptually. It therefore extends the dissertation's visual argument from representation efficiency to rendering efficiency.

Chapter 5: Saliency-Guided Image Generation. After representation and rendering, we turn to content generation. Rather than optimizing only for fidelity or spatial layout, Chapter 5 studies how generated images and videos can be controlled to produce desired viewer attention. By introducing saliency-guided generation, we connect graphics and generative AI to a more explicit model of human visual attention. This chapter is important in the dissertation arc because it moves the discussion from encoding or displaying content efficiently to actively shaping content

in ways that better align with human viewing behaviors and application design intent.

Chapter 6: Visual-Auditory Perception and Reaction. The next step expands from vision-only considerations to multimodal perception. In Chapter 6, we study visual-auditory integration through a probabilistic model of human reaction latency in immersive environments. We examine how quickly users respond to combined visual-auditory stimuli and how those response patterns can inform applications such as multimodal content design and interface cueing. Within the dissertation, this chapter broadens the scope from visual quality and attention to human multimodal behavioral performance, showing that human-centered graphics design must also reason about when and how people react.

Chapter 7: Force-Aware Interfaces for XR Interaction. We then move from passive viewing and listening to active embodied interaction. In Chapter 7, we study force-aware XR haptic interfaces centered on how finger force can be inferred from muscle activity and incorporated into immersive interaction loops. This chapter adds haptics and physical action to the dissertation narrative by showing how graphics systems can more faithfully capture user intent and support dexterous, physically grounded interaction without relying on cumbersome on-hand hardware. It also strengthens the multimodal perspective of the dissertation by connecting visual, auditory, tactile, and motor aspects of XR experience.

Chapter 8: Toward Ergonomically Friendly XR Experiences. The final technical chapter considers a broader yet closely related human-centered constraint: physical workload and ergonomic efficiency. In Chapter 8, we model and predict neck muscle contraction during immersive viewing, enabling immersive systems to reason about user comfort before or during interaction. This chapter concludes the technical arc by extending the dissertation beyond perceptual fidelity or interaction expressiveness toward sustainable, comfortable, and ergonomically informed immersive computing. In doing so, it highlights that human-centered graphics systems must account

not only for what users perceive and do, but also for the physical effort required to participate in these experiences over time.

The remainder of this dissertation is organized as follows. Chapter 2 reviews the background literature in visual content representation, rendering, and generation; human perception and immersive interaction; and human-centered computer graphics. Chapters 3 to 8 present the six technical contributions in the order described above, progressing from visual representation, rendering, and generation to multimodal cueing, haptic interaction, and ergonomic modeling. Chapter 9 concludes the dissertation by synthesizing the shared lessons of these works and discussing future directions for multimodal interactive graphics systems that are increasingly adaptive, perceptually informed, and responsive to user needs.

Chapter 2

Background and Related Work

This chapter reviews the technical background behind the dissertation, focusing on three connected themes that recur throughout the chapters that follow. The first concerns how visual content is represented, rendered, and generated, ranging from classical image and scene representations to modern neural models and Gaussian-based methods. The second concerns how humans perceive and interact with immersive media, with emphasis on visual perception, visual attention, multisensory integration, and force-aware interaction. The third concerns human-centered computer graphics, namely methods that explicitly incorporate perceptual priorities, attentional structure, and ergonomic constraints into graphics system design. Together, these themes establish the conceptual bridge between computational graphics methods and human perceptual and behavioral processes that this dissertation seeks to model and leverage.

2.1 Visual Content Representation, Rendering, and Generation

The first group of background topics concerns how visual content is encoded, displayed, and synthesized. This line of work spans 2D images and textures, 3D scene representations, and

modern generative models for image synthesis. Across these topics, a common question emerges: which representation best supports the downstream goals of fidelity, efficiency, controllability, and real-time interaction? This question is central to the dissertation because Chapters 3 to 5 revisit it from different perspectives, emphasizing adaptive 2D representation, perceptually guided 3D rendering, and attention-aware content generation, respectively.

2.1.1 Image and Texture Representation

Images and textures are among the most fundamental data types in computer graphics. Classical image formats were primarily designed for storage and transmission efficiency. Lossless compression methods typically exploit statistical redundancy through dictionary coding and entropy coding, as in Lempel-Ziv-Welch (LZW) approaches [14]. Lossy formats, by contrast, trade exact recovery for higher compression ratios by transforming image blocks into frequency domains and quantizing coefficients, with JPEG remaining a canonical example [15]. In practice, these formats are highly effective for photographs and general visual media distribution, but they are not always ideal for real-time graphics workloads, where random access, decoding latency, and support for multi-channel material maps are often as important as bitrate.

Textures introduce a related but distinct set of requirements. In rendering systems, textures are frequently sampled in parallel on GPUs and are often accessed locally rather than decoded in full. This has motivated dedicated texture compression families, such as the Block Compression (BC) series, which are designed for fixed-rate compression, random access, and hardware-friendly decoding [16, 17]. These representations are highly practical for interactive rendering because they reduce bandwidth and memory pressure while maintaining predictable decoding costs. At the same time, their reliance on independent blocks and fixed encoding structures limits their ability to adapt to complex or highly non-uniform content, particularly when different visual channels or material attributes exhibit very different spatial statistics.

Recent work in learned image compression and neural image representation has opened a different design space. Instead of hand-designing transforms and codebooks, learned compression methods train models end-to-end so that latent representations are optimized jointly for reconstruction quality and bitrate [18, 19]. A related line of work studies per-image implicit representations, in which a neural network directly models the continuous mapping from spatial coordinates to color values [20, 21]. These methods are attractive because they can represent images continuously, avoid explicit pixel grids, and allocate representational capacity through learned parameters rather than fixed sampling layouts. However, many such models incur substantial decoding cost or require lengthy optimization, which can limit their use in real-time graphics systems.

To address these issues, hybrid approaches combine learned decoders with explicit data structures. Multi-resolution grids and hash encodings improve scalability and speed by caching local features while preserving some of the flexibility of continuous representations [22]. From the standpoint of graphics, this is an important development: the representation no longer has to be strictly grid-based or strictly neural, but can instead combine explicit and learned components to achieve favorable trade-offs among compression ratio, visual fidelity, and random-access performance.

Gaussian-based image representations add another perspective. Their recent popularity in graphics largely follows the success of 3D Gaussian Splatting (3DGS) for scene rendering [23], but the underlying idea is broader: Gaussian primitives offer continuous, anisotropic building blocks that can vary naturally in position, scale, orientation, and appearance. In image domains, this makes them an appealing alternative to rigid pixel blocks or fixed-resolution grids. Because image content is often spatially non-uniform, with sharp edges, smooth regions, oriented structures, and varying semantic salience, flexible primitives can in principle encode different regions with different effective support sizes. This broader representational viewpoint motivates the dissertation's treatment of 2D Gaussian-based image representation in Chapter 3: image and texture

representations are not only storage formats, but also computational models whose structure affects efficiency, fidelity, and downstream perceptual alignment.

2.1.2 3D Scene Representation and Rendering

3D graphics has long relied on a diverse set of scene representations, each with different trade-offs in expressiveness, efficiency, and rendering compatibility. Polygon meshes, especially triangle meshes, remain the dominant representation in real-time graphics because they align naturally with rasterization pipelines and are supported by mature hardware and tooling [17, 24]. Point clouds represent geometry as discrete samples in space and are often attractive for scanned data and reconstruction pipelines, though they require additional structure or splatting strategies to support high-quality rendering. Voxel grids discretize space volumetrically and make occupancy, density, and attribute queries simple, but they can become memory-intensive at high resolution. Parametric surfaces, including Bézier and Non-Uniform Rational Basis Spline (NURBS) formulations, are especially important in computer-aided design because they support compact descriptions and precise control of smooth surfaces. Implicit surfaces instead define geometry as the zero set of a function, offering a flexible way to represent smooth topology and complex shape families. Classical rendering algorithms, including rasterization and ray tracing, were developed largely around these families of explicit or queryable scene representations [17, 24].

Although these traditional representations remain highly effective, they often require substantial human authoring, meshing, simplification, or precomputation to achieve high visual fidelity in complex scenes. Neural scene representations emerged in part to reduce this burden. A foundational step in this direction was Neural Radiance Fields (NeRF), which represent a scene through a neural function that maps spatial location and viewing direction to density and radiance [25]. NeRF demonstrated that novel-view synthesis with high photorealism could be achieved without explicitly constructing conventional meshes or material models. This result had a profound impact

on graphics and vision because it suggested that scene representation, geometry recovery, and appearance modeling could be learned jointly from images rather than assembled through separate stages.

Subsequent work focused on improving NeRF along two major axes: quality and efficiency. Quality-oriented variants addressed aliasing, view coverage, and unbounded scenes [26, 27]. Efficiency-oriented methods reduced the cost of neural rendering by introducing explicit acceleration structures, tensor decompositions, or compact encodings [22, 28, 29]. Even with these advances, however, NeRF-style volume rendering remains computationally expensive because it depends on repeated sampling and accumulation along rays. This makes high-resolution, low-latency interactive rendering difficult, especially in settings such as mixed reality where stereo rendering, high refresh rates, and limited power budgets are all critical concerns.

3D Gaussian Splatting (3DGS) addresses this limitation by replacing expensive volumetric ray marching with an explicit set of anisotropic 3D Gaussians rendered through GPU-friendly rasterization [23]. This shift from implicit field evaluation to explicit Gaussian primitives is important for two reasons. First, it preserves many of the advantages of learned scene reconstruction while substantially improving rendering speed. Second, it makes level-of-detail control, pruning, and structured resource allocation more tractable, which is essential for interactive and immersive applications. Follow-up work has therefore extended 3DGS in several directions, including hierarchical representations and other forms of efficiency-aware organization [30].

From the dissertation's perspective, the broader significance of this line of work is that 3D scene representation is no longer only a question of geometric fidelity or reconstruction accuracy. It is also a question of how a representation interacts with rendering cost, display constraints, and user perception. This motivates the transition from classical rendering concerns to the perceptually guided rendering strategies discussed later in this chapter and developed more directly in Chapter 4.

2.1.3 Generative Modeling and Image Generation

Generative modeling addresses a different but increasingly important problem: not how to store or render an existing scene, but how to synthesize new visual content. Early deep generative models established two especially influential paradigms. Variational Autoencoders (VAEs) introduced a probabilistic latent-variable formulation in which an encoder and decoder are jointly trained to reconstruct data while regularizing the latent space [31]. Generative Adversarial Networks (GANs) instead frame learning as a competition between a generator and a discriminator, enabling sharp and realistic image synthesis through adversarial training [32]. These two families shaped much of the early modern literature on image generation: VAEs emphasized latent structure and probabilistic modeling, whereas GANs prioritized realism and high-frequency detail.

Despite their success, both paradigms have practical limitations. VAEs often suffer from blurry reconstructions due to the combination of pixel-level losses and variational regularization, while GANs can be unstable to train and provide limited control over the generated output. Diffusion models have recently become the dominant paradigm in image generation because they offer both high-quality synthesis and a flexible conditioning framework. Denoising Diffusion Probabilistic Models (DDPMs) formalized the core framework of gradually corrupting data with noise and learning a reverse denoising process [33]. Later work showed that diffusion models can be made scalable and practical for text-conditioned synthesis, with latent diffusion models providing a particularly important step by performing denoising in a compressed latent space rather than directly in pixels [34].

The success of diffusion models has shifted the focus of image generation research from unconditional sampling toward controllable generation. Users increasingly expect generation systems to respond not only to text prompts, but also to structural, semantic, and stylistic constraints. Lightweight adaptation modules and control networks have therefore become central mechanisms for introducing user control. Examples include low-rank adaptation for efficient

model customization [35], grounded generation methods such as GLIGEN [36], and spatially conditioned diffusion frameworks such as ControlNet [37] and Uni-ControlNet [38]. These methods substantially improve controllability, but most of them focus on layout, pose, depth, segmentation, or reference appearance. In other words, they mainly control what is generated and where it appears.

This shift toward controllability is especially important for interactive graphics. A generation model used in a creative workflow is rarely judged only by sample realism; it is also judged by how predictably it responds to user intent. Text prompts provide high-level semantic guidance, but they are often insufficient when users want explicit control over composition, structure, or downstream perceptual effects. As a result, current research increasingly treats image generation as a conditional interface problem rather than as a pure sampling problem. This broader framing is essential for the dissertation because it clarifies why attention-aware generation is not an isolated application detail, but part of a larger effort to align generative models with how people specify, interpret, and consume visual content.

For this dissertation, an important remaining gap is that generation quality and controllability are still often discussed without explicit reference to how human viewers distribute their attention. Spatial layout and semantic content certainly influence attention, but they do not fully determine it. Visual saliency depends on a more complex interaction among low-level image features, local structure, and high-level semantics. This observation motivates the attention-aware perspective developed later in this chapter and in Chapter 5, where content generation is treated not only as a semantic synthesis problem but also as a human-centered perceptual control problem.

2.2 Human Perception and Immersive Interaction

The second background theme concerns the human side of the graphics interaction loop. With the first section addressing how visual content can be efficiently represented, rendered, and generated,

this section addresses how that content is perceived and acted upon by human users. Two topics are particularly important for the dissertation. The first is human visual perception, including both low-level characteristics of the visual system and higher-level mechanisms of visual attention. The second is multimodal perception and interaction, especially visual-auditory integration and force-aware haptic interaction. These topics are central to the dissertation because they provide the perceptual and behavioral foundations for the design principles developed in Chapters 3 to 7.

2.2.1 Human Visual Perception

Human visual perception is shaped by strong spatial and functional non-uniformities. The fovea supports high acuity and detailed color vision, whereas peripheral vision exhibits lower spatial resolution and contrast sensitivity [1–3]. These differences are not incidental; they are fundamental to how visual information is encoded and processed. As eccentricity increases, fine detail becomes harder to resolve, color discrimination weakens, and sensitivity to different spatial frequencies changes. For graphics, this means that visually equivalent errors are not perceived equally across an image or display. A distortion or reduction in fidelity that is highly visible near the fovea may be barely noticeable in the periphery, whereas other changes may become salient because of local contrast or scene structure. This basic fact underlies many later ideas in perceptual rendering and display optimization.

The visual system is also selective in a cognitive sense. Not all visible information is attended equally, and not all attended information is determined purely by low-level stimulus properties. Research on visual attention has long distinguished between bottom-up factors, such as contrast or local feature contrast, and top-down factors, such as task goals, semantics, memory, and prior knowledge [7]. Early computational saliency models often emphasized bottom-up feature maps and center-surround differences [39, 40]. These models were influential because they showed that gaze behavior could be predicted, at least partially, from image statistics. However, they also made

clear that saliency is not reducible to low-level cues alone.

To better understand actual viewing behavior, the field increasingly turned to eye-tracked datasets and data-driven prediction. Datasets such as MIT1003 and SALICON for images, as well as later video and panoramic saliency datasets, enabled saliency prediction to be formulated as a supervised learning problem [9, 41–43]. With these data resources, saliency models evolved from hand-crafted feature pipelines to deep neural predictors that learn from large-scale gaze data [44]. This transition mirrors a broader trend in graphics and vision: rather than relying only on manually designed heuristics, perceptual models can increasingly be learned from human data while still retaining interpretable ties to visual behavior.

For this dissertation, two aspects of visual perception are especially important. First, non-uniform visual sensitivity provides the foundation for perceptually guided allocation of representational and rendering resources. Second, visual attention provides a bridge from low-level visibility to higher-level viewing behavior, which is crucial when discussing generated content and design intent. These aspects are closely related but not identical. Visibility concerns what can be perceived, whereas attention concerns what is likely to be selected, fixated, or behaviorally prioritized. Much of the dissertation builds on the distinction between these two ideas: some chapters primarily address perceived quality, while others address attentional structure and visual guidance.

2.2.2 Multimodal Perception and Interaction

Immersive interactive systems are inherently multisensory. Even when visual information dominates an experience, users often integrate it with sound, force, and proprioceptive feedback to interpret events and guide action. A large body of perceptual research has shown that multisensory integration is not simply additive. Depending on reliability, spatial alignment, and timing, one modality may reinforce, bias, or even dominate another [11, 12]. In audiovisual settings, for

example, reaction behavior depends not only on the presence of both cues but also on how their spatial and temporal properties interact [10]. This is particularly relevant to XR, where wide field of view, spatialized audio, and attention shifts together influence how quickly users notice and respond to events.

From a systems perspective, multimodal interaction also involves a problem of bidirectionality. Visual rendering is often a one-way channel from system to user, but haptic and force-based interaction are fundamentally two-way. Users exert forces on objects and devices while simultaneously receiving tactile or kinesthetic consequences. This makes force a particularly important modality for human-computer interaction and immersive embodiment. A substantial line of research has therefore focused on haptic rendering, including tactile wearables, ultrasound-based mid-air haptics, and other interfaces that produce sensations of touch or material response [45–48]. These methods demonstrate that the sense of physicality can be strengthened even when the virtual object itself has no physical substance.

At the same time, naturally sensing user-exerted force remains difficult. Wearable tactile gloves and force-sensing surfaces can provide precise measurements, but they are often obtrusive and may interfere with natural hand motion, particularly in tasks requiring dexterity [49, 50]. This limitation has motivated interest in physiological interfaces that infer force or hand state from muscle signals rather than directly instrumenting the hand. Surface electromyography (sEMG) has emerged as one of the most promising approaches in this regard. It can be used for gesture recognition, pose tracking, and increasingly for continuous estimation of hand-related quantities [51–53]. In XR, the appeal of sEMG is especially clear: it can capture aspects of user intention and force without fully relying on optical tracking, and without the same degree of interference introduced by bulky on-hand devices.

Taken together, this literature highlights a useful distinction between multimodal output and multimodal interaction. Many XR systems already combine vision and sound, or provide haptic feedback through wearable devices, but fewer systems model how multiple modalities

jointly shape task performance or how user-generated physical signals can become reliable inputs to the graphics loop itself. The dissertation adopts this broader view. Some chapters focus on how multiple sensory cues jointly influence perception and reaction, while others focus on how physiological or motor signals can be sensed to support richer forms of embodied control. In both cases, the emphasis is on closing the gap between what the system computes and what users actually sense and do.

The dissertation draws on this multimodal background in two ways. Chapter 6 focuses on how visual and auditory cues jointly influence behavior, emphasizing perceptual timing and response. Chapter 7 instead focuses on how tactile and motor information can be sensed and incorporated into immersive interaction loops. In both cases, the central theme is the same: human-centered graphics systems should not assume that perception and action are purely visual, nor that interaction quality can be understood without modeling how multiple sensory channels contribute to experience and performance.

2.3 Human-Centered Computer Graphics

The final background theme concerns methods that explicitly incorporate human factors into interactive graphics system design. The previous sections described the computational side of representation, rendering, and generation, as well as the perceptual and behavioral characteristics of human users. This section brings those threads together by reviewing three families of human-centered graphics research that connect most directly to the dissertation: perceptually guided rendering, attention-driven graphics, and ergonomically friendly interaction. These topics illustrate how knowledge about human users can become a valuable active design signal rather than solely a passive evaluation criterion.

2.3.1 Perceptually Guided Rendering

Perceptually guided rendering begins from a simple observation: the human visual system is not equally sensitive to all errors, all locations in the visual field, or all image structures. If rendering resources can be allocated according to what users are likely to notice, then substantial savings in computation or bandwidth may be achieved without a corresponding loss in perceived quality. Foveated rendering is the canonical example of this idea. Because acuity is highest near the point of gaze and declines rapidly with eccentricity, rendering quality can be concentrated in the foveal region while peripheral regions are rendered more coarsely [5, 54]. Subsequent work refined this idea by incorporating learned priors over natural video statistics and perceptually informed degradation models [6, 55, 56].

More recent perceptual models have shown that eccentricity alone is not the full story. Visibility also depends on content-aware factors such as local contrast, luminance, spatial frequency, and motion [4, 57]. This means that two regions at the same eccentricity may not deserve the same rendering budget if their perceptual importance differs. Such observations have encouraged a shift from simple eccentricity-based heuristics toward richer perceptual metrics and predictive models.

These ideas have increasingly been adapted to neural rendering. Foveated NeRF methods, for example, reduce rendering effort by lowering fidelity in peripheral regions while preserving central detail [58]. Similar ideas are now being explored in 3DGS-based rendering, where the explicit Gaussian representation makes level-of-detail control more natural [59]. The dissertation builds on this broader literature in Chapter 4, but with a more explicit emphasis on continuous, content-aware perceptual guidance rather than fixed concentric zones alone. In that sense, perceptually guided rendering is one of the clearest precedents for the dissertation's central thesis that human perceptual structure should directly inform computational allocation.

2.3.2 Attention-Driven Computer Graphics

Where perceptually guided rendering typically focuses on visibility and efficient resource allocation, attention-driven graphics focuses on how visual content can be edited, prioritized, or generated in ways that influence what viewers are likely to notice. This line of work relies on the fact that attentional structure can be estimated computationally, either from hand-crafted saliency models or from learned predictors trained on gaze data. Once such estimates are available, they can be incorporated into graphics pipelines to guide editing, optimization, or generation.

Prior work has used saliency to enhance existing images by identifying and suppressing distracting regions, preserving fidelity where attention is expected to fall while reducing or modifying less important content [60–63]. Related ideas have appeared in resource prioritization, where salient image regions are loaded or processed preferentially, and in animation or display adaptation, where predicted attention is used to guide what should be emphasized [64]. The key idea across these efforts is that attention is not merely a descriptive property of viewing behavior; it can be used as a control signal for graphics and imaging decisions.

This literature also illustrates an important conceptual distinction. Perceptually guided rendering is usually concerned with minimizing visible error under resource constraints, whereas attention-driven graphics is often concerned with shaping what the user notices, remembers, or prioritizes. The former is closely tied to visibility thresholds and quality metrics; the latter is more closely tied to communication, salience, and task structure. In practice, these two perspectives overlap, but they are not identical. A region may be highly visible without being especially attention-grabbing, and a visually subtle change may strongly alter where a viewer looks. This distinction is especially relevant for the dissertation because attention-aware generation belongs more naturally to the latter category.

For the dissertation, this literature is important because it provides the conceptual bridge from attention modeling to attention-aware generation. Most prior attention-driven approaches operate

on existing content, editing or optimizing it after the fact. By contrast, Chapter 5 asks whether attention can be incorporated directly into the synthesis process so that generated content is aligned with intended viewing patterns from the outset. This shift from post-hoc optimization to generative control is a natural extension of attention-driven graphics, and it further illustrates the broader dissertation theme of treating human perceptual behavior as a design input rather than only an evaluation outcome.

2.3.3 Ergonomically Friendly Interaction

Human-centered graphics systems must account not only for what users perceive and how they interact, but also for the physical cost of participation. This is especially true in XR, where head-mounted displays (HMDs), sustained postures, and repeated head or hand movements can introduce muscular workload and discomfort over time. Prior ergonomic studies have shown that head-supported mass, balance, and posture can significantly affect neck muscle activation and user comfort [65–67]. These findings are important because they reveal that immersive system quality cannot be measured solely by realism, responsiveness, or task performance. A visually compelling system may still be undesirable if it imposes excessive physical burden on the user.

Muscle contraction and fatigue are therefore critical background concepts. In biomechanics and rehabilitation, muscle activity is often studied as an indicator of exertion, fatigue, or injury risk [68]. In practice, electromyography has become one of the most common tools for measuring muscle activation because it provides a non-invasive window into neuromuscular activity. This has made sEMG especially useful for studying interaction behaviors that are difficult to infer from motion alone. In graphics and human-computer interaction, this is a notable point: optical tracking can reveal what movement occurred, but not necessarily how much internal effort that movement required.

This distinction motivates ergonomics-aware interaction methods that explicitly optimize for

physical effort. Work in content and path optimization has shown that virtual content can be designed in ways that reduce exertion rather than only improving geometric or perceptual criteria [69]. In XR, this perspective suggests that interface layout, target placement, and interactive tasks can all be shaped to reduce muscular burden or support more sustainable use. Chapter 8 follows this line of thinking by treating neck muscle contraction not merely as a medical or usability measurement, but as a predictive quantity that can be modeled and used computationally to guide immersive interaction design.

This viewpoint expands the notion of human-centered graphics beyond visibility and usability alone. A perceptually optimized system may still be problematic if it systematically encourages uncomfortable postures, excessive head movement, or prolonged muscular loading. Conversely, ergonomics-aware interaction treats the body as part of the design space: display form factor, task layout, target placement, and temporal interaction patterns can all influence physical comfort. By bringing these concerns into the graphics loop, ergonomics-aware methods help connect immersive system design with longer-term comfort, accessibility, and sustainable everyday use.

Summary. Taken together, the literature reviewed in this chapter motivates the dissertation's overarching perspective. Visual representations and rendering methods are increasingly flexible and adaptive; perceptual and attentional processes are increasingly quantifiable and interpretable; and human-centered design criteria such as multimodal perception, force-aware interaction, and ergonomic efficiency are increasingly relevant to immersive graphics systems. The chapters that follow build on these foundations to argue that interactive graphics systems should be designed not only for computational efficiency or visual fidelity in isolation, but also for effective alignment with human perception, behavior, and embodied experience.

Chapter 3

Content-Adaptive Image Representation

Recent advances in generative AI have dramatically increased the availability of high-resolution visual content in domains ranging from gaming to graphic design [70, 71]. Effectively deploying these assets, particularly to resource-constrained devices, requires representations that are not only compact but also efficient to decode, easy to sample in parallel, and flexible enough to adapt to different runtime budgets. Representing images as dense pixel grids has been successful for storage, transmission, and display, yet modern graphics workflows increasingly demand representations that do more than preserve raw pixel values. High-resolution images and texture stacks must often be streamed to constrained hardware, decoded with low latency, sampled randomly on GPUs, and adapted smoothly to different quality targets. These requirements expose a mismatch between conventional formats optimized primarily for compression and the broader computational demands of interactive graphics pipelines [15–17]. Even when classical codecs achieve strong compression ratios, they are rarely designed around the kinds of random access, progressive refinement, and per-region quality control that real-time graphics systems often need. Put differently, the image is no longer just a file to store and retrieve: it is a computational asset that must be queried, adapted, and rendered repeatedly inside a larger graphics system.

This mismatch is visible in both conventional image codecs and texture compression schemes.

Formats such as JPEG and PNG exploit spatial redundancy well, while block-based texture compressors such as BC1–BC7 are tailored to GPU-friendly decoding and compact memory footprints. Yet both families still expose only limited structural control over where representational capacity is spent. That limitation matters when the same asset must be decoded at different resolutions, accessed non-uniformly, or reused across multiple downstream graphics tasks. In practice, a representation that is merely compressed is often not enough; the representation should also be explicit, fast to query, and flexible enough to support later adaptation. Traditional image formats are therefore often inadequate for graphics use cases that demand both compression efficiency and direct computational accessibility [72].

This chapter studies a different representational perspective: images as collections of explicit, anisotropic 2D Gaussian primitives. The motivation is twofold. First, image content is spatially non-uniform. Sharp edges, smooth regions, repeated patterns, and semantically important structures are not distributed evenly, so allocating the same representational support everywhere is often inefficient. Second, modern graphics systems benefit from representations that remain explicit and hardware-friendly while still being adaptive. Neural image representations have emerged as a promising alternative for encoding visual data [21, 73]. When integrated into learning-based pipelines, they can improve visual fidelity and memory efficiency compared with classical formats. However, many existing methods rely on fixed data structures that lack content adaptivity [74, 75] or on compute-intensive implicit neural models [20, 76], leading to poor scalability and slow decoding. These drawbacks hinder their usage in real-time graphics, where fast random access and dynamic quality adaptation to device capabilities are critical [17]. In this chapter, we pursue a representation that preserves the local structure and interpretability of explicit primitives while still allowing the allocation of capacity to follow content. The resulting model is simple enough to decode efficiently, yet expressive enough to concentrate detail where gradients, edges, and semantic structures indicate that visual support is most needed.

The resulting representation, Image-GS, uses colored 2D Gaussian primitives to encode images



Figure 3.1: *Content-adaptive image representation with Image-GS.* Image-GS reconstructs an image by adaptively allocating and progressively optimizing a set of colored 2D Gaussians. It achieves favorable rate-distortion trade-offs, hardware-friendly random access, and flexible quality control through a smooth level-of-detail stack. A 20% randomly sampled subset of Gaussians is shown to illustrate the optimized spatial distribution.

and multi-channel texture stacks. Each primitive is parameterized by a mean, a covariance, and a color vector, so Image-GS can adapt to local structure without relying on a fixed block layout or a heavy implicit decoder. Given a target image, Image-GS initializes Gaussians adaptively using local image-gradient magnitudes, allocating more support to high-frequency regions, and then optimizes these parameters with a custom differentiable renderer. Additional Gaussians are progressively added to regions with persistent residual errors so that capacity continues to follow content during optimization. Because the primitive basis is explicit, it also supports random access, progressive refinement, and smooth quality control in a way that many latent image models do not. This adaptive spatial behavior is illustrated in Figure 3.1, where the optimized primitives concentrate around structured regions instead of being distributed uniformly across the image. That same content-aware allocation is central to the dissertation-wide theme of perceptually and behaviorally informed graphics: visual content is not uniform, and neither should the computational support

that represents it. In that sense, Image-GS is not only a compression method, but also a concrete example of how representational form can be aligned with content structure and downstream task requirements.

This chapter begins by formalizing images as optimized sets of 2D Gaussians and describing the rendering and optimization procedures used to fit them. It then evaluates the resulting representation on both image and texture compression, focusing on rate-distortion behavior, system performance, and progressive level-of-detail (LOD)control. We compare Image-GS against recent neural image representations and industry-standard texture compressors across diverse images and textures, showing favorable trade-offs between visual fidelity and memory or computation cost, especially for graphics assets with non-uniformly distributed features and for low-bitrate settings. The last two sections extend Image-GS to two applications that further illustrate its usefulness beyond pure reconstruction: semantics-aware image compression for machine vision and joint image compression and restoration. In the first application, Image-GS is biased toward semantically important regions so that downstream machine-vision tasks can remain robust under strong compression. In the second, the same representation is used as a joint coding and restoration prior, showing that explicit Gaussian primitives can support not only compression but also image recovery. Taken together, these results show that content-adaptive Gaussian primitives provide a practical and expressive alternative to both conventional image formats and fixed-structure neural representations, especially when the representation must remain useful after compression rather than merely visually acceptable.

This chapter's main contributions are a content-adaptive image representation with hardware-friendly random access and flexible rate-distortion trade-offs, a differentiable renderer that makes the representation practical at high resolution, and two downstream applications that show how the same representation can be reused for semantics-aware compression and restoration. These components make Image-GS a practical image representation for graphics pipelines. Our source code and evaluation dataset are released at `https://github.com/NYU-ICL/image-gs`.

3.1 Representing Images as 2D Gaussians

This section introduces the representation, rendering formulation, and optimization procedure used throughout the chapter. The core idea is to reconstruct a target image using a set of explicit anisotropic 2D Gaussians whose spatial distribution adapts to local image content. In contrast to view-dependent 3D Gaussian rendering, the 2D setting does not require depth ordering or opacity accumulation. This simplification makes it possible to retain the expressive geometry of Gaussian primitives while designing a rendering pipeline that is better aligned with image-domain reconstruction, fast random access, and later quality adaptation. Figure 3.2 summarizes the full optimization loop, from content-aware initialization through differentiable rendering and residual-driven primitive insertion.

3.1.1 2D Gaussian Parameterization

Similar to the Gaussian primitives in 3D Gaussian Splatting [23], a 2D Gaussian's geometry is defined by a mean vector $\mu_i \in \mathbb{R}^2$ and a covariance matrix $\Sigma_i \in \mathbb{R}^{2\times2}$, and its value evaluated at an arbitrary pixel location $x \in \mathbb{R}^2$ can be expressed as:

$$G_i(x) = \exp\left(-\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right), \tag{3.1}$$

To ensure that Σ_i remains positive semi-definite during numerical optimization, we instead work with its factorized form that consists of a rotation matrix $R(\theta_i)$ and a scaling matrix $S(s_i)$:

$$\Sigma_i = R(\theta_i)S(s_i)S(s_i)^TR(\theta_i)^T, \tag{3.2}$$

Specifically, we establish and maintain a rotation angle $\theta_i \in [0, \pi]$ and a scaling vector $s_i \in \mathbb{R}^2_+$ for each 2D Gaussian. These attributes are updated via stochastic gradient descent and clipped to the valid ranges during training. The rotation and scaling matrices $R(\theta_i)$ and $S(s_i)$ are constructed

from θ_i and s_i on the fly during both training and inference.

Unlike 3D Gaussian Splatting, which models view-dependent appearance using spherical harmonics [23], we only need to associate each 2D Gaussian with a vector $c_i \in \mathbb{R}^d$ to store its color, as an image essentially shows a single view of a 3D scene. Notably, the design choice of a variable color dimension d enables Image-GS to flexibly support a diversity of image formats, including grayscale, RGB, and CMYK images, as well as multi-channel texture stacks.

In addition, 3D Gaussian Splatting relies on an opacity attribute for depth-ordered occlusion and α-blended rendering. By contrast, the color information of 2D Gaussians can be effectively aggregated regardless of their relative order, as detailed in Section 3.1.2. Therefore, our 2D Gaussians do not have an opacity attribute.

By combining the above components, each 2D Gaussian primitive in Image-GS is fully characterized by $5 + d$ trainable parameters:

$$p_i \coloneqq (\mu_i, \theta_i, s_i, c_i) \in \mathbb{R}^{5+d}, \quad 1 \le i \le N_g \tag{3.3}$$

3.1.2 Rendering 2D Gaussians into Images

Given a set of Gaussians, a naive rendering strategy would be summing up every Gaussian contribution at every pixel:

$$\hat{I}(x) = \sum_{i=1}^{N_g} G_i(x)\, c_i, \tag{3.4}$$

Although simple, this dense formulation is unnecessarily expensive and undermines data locality. It can be viewed as a 2D simplification of point-based α-blending [77, 78], except that order does not matter in our setting because the image has no depth-ordered occlusion. To make rendering efficient, the image is divided into fixed tiles. For each tile, only the Gaussians whose 3σ support overlaps the tile are considered. Then, for each pixel, the top-K Gaussians with the highest

responses are retained and renormalized:

$$\hat{I}(x) = \frac{1}{\sum_{i \in \mathcal{N}_K(x)} G_i(x)} \sum_{i \in \mathcal{N}_K(x)} G_i(x)\, c_i, \tag{3.5}$$

where $\mathcal{N}_K(x)$ denotes the retained set of contributors for pixel x.

This design serves two purposes. Computationally, it enforces locality and dramatically reduces the cost of forward and backward passes. Representationally, it encourages the model to explain each image location with a small set of nearby primitives rather than a diffuse superposition of many weak contributions. The top-K normalization also improves optimization behavior in practice by regularizing how color is distributed among overlapping Gaussians. Although the ranking step itself is not differentiable, gradients flow through the retained Gaussians, and this structured sparsity is one reason the renderer scales well to high-resolution images.

3.1.3 Adaptive Initialization and Progressive Optimization

The representation is fit to a target image through stochastic gradient descent. Because the primitives are explicit and interpretable, initialization matters. Rather than sampling Gaussian positions uniformly, the method initializes them using a mixture of image-gradient guidance and uniform coverage:

$$\mathbb{P}_{\text{init}}(x) = (1 - \lambda_{\text{init}}) \frac{\|\nabla I(x)\|_2}{\sum_{x'} \|\nabla I(x')\|_2} + \frac{\lambda_{\text{init}}}{HW}, \quad \lambda_{\text{init}} \in [0, 1] \tag{3.6}$$

where H and W are the image dimensions and λ balances edge awareness and full-domain coverage. High-frequency regions therefore receive denser initial support, while smooth regions are not neglected entirely. Gaussian colors are initialized from the target image at the sampled positions.

During optimization, the rendered image is compared against the target using a combined L_1 and SSIM objective. Additional Gaussians are then introduced progressively into regions with high

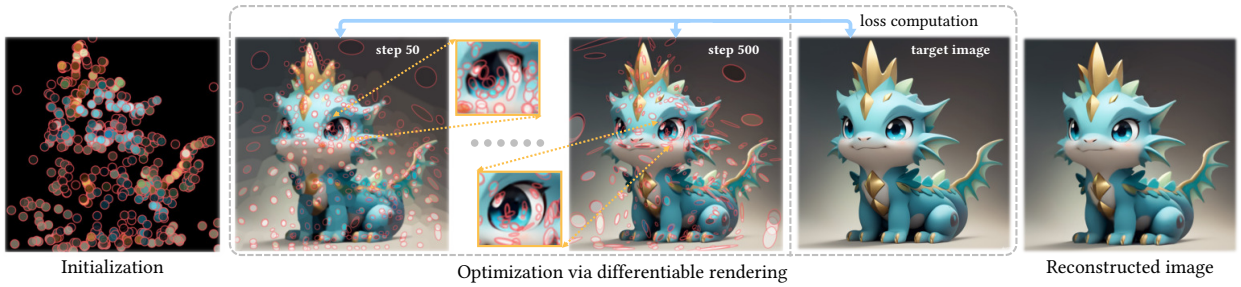


Figure 3.2: *Image-GS optimization pipeline.* At initialization, a set of 2D Gaussians is adaptively spawned based on local image gradient magnitudes, with more Gaussians allocated to high-frequency regions (Section 3.1.3). During training, the Gaussian parameters are optimized using the differentiable renderer described in Section 3.1.2 to reconstruct the target image, and additional Gaussians are progressively allocated to regions exhibiting persistent reconstruction errors. A 20% randomly sampled subset of Gaussians is visualized as colored ellipses, whose scale and orientation are determined by their covariance, to illustrate the optimization progress.

residual error. This error-guided refinement produces two practical benefits. First, it improves reconstruction quality by allocating more capacity where the current approximation is weakest. Second, it naturally creates a progressive hierarchy of increasingly detailed solutions, which can later be exploited as a smooth level-of-detail (LOD) representation. In contrast to many neural image representations that require retraining or architecture changes to move across quality budgets, Image-GS exposes a continuous path from coarse to fine reconstruction within a single optimization run.

The additional Gaussians are sampled from the residual map:

$$\mathbb{P}_{\text{add}}(x) = \frac{\left|\hat{I}(x) - I(x)\right|}{\sum_{x'} \left|\hat{I}(x') - I(x')\right|}, \tag{3.7}$$

so that areas that remain poorly reconstructed are given more capacity over time. In practice, the representation begins with a coarse support set and then refines itself in a content-aware way, which is especially beneficial for images with a small number of visually dominant structures.

This progressive fitting process is also what makes Image-GS useful beyond a single fixed operating point. The optimization path itself becomes a reusable hierarchy of progressively richer

reconstructions, which later sections exploit when evaluating level-of-detail control, compression performance, and downstream applications.

3.2 Image and Texture Compression Performance

This section evaluates Image-GS as a practical image and texture codec. The goal is not only to measure reconstruction fidelity, but also to understand how the representation behaves under constrained memory budgets, how quickly it can be trained and rendered, and whether it provides useful quality control for deployment. We therefore study both conventional compression metrics and graphics-oriented system properties, including progressive refinement, random access, and decoding efficiency.

3.2.1 Experimental Setup

The image compression experiments use a dataset of 45 RGB images spanning vector-style art, photographs, digital art, anime, and paintings, with 9 samples per category. The texture compression experiments use 19 texture stacks, each containing nine channels commonly used in graphics workflows, including diffuse color, normal map, ambient occlusion, roughness, and metalness. All assets were prepared to 2K $\times$ 2K resolution. The source images and textures were obtained from Adobe Stock and Poly Haven, respectively. This dataset design is important because the chapter focuses on graphics-oriented visual assets rather than only natural-image benchmarks.

Image quality is measured using PSNR, MS-SSIM, LPIPS, and FLIP, while texture quality is measured using PSNR and SSIM [79–81]. Compression efficiency is reported in kilobytes (KB), bits per pixel (bpp) for images, and bits per pixel per channel (bppc) for textures. The implementation is built on a differentiable Gaussian rasterization pipeline with custom CUDA kernels for efficient forward and backward computation [82]. All trainable parameters are stored in float16, and the image domain is normalized to $[0, 1]^2$ so the same implementation can handle arbitrary aspect

ratios and resolutions. We train with Adam for 5K steps using a combined $L_1 + 0.1L_{\text{SSIM}}$ objective, initialize Gaussian scales to 5 pixels and rotation angles to 0, and optimize the inverse scales for more stable convergence. The default tile size is 16×16, the top-K setting is $K = 10$, and the initialization interpolation parameter is $\lambda = 0.3$. The learning rates for the Gaussian mean, color, scale, and rotation parameters are 5e−4, 5e−3, 2e−3, and 2e−3, respectively. During progressive fitting, the representation starts with $\lfloor N/2 \rfloor$ Gaussians and adds another $\lfloor N/8 \rfloor$ every 0.5K steps until the target budget is reached.

The main image-compression baselines include recent neural image representations and a Gaussian-based baseline, along with JPEG as a conventional reference [15, 20–22, 83]. For fair comparisons, the baseline implementations are adapted to the same target budget range by reducing grid resolution in ReLU-F and I-NGP, reducing the number of primitives in GI, or shrinking the number of layers and hidden dimensions in I-NGP, WIRE, SIREN, and FFN. Texture compression experiments adopt industry-standard GPU-oriented block compression methods as baselines, including BC1, BC7, and ASTC. These comparisons are designed to test whether an adaptive Gaussian representation can compete simultaneously with neural representations on images and hardware-oriented texture encoders in graphics settings.

3.2.2 Image Compression Performance

The aggregate rate-distortion trends are summarized in Figures 3.3 and 3.4. Across the evaluated bitrate range, Image-GS achieves favorable rate-distortion trade-offs, especially in the low-bitrate regime where spatial adaptivity matters most. At 0.366 bpp, it reaches 32.99 ± 4.49 dB PSNR, 0.966 ± 0.020 MS-SSIM, 0.083 ± 0.057 LPIPS, and 0.078 ± 0.029 FLIP; even at 0.122 bpp, it still maintains 29.20 ± 4.57 dB PSNR, 0.924 ± 0.042 MS-SSIM, 0.173 ± 0.082 LPIPS, and 0.116 ± 0.043 FLIP. It preserves oriented edges, sparse high-frequency details, and stylized structures more effectively than baselines that rely on fixed grids or heavily compressed network capacity. This advantage

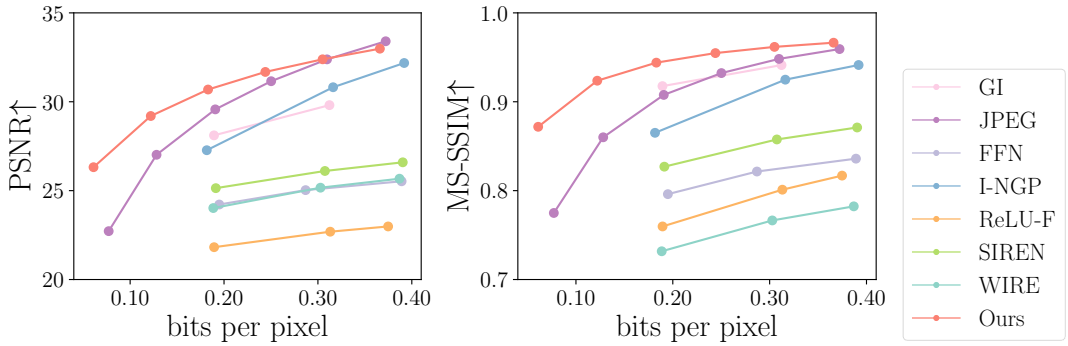


Figure 3.3: *Rate-distortion curves for image compression experiments.* PSNR and MS-SSIM scores averaged over the evaluation set of 45 RGB images described in Section 3.2.1

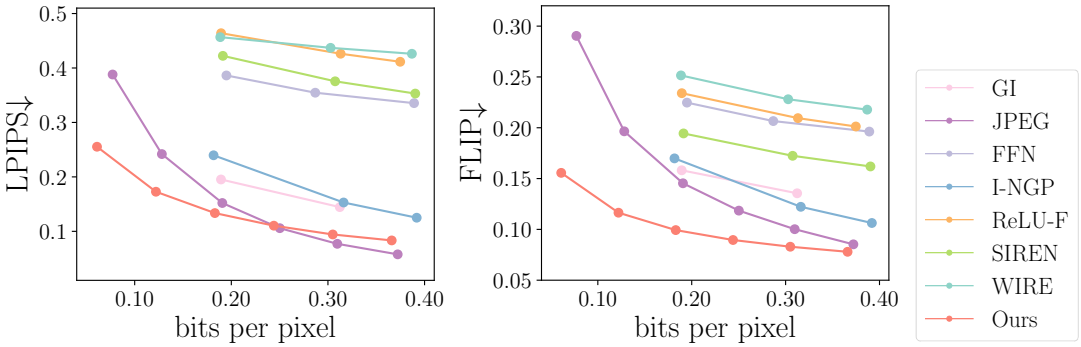


Figure 3.4: *Rate-distortion curves for image compression experiments.* LPIPS and FLIP scores averaged over the evaluation set of 45 RGB images described in Section 3.2.1.

is particularly visible for graphics assets whose important content is concentrated in localized regions rather than spread uniformly across the image. Figure 3.5 provides a compact preview of this behavior under a constrained budget, while Figure 3.6 and Figure 3.7 show the broader qualitative evaluation.

The qualitative comparisons reveal distinct failure modes for the competing methods. Grid-based encodings are prone to block artifacts and loss of local detail when their resolution is reduced. Implicit neural representations tend to introduce ringing, blurring, or ghosting under aggressive compression. By contrast, the Gaussian representation degrades more gracefully because it can concentrate primitives where edges, contrast transitions, and structured details are most important. This content-adaptive behavior is a central reason the method remains competitive under severe memory constraints. When the bitrate falls below 0.244, Image-GS even surpasses JPEG by a

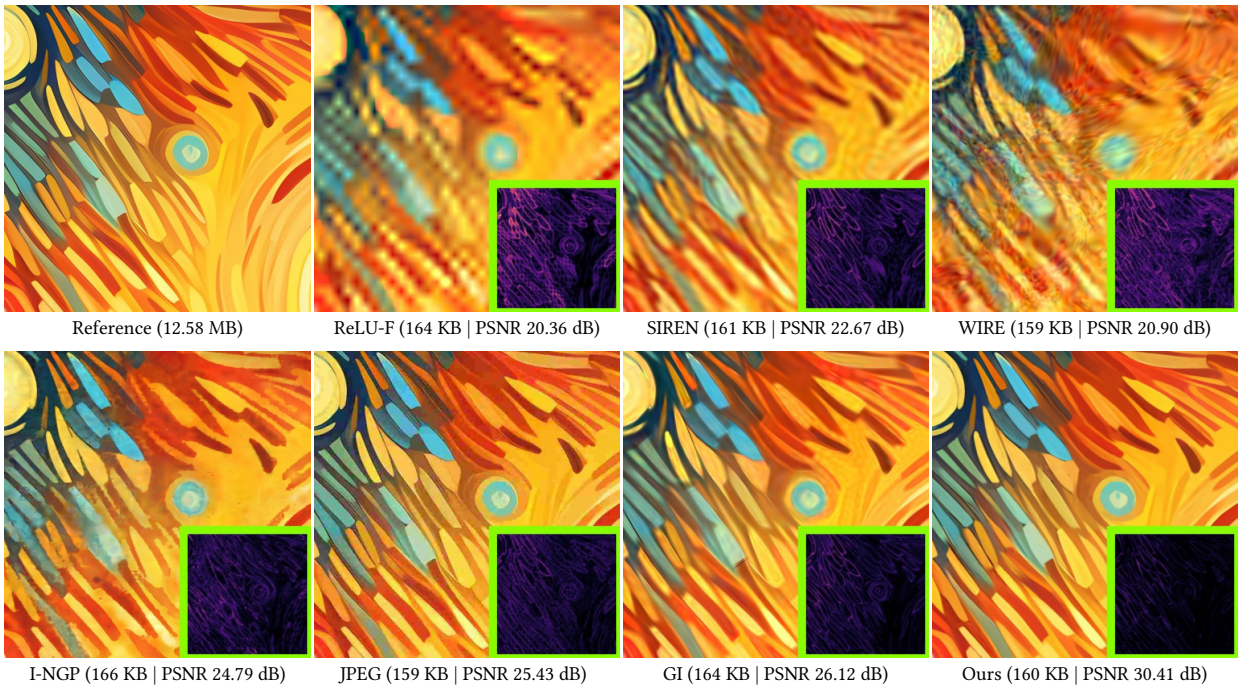


Figure 3.5: *Qualitative comparison under low-bitrate regime.* Image-GS's explicit content-adaptive design effectively captures non-uniformly distributed image features and better preserves fine details under constrained memory budgets. In the inset error maps, brighter colors indicate larger errors.

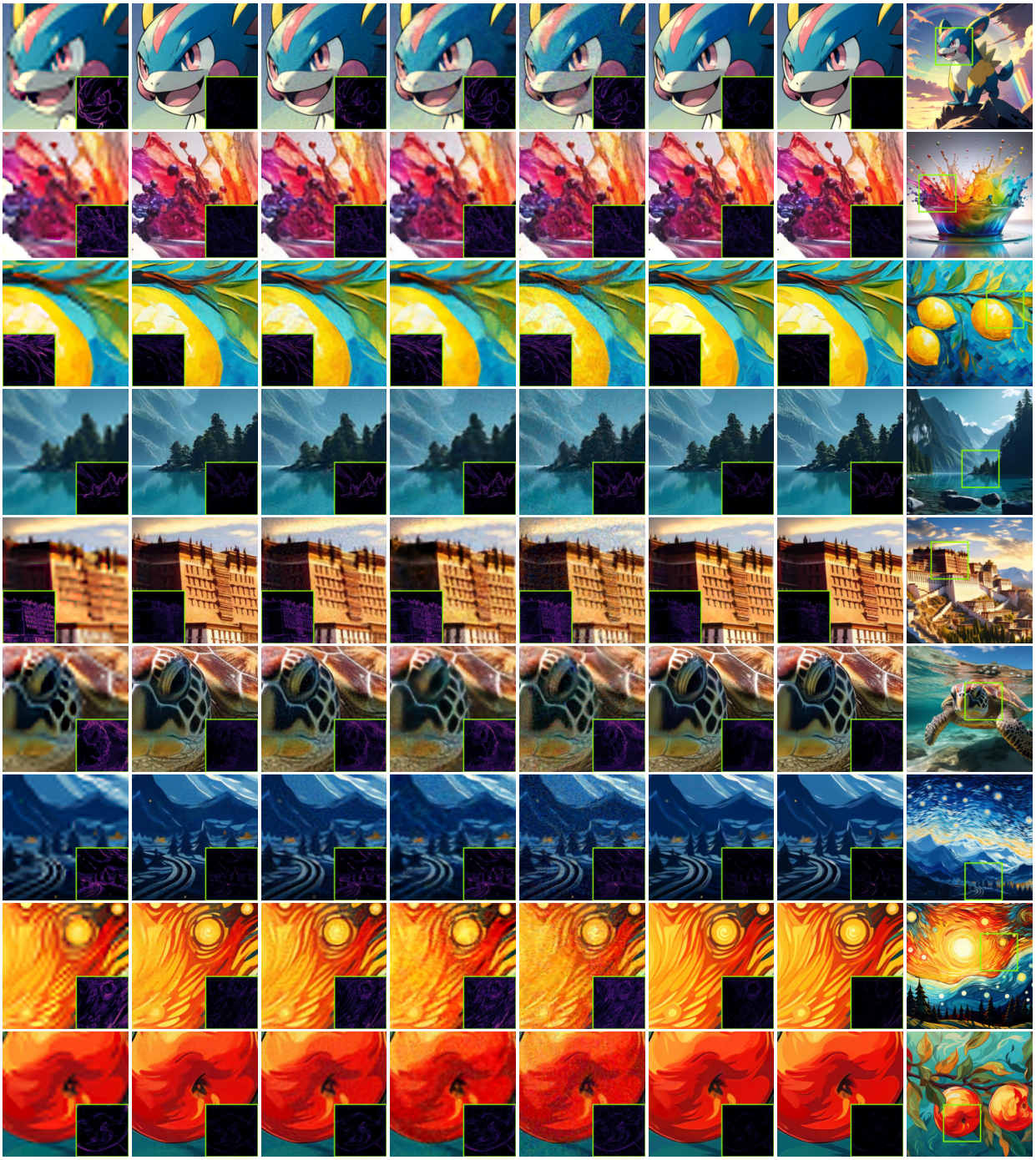
noticeable margin, despite JPEG and GI both relying on entropy coding that breaks data locality in order to improve compression. The examples in Figures 3.6 and 3.7 make this particularly clear for stylized assets, paintings, and digital art, where localized structure matters more than uniformly preserving all pixels.

Although the chapter emphasizes graphics-oriented visual assets, the representation also performs competitively on a more conventional natural-image benchmark. Figure 3.8 reports PSNR and MS-SSIM on the professional validation split of CLIC2020 [84], while Figures 3.9 and 3.10 show qualitative reconstructions at 0.50 bpp. The CLIC results are slightly lower than the graphics-oriented set because natural images contain more pixel-level sensor noise and the constant bitrate budget yields fewer Gaussians per image at the same resolution. These results suggest that the adaptive Gaussian basis generalizes beyond the asset types for which it is most directly motivated, although its strongest advantage still appears on structured graphics content where the spatial



(a) ReLU-F (b) I-NGP (c) SIREN (d) FFN (e) WIRE (f) GI (g) Ours (h) Ref

Figure 3.6: *Qualitative comparison against conventional and neural image representations I.* Representative 2K × 2K results under closely matched memory budgets. For the examples shown here, the model sizes of ReLU-F, I-NGP, SIREN, FFN, WIRE, GI, and Image-GS are 164, 166, 161, 154, 159, 164, and 160 KB, respectively.



(a) ReLU-F (b) I-NGP (c) SIREN (d) FFN (e) WIRE (f) GI (g) Ours (h) Ref

Figure 3.7: *Qualitative comparison against conventional and neural image representations II.* Representative 2K × 2K results under closely matched memory budgets. For the examples shown here, the model sizes of ReLU-F, I-NGP, SIREN, FFN, WIRE, GI, and Image-GS are 164, 166, 161, 154, 159, 164, and 160 KB, respectively.

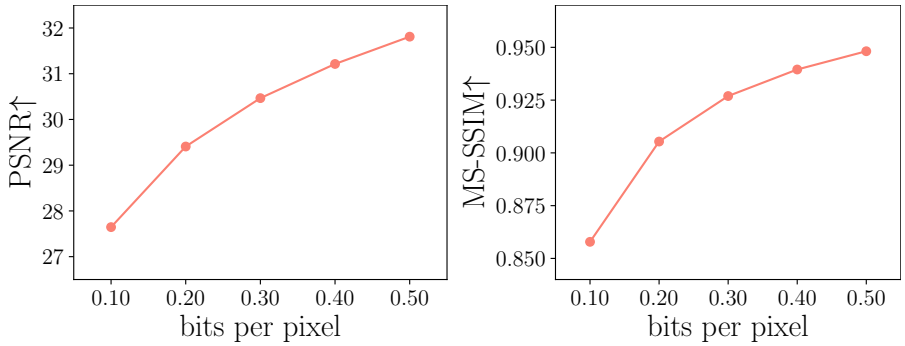


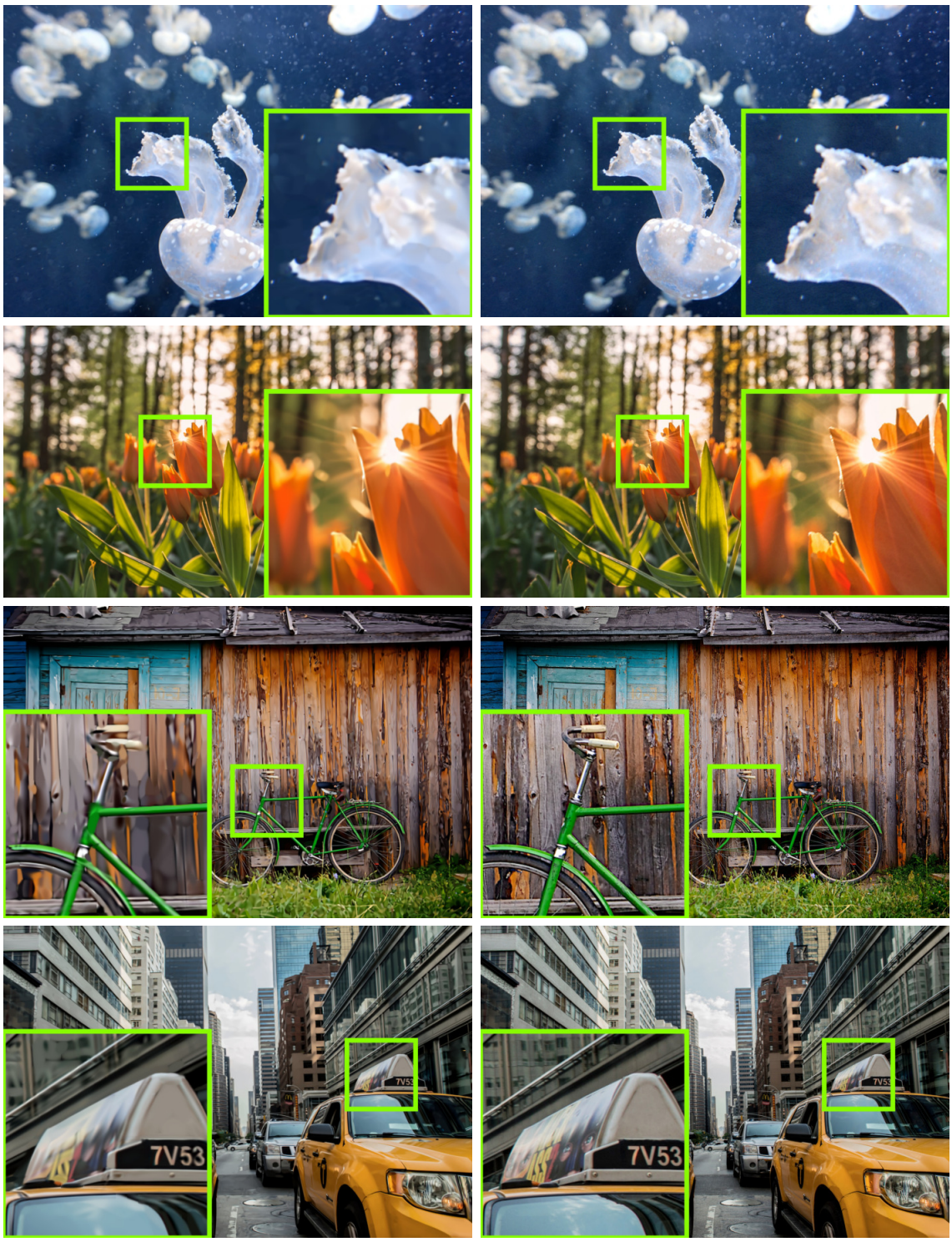
Figure 3.8: *Rate-distortion curves for image compression experiments on the CLIC2020 benchmark.* PSNR and MS-SSIM averaged over the professional validation split of the CLIC2020 dataset [84]. This evaluation complements the graphics-oriented dataset used in the main image compression evaluation.

allocation of primitives can exploit pronounced local structure.

The ablation study in Table 3.1 clarifies which design choices matter most. Content-aware initialization, inverse-scale optimization, and top-K normalization all contribute measurably to the final reconstruction quality. In particular, removing the adaptive position initialization or replacing it with random initialization reduces performance noticeably, confirming that explicit primitives benefit strongly from spatially informed optimization. The same table also shows that the model remains sensitive to the interaction between sampling, rendering sparsity, and optimization stability, which reinforces the point that Image-GS works well because primitive placement and local rendering support are co-designed rather than chosen independently.

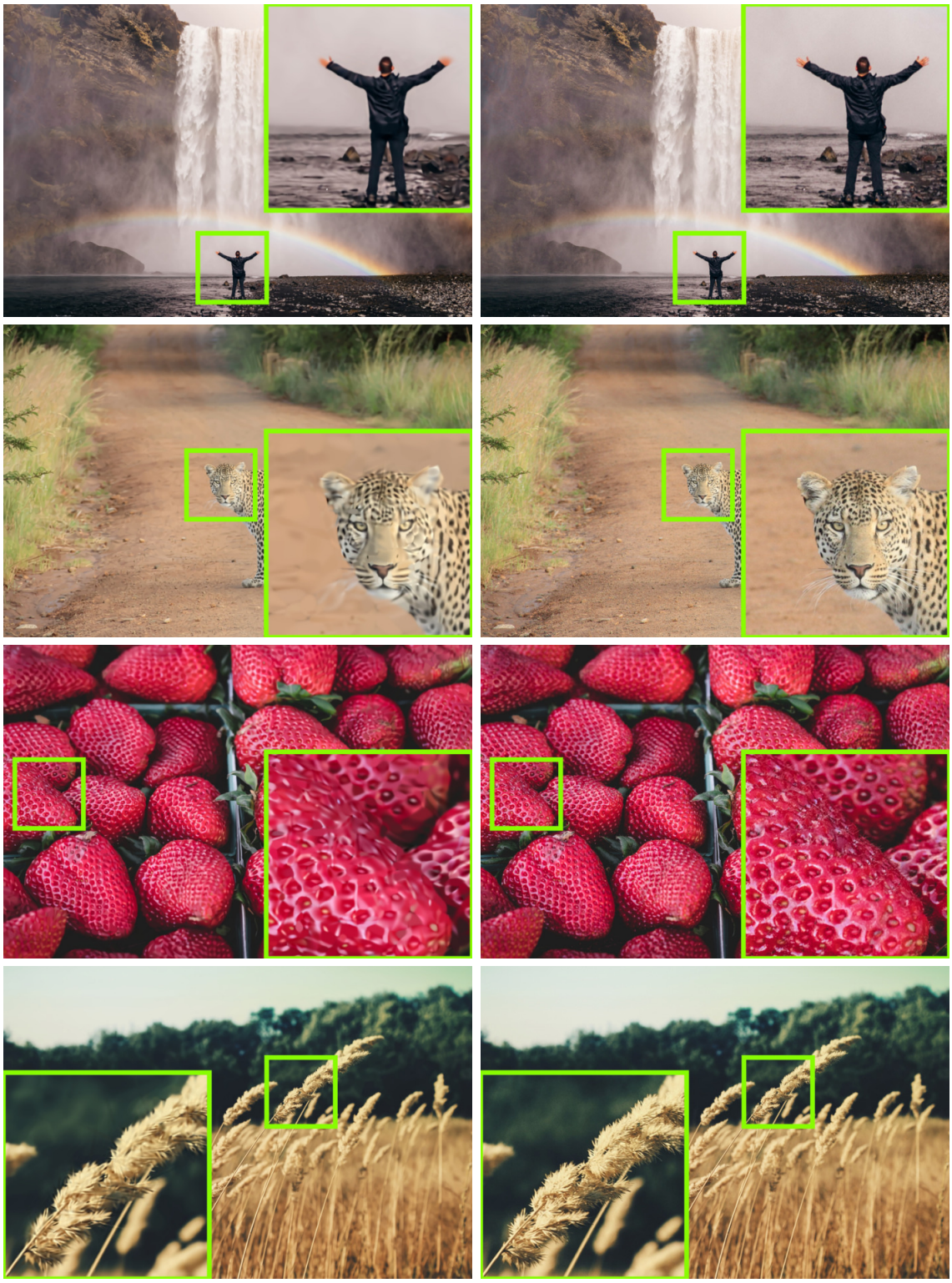
Table 3.1: *Ablation study on the design choices of Image-GS.* The results shown here isolate the effect of content-aware initialization, inverse-scale optimization, and top-K normalization on the reconstruction quality, highlighting the importance of co-designing Gaussian placement and local rendering support.

Variants	PSNR↑	MS-SSIM↑	LPIPS↓
No color init	30.40 ± 4.73	0.951 ± 0.025	0.110 ± 0.064
No position init	29.88 ± 4.18	0.954 ± 0.028	0.135 ± 0.065
Random init	29.54 ± 4.15	0.948 ± 0.029	0.149 ± 0.067
No inverse scale	29.11 ± 4.58	0.933 ± 0.038	0.152 ± 0.073
No top-K norm	29.35 ± 4.43	0.944 ± 0.032	0.182 ± 0.086
Full	31.77 ± 4.73	0.960 ± 0.024	0.102 ± 0.062



(a) Ours (0.50 bpp) (b) Reference

Figure 3.9: *Qualitative results on the CLIC2020 benchmark I.* Qualitative image compression results on the professional validation split of the CLIC2020 dataset [84] at 0.50 bpp.



(a) Ours (0.50 bpp) (b) Reference

Figure 3.10: *Qualitative results on the CLIC2020 benchmark II.* Qualitative image compression results on the professional validation split of the CLIC2020 dataset [84] at 0.50 bpp.

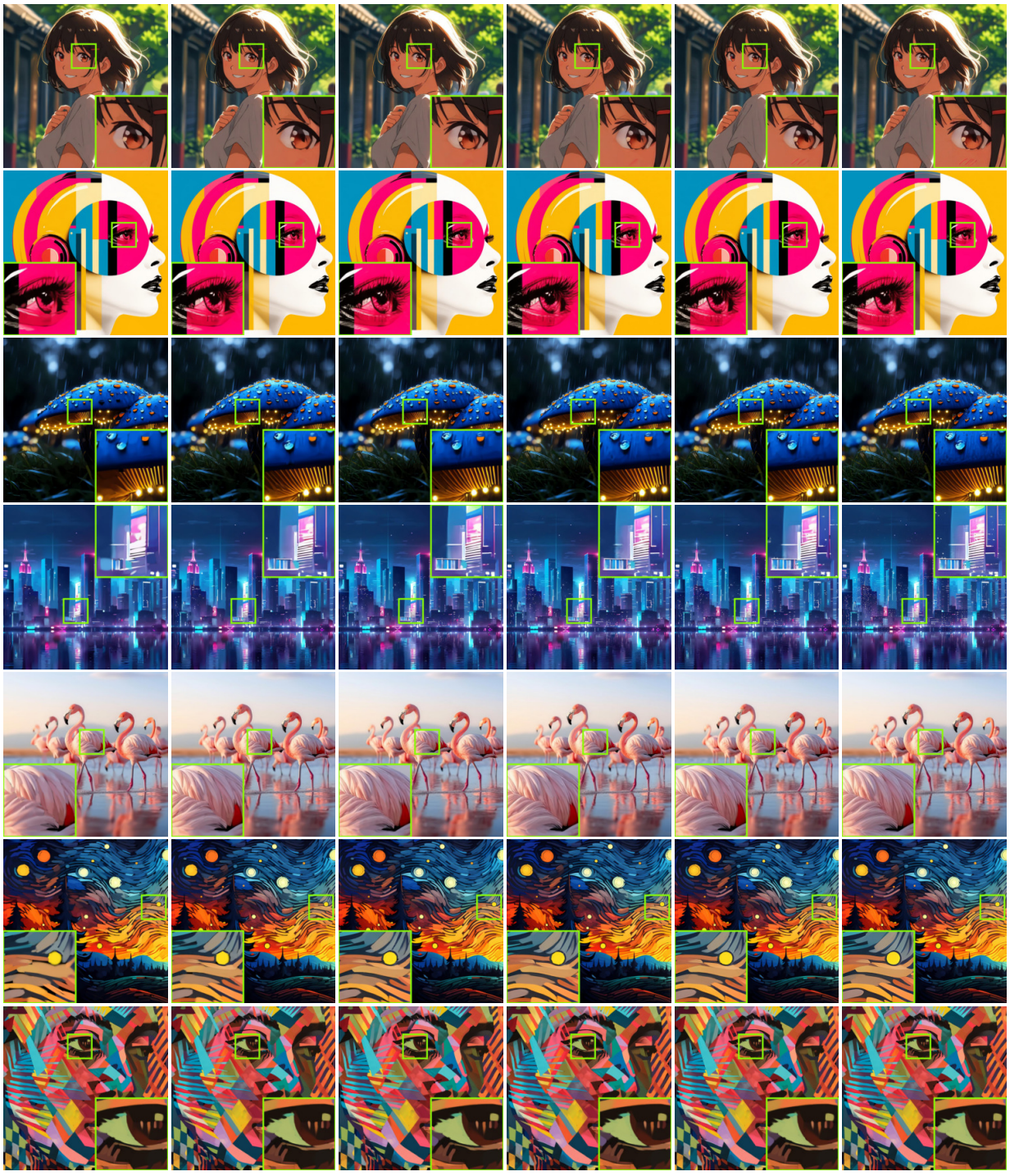
3.2.3 Progressive Level of Detail and System Performance

The optimization procedure naturally produces a smooth LOD hierarchy because Gaussians are added progressively to reduce reconstruction error. Figures 3.11 and 3.12 visualize this coarse-to-fine refinement process. This means the same training run yields a continuum of operating points, enabling deployment-time control over memory usage and quality without retraining. Such behavior is especially useful in graphics systems where the available budget may change across devices or across runtime conditions.

In addition to representational quality, the implementation is efficient. The runtime measurements reported in Figure 3.13 show that training and rendering both scale well with the number of Gaussians, and the explicit representation supports fast random access and parallel decoding. On an Nvidia A6000 GPU, optimizing 10K Gaussians for 1K steps at 2K×2K resolution takes 18.74 seconds and rendering takes only 0.0037 seconds, while 50K Gaussians increase those times only modestly to 26.32 seconds and 0.0045 seconds. The representation therefore behaves more like a practical codec than a heavyweight neural model, which makes it attractive for graphics use where decoding latency and random access matter as much as reconstruction quality. Together with the smooth LOD hierarchy in Figures 3.11 and 3.12, these timing results show that Image-GS is not only compact but also operationally practical.

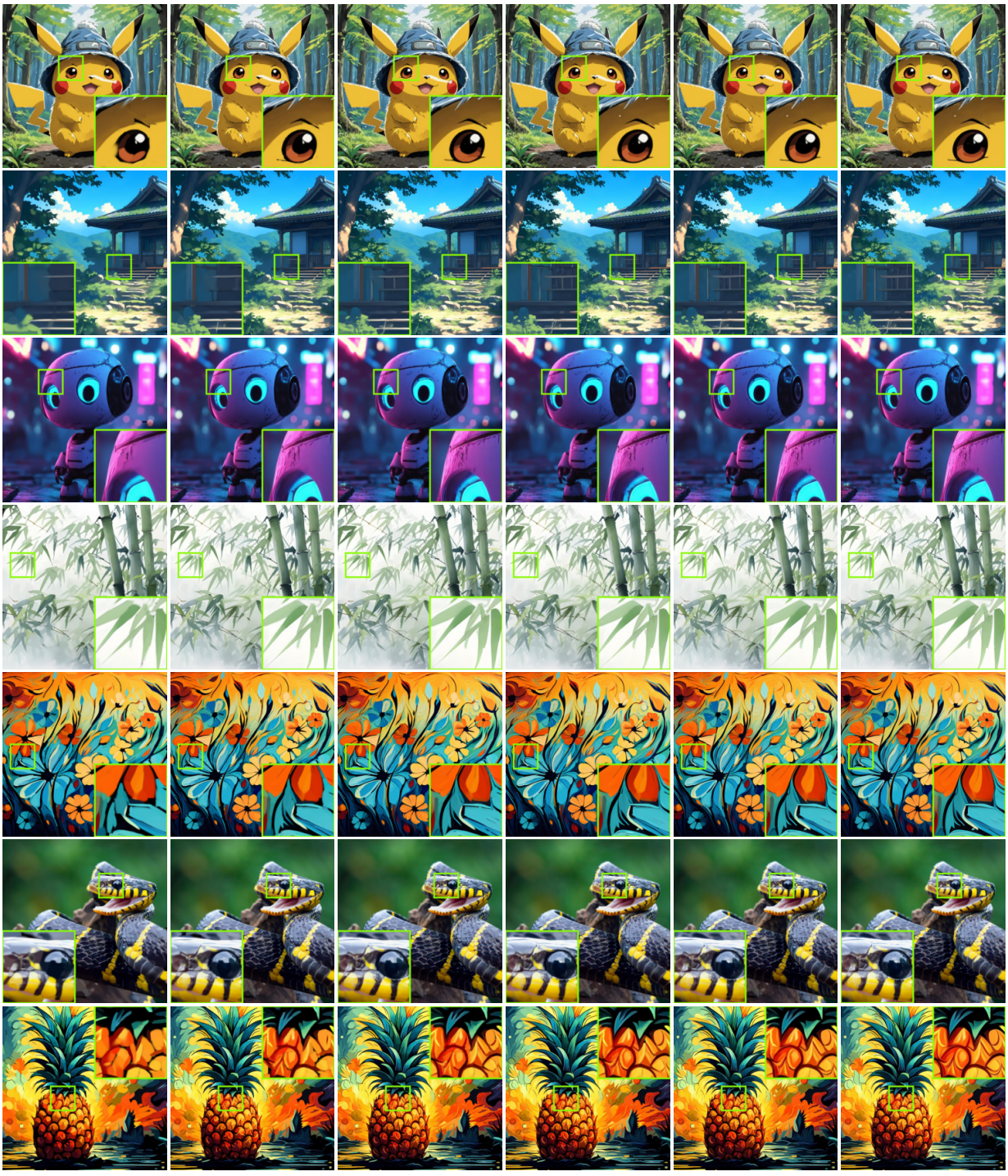
3.2.4 Texture Compression Performance

Texture compression presents a different test from image compression because the target assets are intended for rendering on GPUs and often contain multi-channel material information rather than only natural-image appearance. On this task, Image-GS consistently outperforms BC1 and BC7 and remains competitive with ASTC at comparable bitrates. At an extreme bitrate of 0.059 bppc, it still reaches 32.20 ± 4.06 dB PSNR and 0.869 ± 0.107 SSIM. These results are significant because they show that the same underlying Gaussian representation can serve both image-centric and



(a) 0.061 bpp (b) 0.122 bpp (c) 0.183 bpp (d) 0.244 bpp (e) 0.305 bpp (f) Reference

Figure 3.11: *Image-GS's rate-distortion trade-off through progressive optimization I.* Through the error-guided progressive optimization described in Section 3.1.3, Image-GS naturally constructs a smooth level-of-detail hierarchy in a single optimization run without additional overhead. The resulting hierarchy enables flexible quality adaptation to different device and bitrate budgets.



(a) 0.061 bpp (b) 0.122 bpp (c) 0.183 bpp (d) 0.244 bpp (e) 0.305 bpp (f) Reference

Figure 3.12: *Image-GS's rate-distortion trade-off through progressive optimization II.* Through the error-guided progressive optimization described in Section 3.1.3, Image-GS naturally constructs a smooth level-of-detail hierarchy in a single optimization run without additional overhead. The resulting hierarchy enables flexible quality adaptation to different device and bitrate budgets.

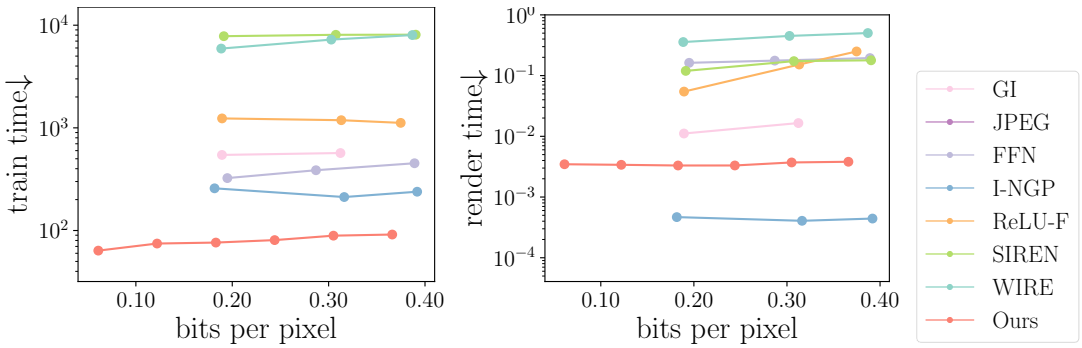


Figure 3.13: *System performance for image compression.* Training and rendering times measured for the image compression experiments. These results relate to the image benchmarks in Section 3.2.2. Our explicit Gaussian representation decodes efficiently, and the runtime grows only modestly as the Gaussian budget increases from 10K to 50K.

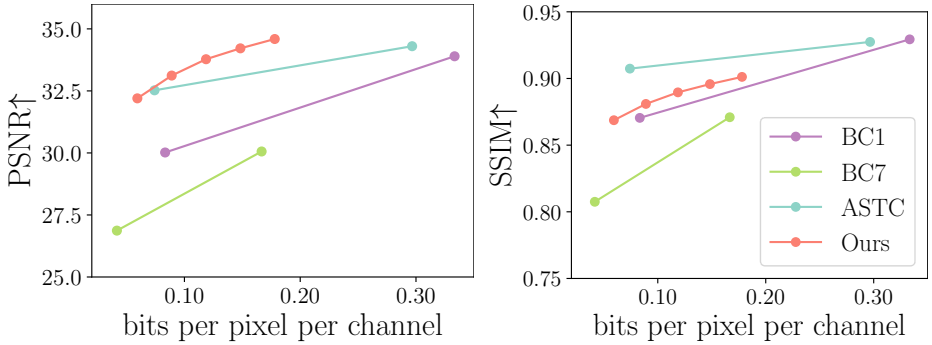
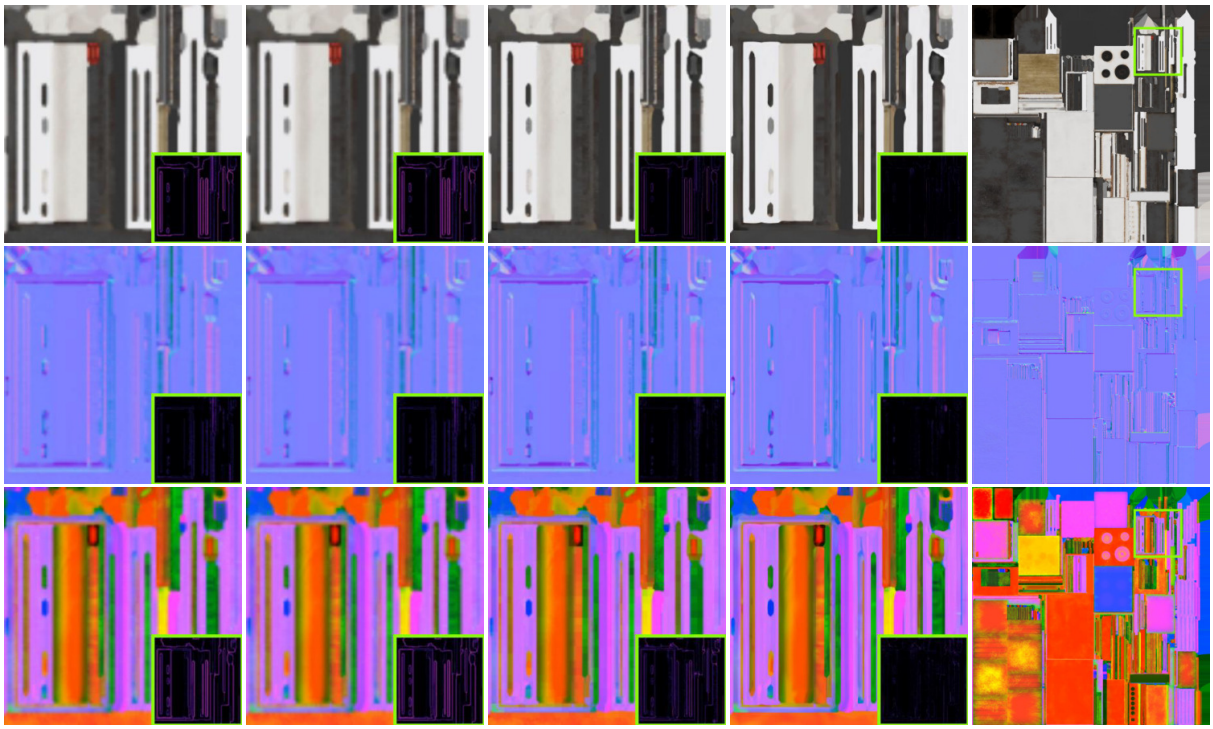


Figure 3.14: *Rate-distortion curves for texture compression experiments.* PSNR and SSIM scores averaged over the evaluation set of 19 multi-channel texture stacks described in Section 3.2.1.

texture-centric use cases without being specialized to one narrow domain. Figure 3.14 summarizes the aggregate texture rate-distortion trends, while Figure 3.15 shows representative qualitative results across competing GPU-oriented codecs. Additional examples are provided in Figures 3.16 to 3.18.

Taken together, the evaluation demonstrates that Image-GS is not merely a new reconstruction model. It is a practical compression-oriented representation that combines rate-distortion performance, progressive quality control, and GPU-friendly access patterns in a way that is well aligned with graphics workloads. These properties are what make the applications in the next two sections possible: once the representation remains explicit after compression, it can be biased



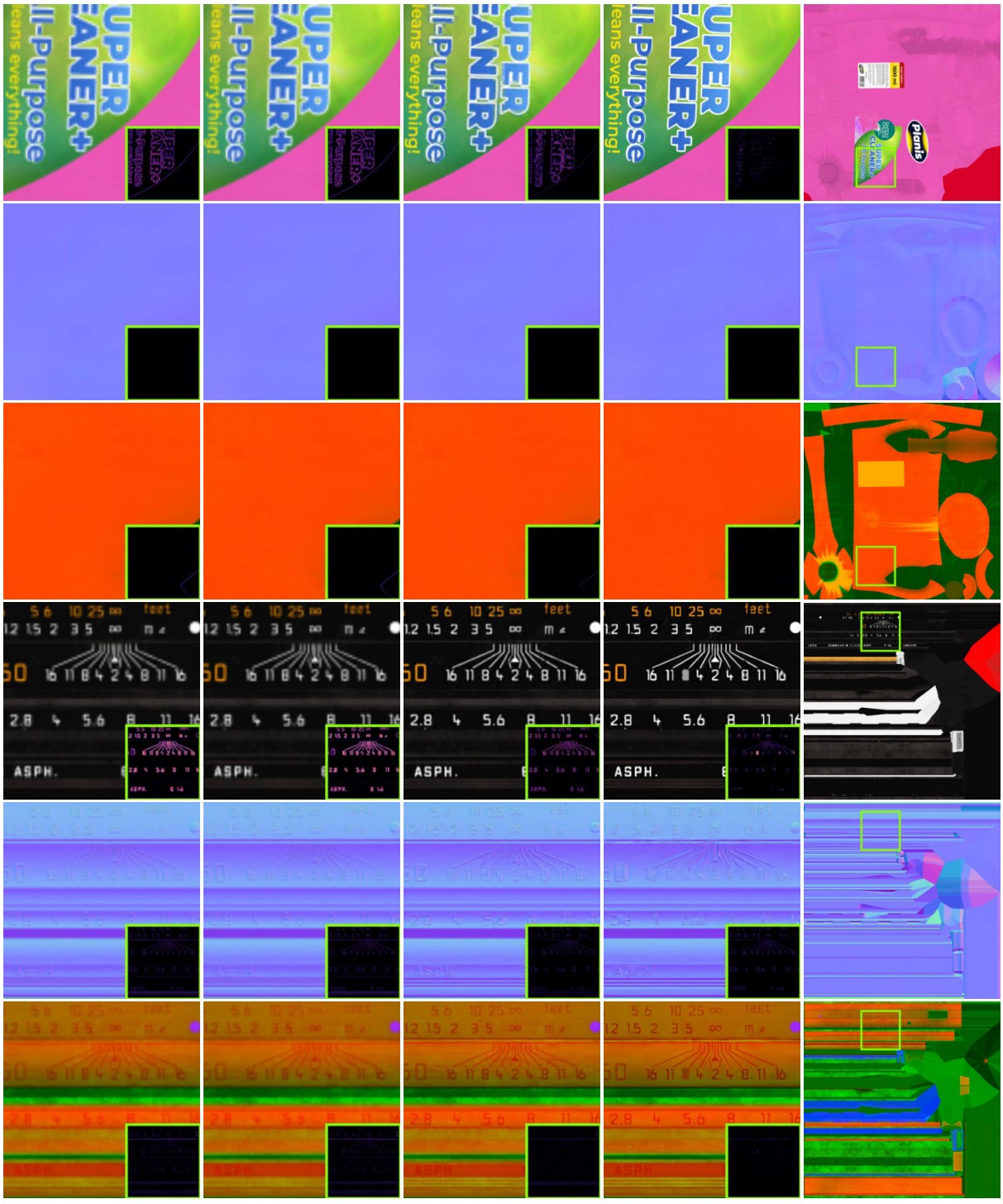
(a) BC1 (b) BC7 (c) ASTC (d) Ours (e) Reference

Figure 3.15: *Qualitative comparison against industry-standard GPU texture compression algorithms I.* Representative 2K × 2K results under closely matched memory budgets: BC1 (0.083 bppc), BC7 (0.167 bppc), ASTC (0.074 bppc), and Image-GS (0.089 bppc).

toward semantics or used as a restoration prior without changing the underlying model family.

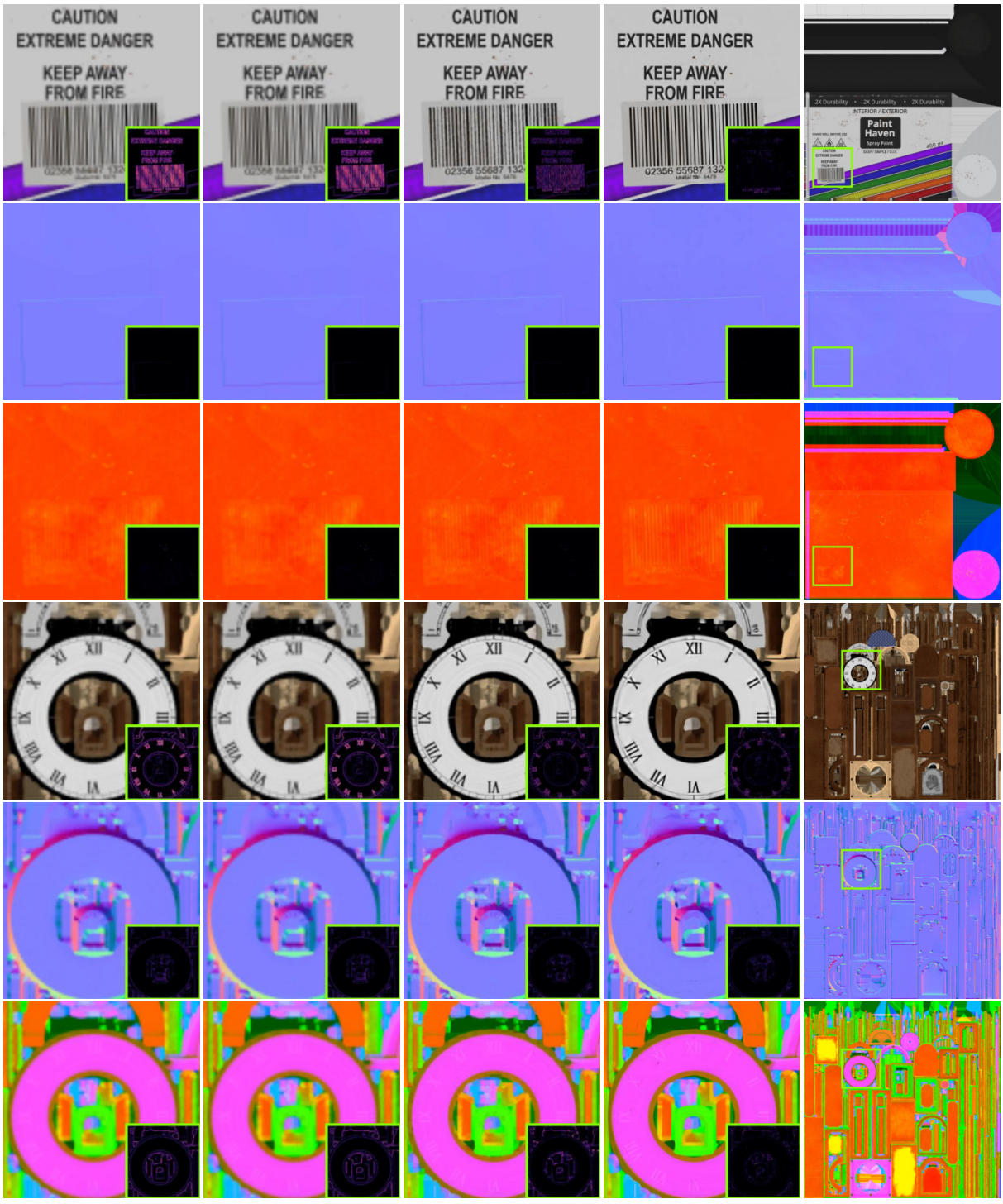
3.3 Semantics-Aware Image Compression

One advantage of an explicit primitive-based representation is that resource allocation can be guided by signals other than reconstruction error alone. In modern cloud-based AI deployment, visual content is often compressed not only for human viewing but also for downstream machine-vision tasks, so preserving semantically important information is critical to both users and service providers [85]. In this section, Gaussian placement is biased using a saliency map predicted from the target image, allowing the representation to preserve semantically important regions more aggressively than less relevant ones.



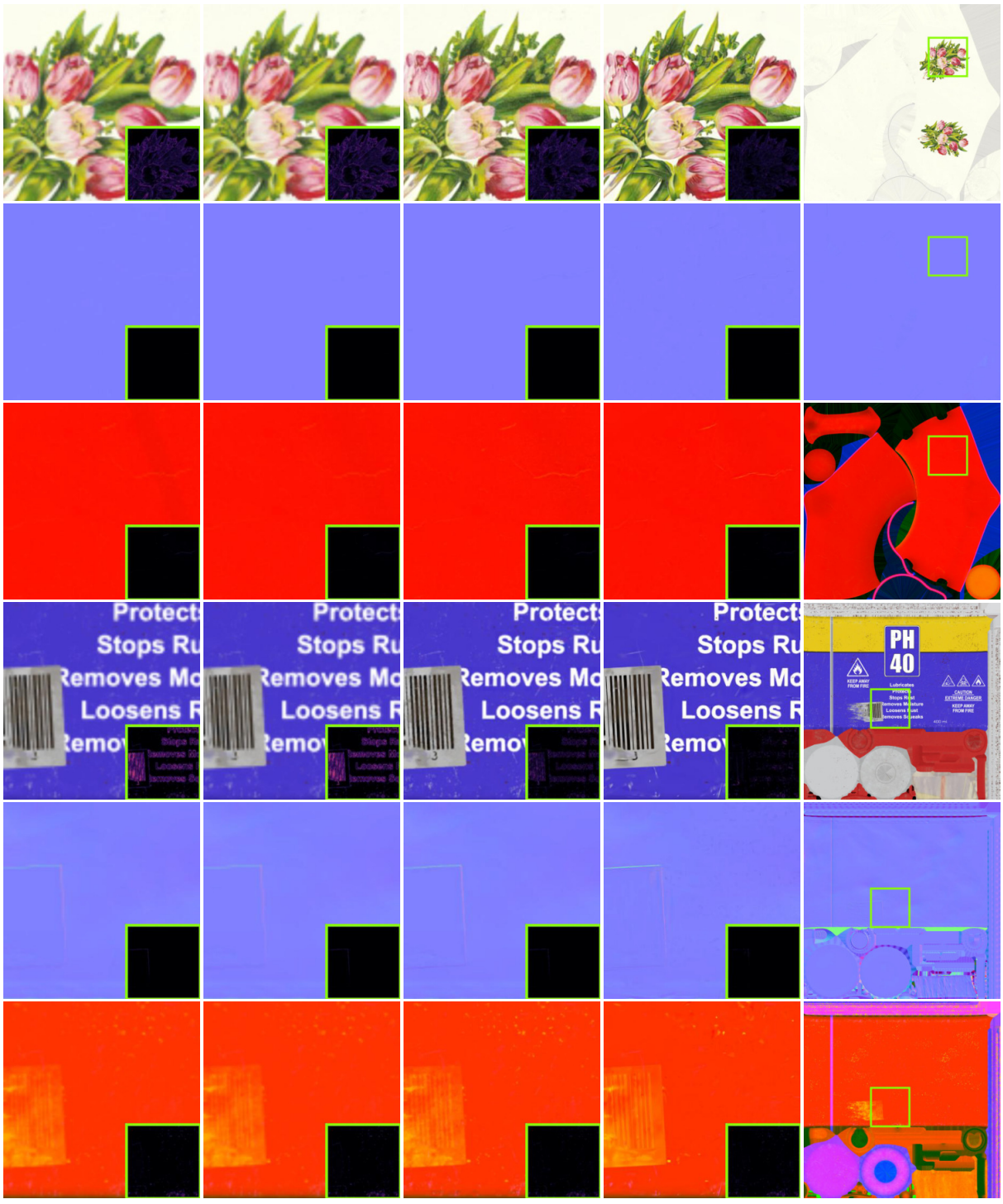
(a) BC1 (b) BC7 (c) ASTC (d) Ours (e) Reference

Figure 3.16: *Qualitative comparison against industry-standard GPU texture compression algorithms II.* Representative 2K × 2K results under closely matched memory budgets: BC1 (0.083 bppc), BC7 (0.167 bppc), ASTC (0.074 bppc), and Image-GS (0.089 bppc).



(a) BC1 (b) BC7 (c) ASTC (d) Ours (e) Reference

Figure 3.17: *Qualitative comparison against industry-standard GPU texture compression algorithms III.* Representative 2K × 2K results under closely matched memory budgets: BC1 (0.083 bppc), BC7 (0.167 bppc), ASTC (0.074 bppc), and Image-GS (0.089 bppc).



(a) BC1 (b) BC7 (c) ASTC (d) Ours (e) Reference

Figure 3.18: *Qualitative comparison against industry-standard GPU texture compression algorithms IV.* Representative 2K × 2K results under closely matched memory budgets: BC1 (0.083 bppc), BC7 (0.167 bppc), ASTC (0.074 bppc), and Image-GS (0.089 bppc).

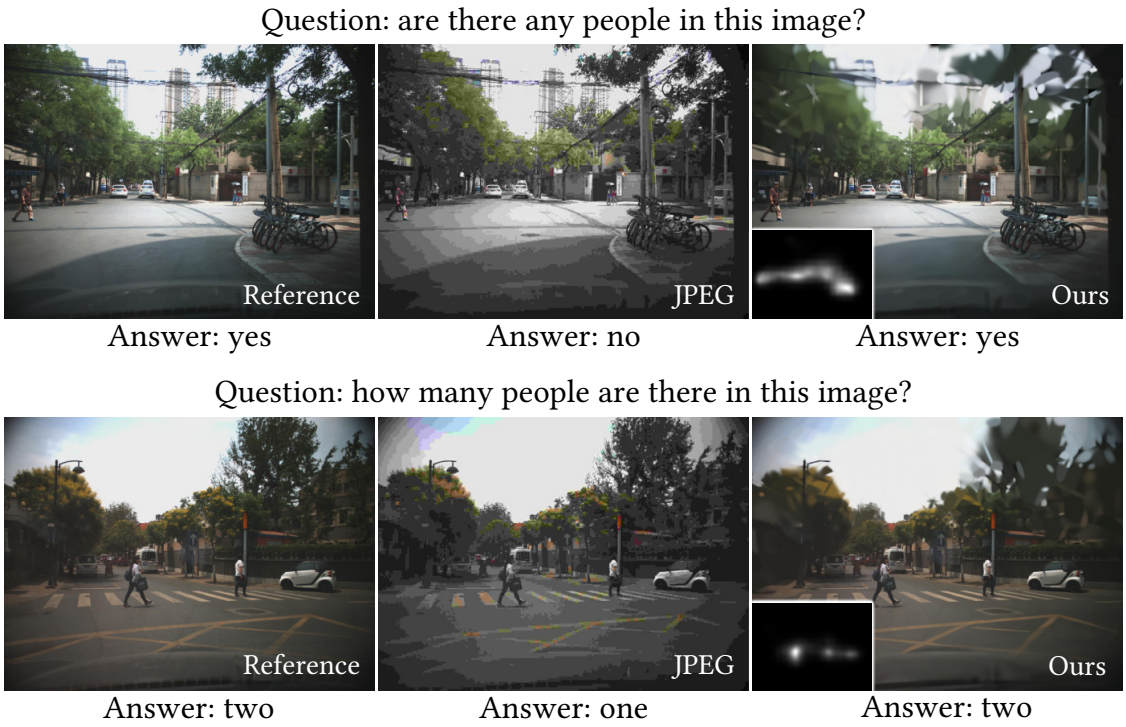


Figure 3.19: *Semantics-aware image compression for machine vision.* At an extreme bitrate of 0.2 bpp, the saliency-guided Image-GS variant preserves task-relevant semantic content more effectively than JPEG in the illustrated BLIP-2 visual-question-answering examples.

Thanks to the explicit nature of Image-GS, we can readily factor in the distribution of such information through visual saliency analysis [39]. Specifically, given a target image, we first use an off-the-shelf saliency predictor [44] to extract a saliency map $S \in \mathbb{R}_{+}^{H \times W}$, and then perform saliency-guided Gaussian position initialization with sampling probability

$$\mathbb{P}_{\text{init}}(x) = \frac{(1 - \lambda_{\text{init}})\, S(x)}{\sum_{x'} S(x')} + \frac{\lambda_{\text{init}}}{H \cdot W}, \qquad \lambda_{\text{init}} \in [0, 1] \tag{3.8}$$

In our experiments, $\lambda_{\text{init}} = 0.1$ balances saliency guidance and uniform coverage. Pixels with higher predicted saliency are therefore sampled more often, while a uniform component is retained to preserve full-image coverage.

The broader motivation for this variant is that compression is often judged not only by pixel fidelity but also by whether the compressed representation preserves the information needed by a

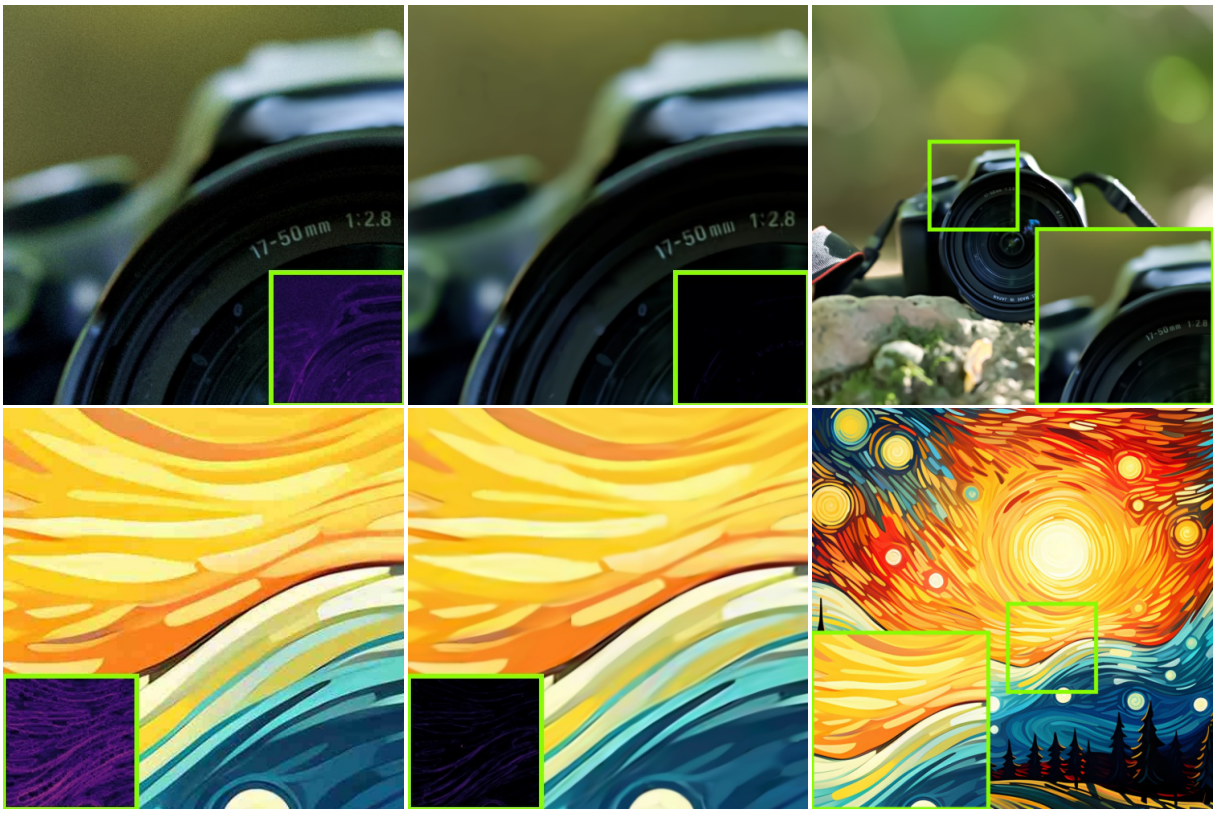
downstream task. To make this concrete, we evaluate Image-GS on a visual-question-answering scenario [86]. JPEG serves as the baseline, and BLIP-2 [87] is used to generate responses. We experiment with 20 randomly sampled images from the TJU-DHD dataset [88] and prepare custom questions for evaluation. The comparison is therefore not simply about perceptual image quality; it asks whether compression preserves the information needed by a downstream machine-vision system.

Under a very low bitrate budget of 0.2 bpp, the saliency-guided Image-GS variant produces compressed images that preserve task-relevant content more reliably than JPEG. Figure 3.19 visualizes two representative image-question pairs. In both cases, the saliency-guided allocation spends more capacity on the semantic regions that drive the answer, which in turn allows the compressed output to elicit the same response as the uncompressed reference. Across all 20 image-question pairs, Image-GS matches the reference response 12 times, whereas JPEG does so only 4 times. These results show that the representation can be optimized not only for appearance but also for semantic utility.

The broader significance of this result is that compression can be aligned with semantic utility rather than only with pixel-space fidelity. When a representation exposes explicit, spatially localized primitives, it becomes relatively straightforward to change where capacity is allocated. This makes the approach attractive not only for human viewing, but also for scenarios where compressed images serve as inputs to downstream analysis systems. In the context of this dissertation, this is one more example of a representation whose structure can be tuned to human or task-level priorities instead of purely signal-level objectives.

3.4 Joint Image Compression and Restoration

The same low-pass bias that makes Gaussian primitives efficient for structured image representation also gives them a useful restoration effect. When the representation is fit under a restricted

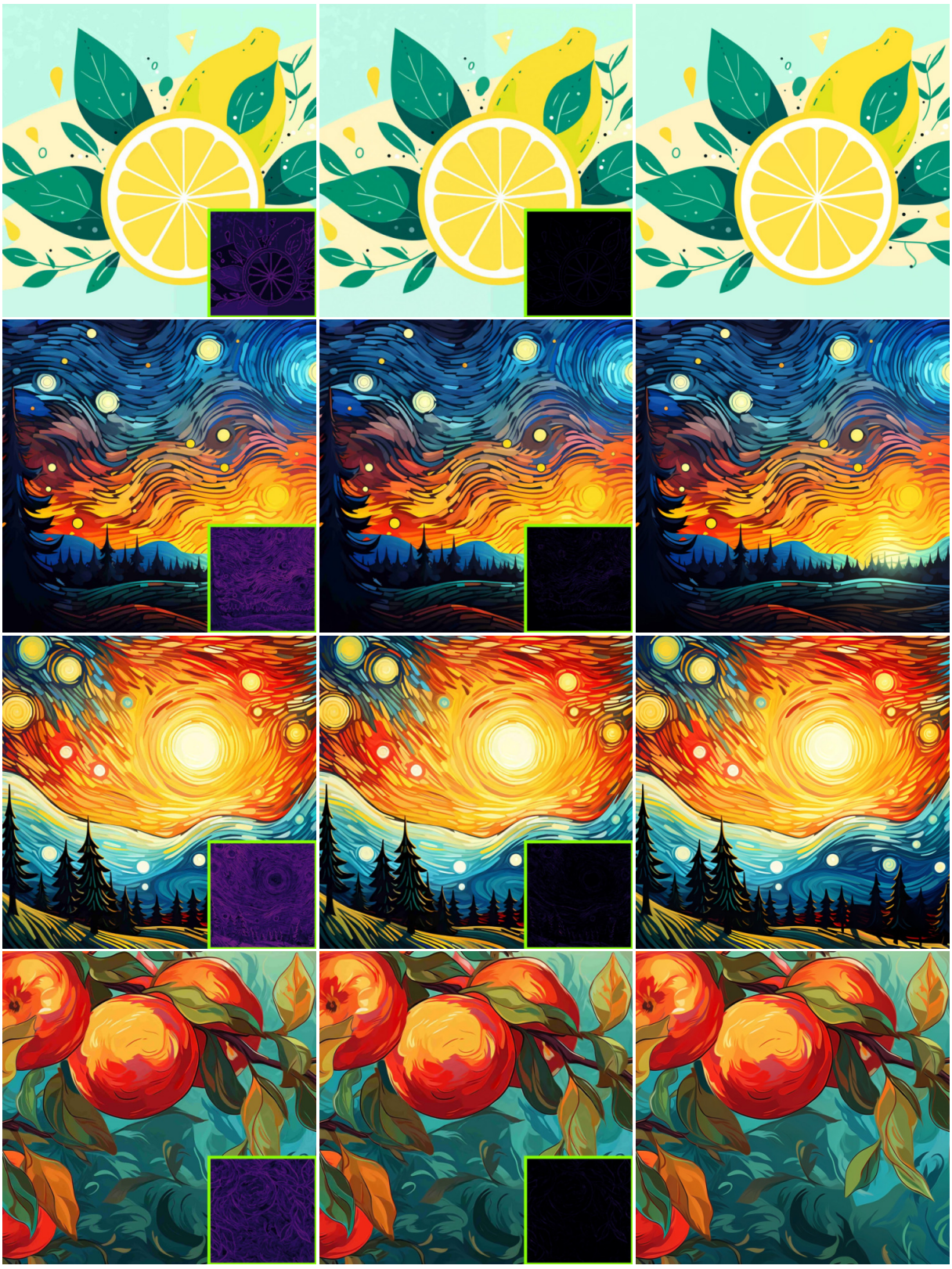


(a) Noisy input image (b) Restored by Image-GS (c) Reference

Figure 3.20: *Joint image compression and restoration I.* At a low bitrate, Image-GS effectively removes high-frequency artifacts from degraded inputs while preserving stable semantic structure. The examples shown here illustrate the restoration of noisy photographs and JPEG-compressed images.

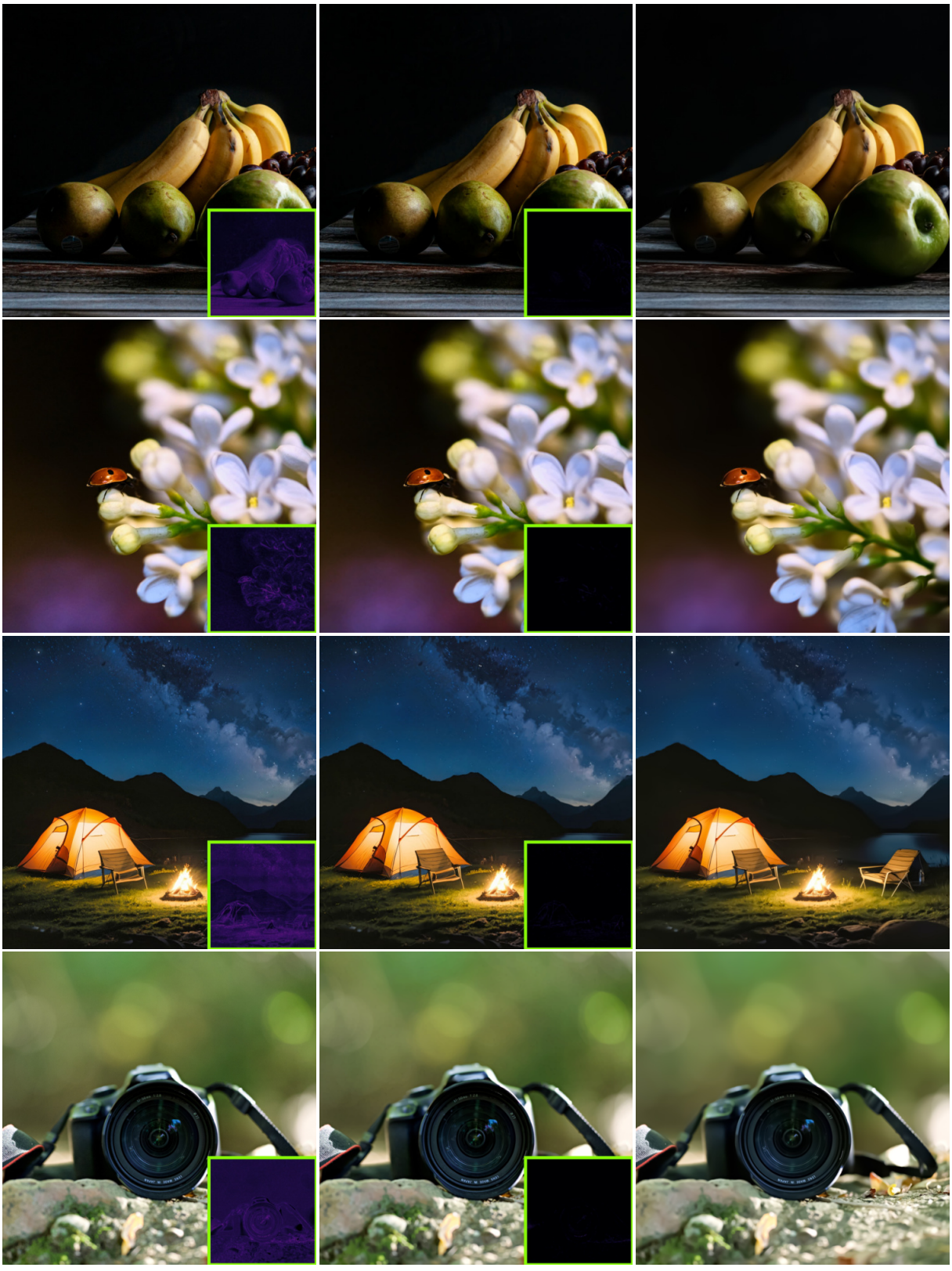
memory budget, it tends to preserve coherent semantic content while suppressing unstable pixel-level artifacts such as sensor noise, ringing, compression contours, color banding, and aliasing. In other words, the compression process itself acts as a form of restoration. The low-pass nature of Gaussian functions endows Image-GS with remarkable robustness against a range of common image distortions and artifacts at low bitrates, including the ringing patterns introduced by lossy compression [89], the color banding and aliasing caused by quantization errors [90], and various forms of transmission or sensor noise [91].

We demonstrate this behavior on two sets of 2K × 2K degraded inputs: six JPEG-compressed images, whose average file size is 187 KB, and eight photographs containing noticeable sensor



(a) Image compressed by JPEG (b) Restored by Image-GS (c) Uncompressed image

Figure 3.21: *Joint image compression and restoration II.* At a low bitrate, Image-GS effectively removes high-frequency artifacts from degraded inputs while preserving stable semantic structure. The examples shown here illustrate the restoration of noisy photographs.



(a) Image with camera noise (b) Restored by Image-GS (c) AI-denoised image

Figure 3.22: *Joint image compression and restoration III.* At a low bitrate, Image-GS effectively removes high-frequency artifacts from degraded inputs while preserving stable semantic structure. The examples shown here illustrate the restoration of JPEG-compressed images.

noise. For quantitative evaluation, the corresponding uncompressed PNG images are used as ground truth for the JPEG images, while AI-denoised outputs serve as reference images for the noisy photographs. In both cases, Image-GS produces reconstructions that are visually cleaner and quantitatively closer to the chosen reference than the degraded inputs. The restoration is not introduced through a separate denoising network or post-processing stage. Instead, it emerges naturally because the representation must spend a limited number of primitives on stable image structure rather than on inconsistent high-frequency corruption.

Figure 3.20 presents the main restoration comparisons, and Figures 3.21 and 3.22 extend them with additional examples. As shown in these figures, Image-GS eliminates most artifacts and noise while preserving the detailed semantic content in the input images. This is because the limited representation budget of 160 KB, corresponding to a 78.64× compression ratio, forces Image-GS to prioritize bits on the more prominent image content instead of inconsistent pixel-level corruption. Despite these high compression ratios, the reconstructed images consistently improve fidelity to the chosen references. On the eight noise-corrupted photos, Image-GS achieves average gains of 1.782 in PSNR and 0.011 in MS-SSIM. On the six JPEG-compressed images, it achieves average gains of 0.354 in PSNR and 0.012 in MS-SSIM.

This application illustrates a broader theme that recurs throughout the dissertation. Representations are not neutral containers. Their inductive biases influence what information is preserved, what information is discarded, and how gracefully a system behaves under limited resources. In this chapter, the Gaussian basis encourages a useful compromise: it remains expressive enough to preserve detailed image structure while being selective enough to suppress many forms of nuisance variation at low bitrates. Notably, these restoration effects emerge directly from the compression process itself, requiring no additional restoration model. That makes Image-GS useful not only as an image codec but also as a compact structural prior for recovering degraded visual data.

3.5 Discussion

This chapter presented an explicit, content-adaptive image representation built from colored 2D Gaussians. This representation combines several properties that are usually difficult to obtain together: competitive rate-distortion behavior, explicit random-access structure, efficient GPU-oriented rendering, and smooth level-of-detail control. Beyond pure reconstruction, it also supports semantics-aware compression and joint compression-restoration, illustrating that its value extends beyond matching pixels as accurately as possible. Taken together, these results show that Image-GS serves as a compact and reusable visual substrate for downstream computation rather than only as a terminal compressed file format.

The main limitation is that the current optimization strategy still favors larger-scale image structure over very fine-grained, noise-like detail. As a result, Image-GS is especially effective for stylized assets and structured graphics content, while dense natural-image detail remains more challenging at extreme compression settings. This limitation is not surprising: the Gaussian basis is explicitly smooth, so it naturally prefers coherent content over isolated pixel variation. In practice, that makes Image-GS a good fit for graphics-oriented assets, but less ideal as a universal replacement for all image codecs without additional structure.

A promising next step is to introduce stronger spatially adaptive optimization or hierarchical partitioning so that primitives can be allocated even more precisely across scales. The level-of-detail results already suggest that the progressive refinement mechanism is useful, but a more aggressive multiscale tree or dynamic binary space partitioning strategy could further improve efficiency for difficult regions while keeping the representation compact elsewhere. Another direction is to extend Image-GS from static images and textures to dynamic visual data such as video, where explicit Gaussian motion models could enable similarly efficient, content-adaptive representations over time. That extension would also make Image-GS more relevant to immersive streaming and mixed-reality delivery, where content often changes continuously and bandwidth

budgets fluctuate. A related practical limitation is that the current fitting procedure is still an offline optimization process; while the resulting representation decodes efficiently, generating it remains more expensive than simply encoding an image with a conventional codec.

More specifically, the current formulation tends to devote more capacity to large image features than to dense pixel-level texture noise. That behavior is visible in the more challenging natural-image cases from the CLIC benchmark, where the results are still competitive but less dramatic than on structured graphics content. A dynamic binary space partitioning tree would be a natural way to refine primitive placement and gradient allocation across scales. This could help balance Image-GS's preference for smooth content with the need to preserve fine details when the input image contains many small structures.

More broadly, this chapter establishes a theme that recurs throughout the dissertation: a graphics representation is useful not just because it reconstructs the input, but because it exposes the right control knobs for downstream systems to reason about quality, semantics, and applications. Image-GS is a concrete example of that idea in the image domain, and the later chapters extend the same principle to 3D rendering and attention-guided generation. In other words, this chapter is not only about encoding pixels more compactly, it is also about building an explicit visual substrate that can be optimized for different objectives without changing the underlying representational form.

Chapter 4

Perceptually Guided 3D Gaussian Splatting

Mixed reality (MR) places unusually strict demands on visual rendering. Displays are wide field of view, often stereoscopic, and expected to update at interactive frame rates while running within tight power and compute budgets. These constraints make straightforward high-fidelity neural rendering difficult to deploy, even when modern scene representations such as 3D Gaussian Splatting (3DGS) provide substantial speedups over volumetric radiance-field methods [23, 30]. In practice, MR rendering must balance three competing requirements at once: reconstruction quality, latency, and computational efficiency.

Recent advances in 3DGS have already transformed traditional graphics pipelines and enabled breakthroughs across scientific visualization [92], interactive gaming [93], autonomous driving [94], robotic control [95], and virtual production [96]. Follow-up work has further improved rendering quality [97, 98], support for dynamic content [99, 100], rendering speed [101–103], memory footprint [101, 104], and hierarchical or octree-based level-of-detail mechanisms [30, 105, 106]. Even so, the resulting systems still fall short of the specific demands of MR, including rapid refresh rates, high-resolution stereo rendering, and the tight power constraints of untethered

devices [107–110].

This chapter addresses that balance through perceptually guided level-of-detail control for 3DGS. The central observation is that rendering quality does not need to be spatially uniform to be perceptually effective. Human visual sensitivity varies strongly across the visual field and is also modulated by local content properties such as contrast and spatial structure [1–4, 57]. These properties have long motivated foveated rendering in conventional graphics, and recent work has begun to extend similar ideas to neural rendering and 3DGS [5, 54–56, 58, 59]. However, many existing approaches still rely on coarse eccentricity zones or fixed falloff rules that cannot respond smoothly to scene content, gaze-contingent changes, or changing quality budgets.

That limitation matters because MR is not simply a smaller screen version of desktop rendering. Users look around continuously, scene content can vary dramatically across the field of view, and acceptable quality depends on both display geometry and the viewer's task. In such a setting, a fixed rule such as "render the center sharply and the periphery coarsely" can be wasteful in some regions and insufficient in others. A more useful strategy is to infer, for each viewing frustum, how much detail is actually required for the current gaze, scene structure, and quality budget. That is the form of control pursued in this chapter.

The contribution of this chapter is to bring perceptual guidance directly into the level-of-detail selection process for 3DGS. Rather than assigning one fixed LOD to the entire scene or a few concentric regions, our method predicts the minimum LOD needed to satisfy a target perceptual-quality level in each viewing frustum. This is made possible by combining a hierarchical 3D Gaussian representation with a lightweight predictor distilled from a perceptual image-quality metric. The resulting framework can continuously modulate rendering complexity across the visual field while still remaining compatible with efficient Gaussian fetching and rendering.

The framework is also designed with deployment in mind. MR headsets often have limited battery and thermal headroom, so compute budgets may need to be adjusted dynamically rather than assumed fixed. To support that setting, this chapter also develops an edge-cloud collaborative

execution model in which the cloud handles the perceptual LOD decision and the edge performs the final rendering pass. In effect, the system turns a perceptual quality target into a routing policy for 3D Gaussian detail, enabling a resource-aware trade-off between network cost, local computation, and perceived image quality in edge-cloud MR streaming.

This distinction is important in practice because the same scene may need to be rendered differently depending on whether the headset is tethered, untethered, or sharing resources with other clients. The resulting design makes perceptual quality a first-class control variable that can be traded against local and remote computation without changing the underlying scene representation. In that sense, this chapter does not merely optimize a rendering kernel; it defines an operational mechanism for deciding where precision matters most in an immersive visual field.

This work is motivated by the dissertation's broader theme that graphics systems should allocate computation according to how people perceive their output, not only according to geometric or photometric fidelity. In the context of 3D scene rendering, this means that level of detail should depend on more than the nominal distance from the gaze point. It should also reflect how visible degradation is likely to be for the user under the current scene, viewing direction, and display conditions. This perspective reframes LOD control as a human-centered inference problem rather than only a geometric or systems problem.

This chapter is developed in four stages. It first introduces a continuous LOD formulation for hierarchical 3D Gaussian representations. It then explains how perceptual quality estimates are transformed into frustum-wise LOD decisions. Next, it presents an edge-cloud collaborative rendering framework together with implementation details and objective evaluation. It concludes with an MR user study that measures whether the perceptually guided approach improves subjective quality under matched compute budgets. Together, these results show that perceptual modeling can substantially improve the trade-off between rendering efficiency and perceived quality in MR.

The main contributions are a continuous and gaze-contingent LOD formulation for hierarchical

3DGS, a lightweight predictor distilled from a perceptual quality metric, an edge-cloud collaborative rendering framework that makes our method practical for untethered MR, and an MR user study with four photorealistic scenes. Our source code, trained models, and evaluation dataset are released at `https://github.com/NYU-ICL/perceptual-3dgs` to support future work on perceptually efficient immersive rendering.

4.1 A Continuous Level-of-Detail Representation for 3DGS

3DGS [23] represents a scene as a point cloud composed of semi-transparent 3D Gaussians, each carrying spatial, color, and opacity information. These Gaussians can be efficiently sampled using simple ray casting operations, which accumulate contributions along each ray in a view-dependent manner. This process ultimately enables the rendering of high-quality 2D images from arbitrary camera viewpoints:

$$I : (\mathbf{p}, \mathbf{d}; \{G_j\}) \rightarrow \mathbb{R}^{H \times W}, \text{ where } j \in \{1..n\} \tag{4.1}$$

where $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{d} \in \mathbb{R}^3$ represent the camera pose and G denotes the parameters of each of the n Gaussians.

Building on hierarchical 3DGS [30], we represent each scene as a binary tree of 3D Gaussians, where progressively smaller Gaussians are placed deeper in the hierarchy to capture increasingly fine-grained scene details. We partition the visual field of each monocular view into a dense, uniform grid of $N_h \times N_w$ viewing frusta, and modulate each viewing frustum independently using an appropriately chosen LOD value. The LOD value l is defined as a projected size threshold, measured in screen pixels, for the bounding boxes of 3D Gaussians. Within each viewing frustum, a cut is performed over the binary tree using the selected LOD value, as illustrated in the LOD fetching stage of Figure 4.2, ensuring that Gaussians on the two sides of the cut edge have their

projected sizes above and below the LOD value, respectively. The Gaussians adjacent to all cut edges are then fetched for rendering.

This representation is important for two reasons. First, it preserves the rendering efficiency of explicit Gaussian splatting while introducing a structured way to modulate detail spatially. Second, it separates representational capacity from rendering budget. The full scene can still be stored once at high fidelity, while different subsets of Gaussians are selected dynamically according to perceptual or systems constraints. This makes the representation particularly suitable for mixed reality, where the optimal rendering budget can change with gaze, hardware conditions, or power availability. The section therefore establishes the structural substrate that later sections turn into a perceptually guided and gaze-contingent rendering policy.

4.2 Mapping Perceptual Quality to 3DGS Level of Detail

Continuous LOD control becomes useful only when the system can determine how much detail is actually needed in each region of the visual field. The key question is therefore perceptual: given a viewpoint, a gaze direction, and a desired quality target, what is the coarsest Gaussian representation that still appears acceptable to the viewer?

To answer this question, the framework adopts a perceptual image-quality metric that explicitly accounts for human visual sensitivity across eccentricity. A test rendering produced with a lower-detail Gaussian subset is compared with a reference rendering at the highest fidelity, and the resulting quality estimate is expressed in just-objectionable differences. This gives a principled way to quantify how visible an LOD-induced degradation is likely to be for a human observer rather than relying solely on pixel error [4, 111]. Formally, for a chosen LOD value l, we evaluate

$$q_l = \text{fvvdp}\left[I\left(\mathbf{p}, \mathbf{d}; \left\{G_j^l\right\}\right), I\left(\mathbf{p}, \mathbf{d}; \left\{G_j^{l_{ref}}\right\}\right), e\right], \tag{4.2}$$

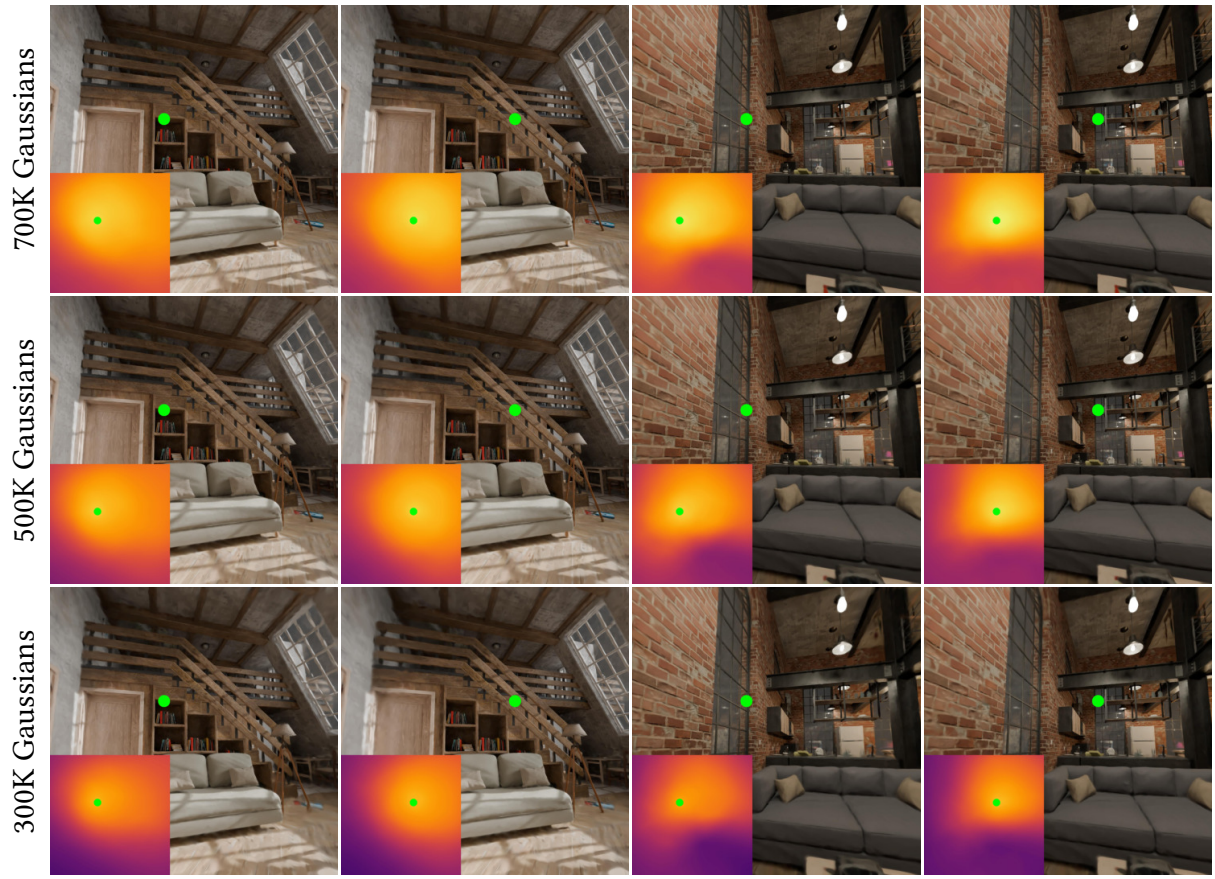


Figure 4.1: *Perceptually guided continuous LOD modulation across eye gaze positions and compute budget levels.* Each row corresponds to a different compute budget level, determined by the average number of Gaussians used for rendering, and each column corresponds to a different eye gaze position. The inset LOD maps illustrate how our method adaptively allocates fine details across the visual field based on perceptual priorities instead of using only coarse, discrete concentric zones seen in previous work. Brighter color indicates finer LOD.

where $l_{ref} = 0$ corresponds to the highest-fidelity rendering, e denotes eccentricity, and the output q_l is expressed in Just-Objectionable Difference (JOD) units [111]. JOD defines a relative quality scale that translates to observer preference probability, where a difference of 1 JOD between two conditions indicates that 75% of observers prefer one condition over the other, 2 JODs correspond to 91%, and so on. Figure 4.1 demonstrates how the predicted LOD maps vary across eye gaze positions and available Gaussian budgets.

Exhaustively evaluating every possible LOD at runtime would eliminate the performance

gains that perceptual modulation is meant to achieve. The chapter therefore trains a lightweight predictor that maps viewpoint parameters, eccentricity, and a target perceptual quality level to the minimum LOD required in each frustum. Training data are created by rendering image patches across many LOD values, scoring them with the perceptual metric, and learning the inverse relationship from desired quality to sufficient detail. For each training camera view, we randomly sample 40 LOD values from $[0, 50]$ and 20 from $[50, 100]$. Each rendered image is compared against the highest-quality reference rendering to obtain a pixel-wise JOD map, from which 100 peripheral directions and eccentricity values are sampled to crop 64×64 patches. Pooling each patch into a scalar JOD value yields tuples of the form $(\mathbf{p}, \mathbf{d}, e, q, l)$, with roughly 6K tuples contributed by every camera view. All tuples from a given scene are used to train a scene-specific predictor. The predictor is implemented as a compact MLP and trained with a mean-squared-error objective, which keeps the runtime overhead low while still capturing the nonlinearity of the quality-to-detail relationship.

This predictor makes the overall framework practical. Instead of using a fixed foveation rule, the system can infer a frustum-wise LOD map that adapts not only to gaze location but also to local scene difficulty. High-contrast or structurally complex regions may therefore retain finer detail even outside the center of gaze, while less demanding regions can be rendered more coarsely. In effect, perceptual quality is converted into an explicit resource-allocation policy over the Gaussian hierarchy.

4.3 An Edge-Cloud Collaborative Rendering Framework for 3DGS

The perceptual LOD formulation is especially useful in an edge-cloud setting, where rendering must remain responsive on power-constrained MR hardware. In the proposed framework, both

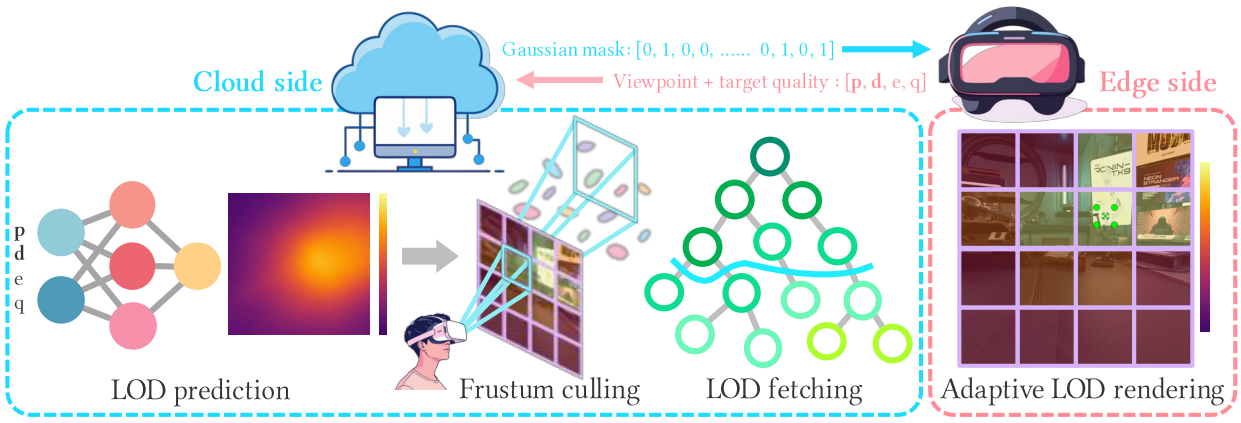


Figure 4.2: *Perceptually guided 3DGS streaming and rendering framework.* Our framework combines a level-of-detail 3DGS scene representation with perceptual quality prediction and frustum-wise LOD selection under an edge-cloud collaborative rendering setting.

the cloud and the edge store the same hierarchical Gaussian scene representation. At runtime, the edge transmits the current viewpoint and desired perceptual quality target to the cloud. The cloud then predicts the frustum-wise LOD map, traverses the hierarchy, and returns a compact binary mask or Gaussian subset to the edge for local rendering.

This division of labor preserves interactive rendering on the edge while shifting the more expensive LOD inference and tree traversal to the cloud. It also gives the system a direct way to adapt to different budget targets: changing the requested perceptual quality level changes the predicted Gaussian subset without requiring a new scene representation or retraining loop. Figure 4.2 illustrates the edge-cloud collaborative 3DGS rendering pipeline and how target perceptual quality is converted into Gaussian subset selection.

The evaluation uses four high-fidelity indoor scenes rendered with photorealistic assets and held-out viewpoints that reflect MR viewing conditions. All scenes were rendered with high-resolution textures and Blender's Cycles path tracer, with cameras uniformly placed across each scene to ensure good coverage and 100 randomly selected views held out for evaluation. Images were rendered at 1920×1920 resolution with an 18 mm full-frame equivalent focal length, corresponding to a $90^\circ \times 90^\circ$ monocular field of view that matches many commercial MR displays.

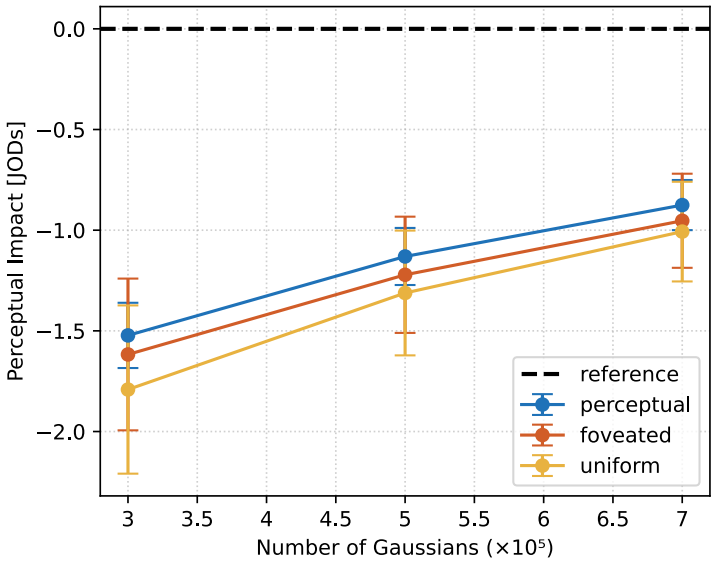


Figure 4.3: *Quantitative evaluation of different LOD modulation approaches.* Perceptual visual quality is measured in relative JOD units, where higher scores indicate better quality. Across matched Gaussian count budgets, our perceptually guided approach outperforms both the uniform and foveated baselines.

To train the LOD predictor, we construct a dedicated dataset for each scene by rendering the training views at varying LOD values and computing FovVideoVDP [4] against the highest-quality reference. For each view, we randomly sample 40 LOD values from $[0, 50]$ and 20 from $[50, 100]$, then evaluate the resulting renderings with FovVideoVDP to obtain pixel-wise JOD maps. From each JOD map, we randomly sample 100 peripheral directions $\theta \in [0°, 360°]$ and eccentricity values $e \in [0°, 45°]$, crop 64×64 patches, and pool them into scalar JOD values. Each camera view therefore yields about 6K tuples of the form $(\mathbf{p}, \mathbf{d}, e, q, l)$, and all tuples from a given scene are used to optimize a scene-specific predictor.

The main baselines are a uniform LOD strategy (**UNIFORM**) and a foveated LOD strategy (**FOVEATED**). **UNIFORM** applies a constant LOD value across the entire visual field, whereas **FOVEATED** modulates LOD according to gaze eccentricity using a conventional center-to-periphery falloff. Both are compared against the perceptual strategy under matched Gaussian-count budgets. Objective evaluation measures perceptual quality in just-objectionable differences



(a) Reference (b) Uniform (c) Foveated (d) Perceptual (Ours)

Figure 4.4: *Visual comparison of different LOD modulation approaches.* All non-reference images were rendered using a 300K Gaussian count budget. Our perceptually guided method more effectively preserves the details in regions that matter most to perceived visual quality than the uniform and foveated LOD baselines.

and uses the number of rendered Gaussians as the proxy for edge-side computation. Figure 4.4 shows representative rendering outcomes, while Figure 4.3 summarizes the aggregate perceptual quality gains over matched budgets.

The experimental setup evaluates each method on the 100 test views at three target compute budgets corresponding to average Gaussian counts of 300K, 500K, and 700K. These budgets are matched differently for each method: **UNIFORM** varies a constant LOD threshold, **FOVEATED** varies the slope of its gaze-contingent falloff, and the perceptual method varies the requested quality target q. For the gaze-contingent methods, we further render each view with three different gaze positions, producing three different LOD maps per budget.

The results show a consistent advantage for perceptually guided LOD modulation. Across all evaluated budgets, the perceptual method achieves better objective quality than both uniform and foveated alternatives. At the 700K-Gaussian budget, the perceptual method reaches -0.84 ± 0.14 JOD, and at 300K Gaussians it still maintains -1.51 ± 0.18 JOD. Relative to the reference rendering with an average of 2.56M Gaussians at 63.51 FPS, the 300K configuration provides roughly a 5× speedup and runs at 331.53 FPS. LOD prediction takes 2.64 ms on average, while the custom LOD fetching and frustum-culling kernel adds 17.76 ms. All time measurements were obtained locally on an Nvidia RTX A6000 GPU. The qualitative comparisons help explain why: rather than preserving detail only near the gaze point, the method preserves detail where degradation is most likely to matter perceptually. This leads to a more effective use of a limited Gaussian budget and a better quality-efficiency trade-off for MR rendering.

4.4 User Study: Perceptual Quality Evaluation in Mixed Reality

Objective perceptual metrics are useful, but the central claim of this chapter concerns subjective visual quality in MR. To test that claim directly, the work includes a user study in which participants compare renderings produced by uniform, foveated, and perceptually guided LOD modulation under matched computational budgets.

The study adopted a two-interval forced choice (2IFC) design on stereo-rendered stimuli shown in a head-mounted MR display with gaze tracking. Participants compare test videos against a high-quality reference while maintaining fixation at designated gaze locations. The evaluated conditions span three Gaussian budget levels and multiple fixation positions so that the study can measure both the effect of compute budget and the effect of LOD modulation strategy. We recruited 16 participants aged 21–29, used a single scene with four sampled camera views, and rendered a 72-FPS stereo video of 1,296 interpolated views for each condition.

Participants. We recruited 16 participants aged 21–29 (11 male, 5 female). All participants reported normal or corrected-to-normal vision and signed written informed consent. The study protocol was approved by an Institutional Review Board (IRB) at the host institution.

Experimental setup. We used one scene for the study. Four test camera views were sampled from that scene and their 6-degree-of-freedom poses were interpolated to obtain a smooth trajectory of $72 \times 18 = 1296$ camera views, displayed as a 72-FPS stereo video lasting 18 seconds. Translation was interpolated linearly and rotation with spherical linear interpolation. For each of the three methods, we rendered this trajectory at three compute budgets and gaze positions similar to those used in the objective evaluation, producing 3 rendering setups for uniform and 9 each for foveated and perceptual rendering. A Meta Quest Pro was used during the study.

Procedure and stimuli. The study followed a 2IFC pairwise comparison protocol, which has been used in prior evaluations of view-synthesis quality [112, 113]. At the start of each trial, participants saw a reference video and could freely switch between it and two test videos. A regular keyboard's arrow keys were used to switch among the three videos, and participants could not continue unless all three had been viewed at least once. Selections were made with the space key. A 500 ms gray screen was inserted while switching conditions to provide a clear visual separation.

Participants were seated during the study. Each trial displayed a fixation crosshair, tracked the user's gaze, and grayed out the display if the gaze deviated from the fixation point for more than 500 ms. This ensured that the evaluated distortions corresponded to the intended gaze location. Participants were instructed to keep their gaze on the crosshair, compare the test videos to the reference, and select the video with higher quality and fewer distortions. A five-trial warmup familiarized participants with the setup and stimuli before data collection began.

All stimuli were stereo rendered in Unity and shown in the headset. A naïve full pairwise design would have required 432 comparisons, which would be impractically long. Instead, the study used the active sampling ASAP algorithm [114] to select the most informative comparisons, resulting in 105 trials per participant. On average, each session took about 45 minutes including training and interpupillary-distance adjustment.

Results. As summarized in Figure 4.5, the user-study results align with the objective evaluation. At comparable Gaussian counts, participants consistently prefer the perceptually guided renderings over both the uniform and foveated baselines, with the clearest gains appearing at lower compute budgets where resource allocation matters most. The per-fixation analysis further suggests that the advantage is not restricted to a single gaze location, but persists across the tested fixation conditions; Figure 4.6 makes this trend explicit by breaking down preference as a function of fixation location. The pairwise comparison data were converted to JOD units using *pwcmp*, and

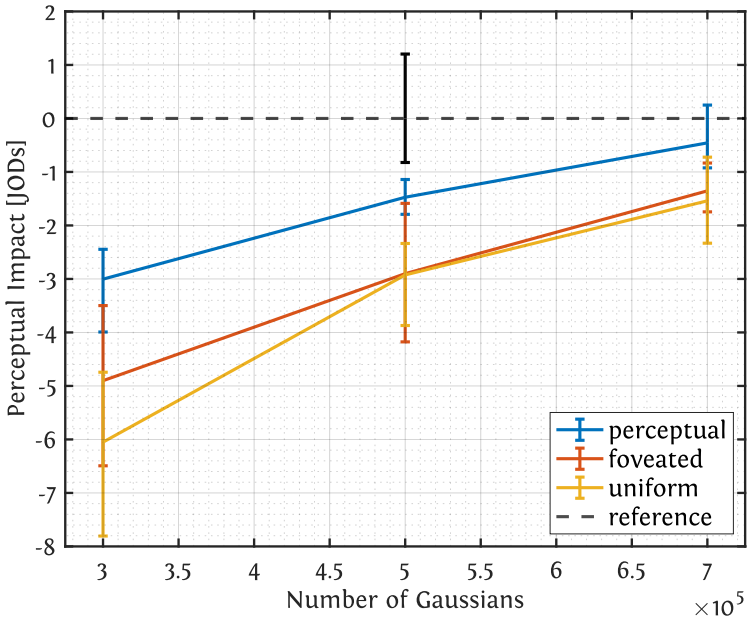


Figure 4.5: *MR user study results.* The x-axis represents the average number of Gaussians used for rendering, while the y-axis indicates perceived visual quality in JOD units. Error bars denote 95% confidence intervals. Across all Gaussian count budgets, our perceptually guided method remains preferred over the uniform and foveated baselines.

the 95% confidence intervals were computed with 1,000 bootstrap samples. An ANOVA on the scaled results shows significant effects from both the average Gaussian count ($p = 0.0032$) and the LOD modulation method ($p = 0.039$), while fixation position is not significant.

These findings are important because they validate the chapter's central premise in the intended application setting. The benefit of the approach is not merely that it optimizes a perceptual proxy more effectively, but that it produces renderings that people actually judge as higher quality under the same resource constraints. This strengthens the dissertation's broader claim that perception-aware computational allocation can improve graphics systems in ways that are both measurable and experientially meaningful.

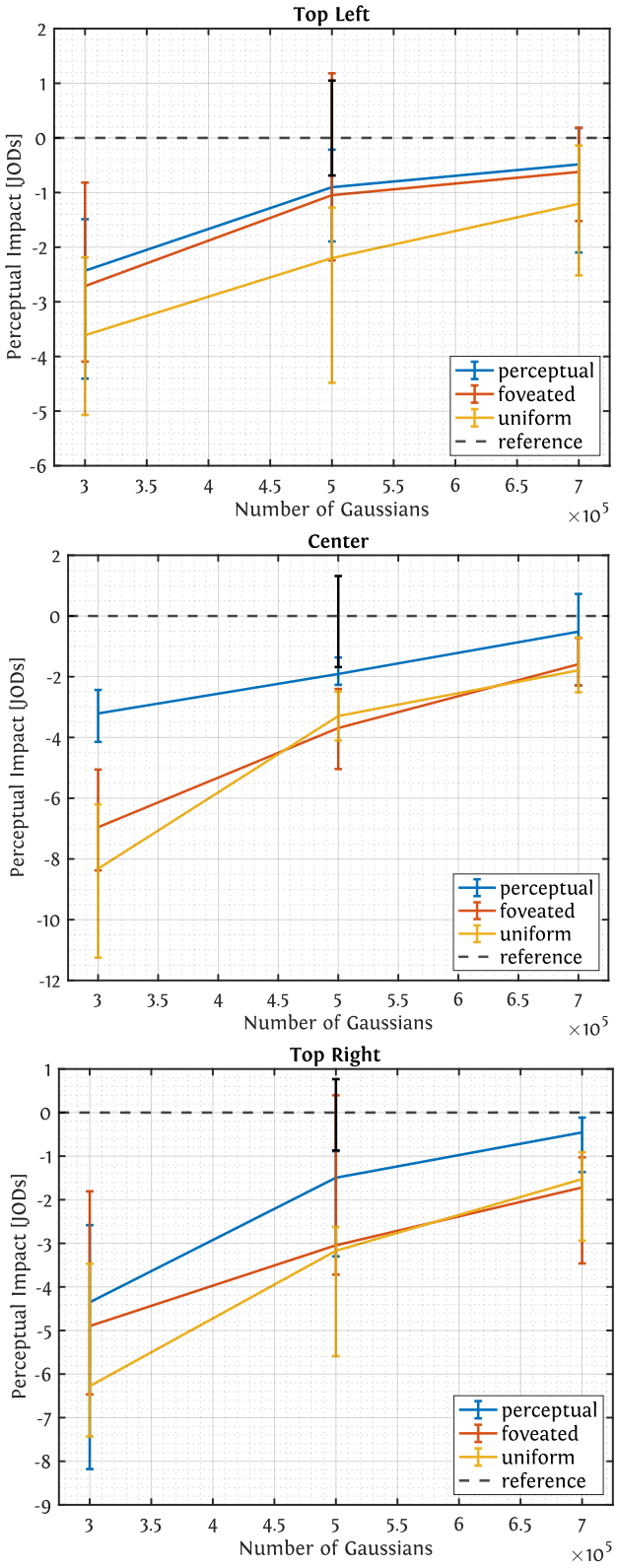


Figure 4.6: *Per-fixation MR user study results.* User study results are broken down by the three fixation locations: top left, center, and top right. The x-axis represents the average number of Gaussians used for rendering and the y-axis indicates perceived visual quality in JOD units. Error bars denote 95% confidence intervals.

4.5 Discussion

This chapter showed how perceptual quality modeling can be integrated directly into 3DGS rendering for mixed reality. By combining a hierarchical 3DGS representation, a lightweight perceptual LOD predictor, and an edge-cloud execution model, our method improves both objective and subjective quality under constrained rendering budgets. More broadly, it demonstrates that level-of-detail control in neural rendering can be framed as a human-centered allocation problem rather than only a geometric simplification problem. The central outcome is a superior quality-bandwidth trade-off for MR rendering that is validated by both FovVideoVDP and user judgments.

The current system still has important limitations. Runtime LOD inference and Gaussian-subset selection remain nontrivial even when offloaded to the cloud, and the present formulation assumes access to gaze information at runtime. In addition, the predictor is trained per scene, which limits scalability when many new scenes must be deployed quickly. These constraints are practical rather than conceptual, but they matter for deployment. MR systems are sensitive to latency, so any additional cloud-side overhead or dependence on specialized sensors must be justified by a clear perceptual gain.

Future work should therefore focus on faster cloud-side inference, reduced dependence on explicit eye tracking, and more scene-general predictors that can transfer across diverse content. The per-scene predictor used here is effective because it can specialize to the scene's geometry and appearance, but a more universal predictor would reduce engineering cost and improve reuse. Another direction is to explore more direct content-aware or task-aware quality objectives, so that the same framework can adapt not only to gaze and eccentricity but also to scene semantics, user tasks, or display characteristics. Another practical extension would be to couple the LOD predictor with a lower-cost estimate of gaze or saliency, which would make our method easier to deploy on less sensor-rich devices while preserving much of the perceptual benefit. Even with these limitations, this chapter provides a concrete example of how perceptual modeling can make

modern learned scene representations substantially more practical for immersive rendering.

There are also a few concrete engineering next steps. The current cloud-side subset decision is still expensive, and GPU-based tree traversal or other parallel hierarchy search strategies could reduce the cost of fetching LOD-specific subsets. Likewise, a gaze-free variant would be valuable for devices without eye trackers, either by extending the foveal region conservatively or by predicting gaze saliency from the scene. Finally, because the predictor is scene-specific, learning a compact scene-agnostic feature space would improve deployment efficiency when many scenes are introduced over time.

The broader dissertation-level takeaway is that perceptual allocation can be expressed at multiple levels of the graphics stack. In Image-GS, the control point is where image support is placed; here, it is where rendering detail is retained in 3D space. Both chapters point toward the same design principle: modern interactive graphics systems become more efficient and usable when they allocate computational budget based on what people are likely to perceive. For MR in particular, that means the renderer should not only preserve geometry, but also make wise decisions about where visual precision matters most.

Chapter 5

Saliency-Guided Image Generation

The recent emergence of generative artificial intelligence marks a paradigm shift for computer graphics. Diffusion models, in particular, enable the generation and editing of both photorealistic and stylized images, videos, and 3D objects with little more than a text prompt or high-level user guidance as inputs [115]. In many applications, including graphic design and advertisement, it is desirable to generate visual content that attracts a viewer's attention toward specific areas of interest. Such a human-centric control mechanism for the generation process, however, is not supported by current image diffusion models.

Popular approaches to controlled image [37] and video [116] generation include lightweight adaptation modules built around a foundation model. The adapter networks are usually conditioned by depth maps, semantic segmentation masks, body poses, or bounding boxes [36–38, 117] to control the spatial layout of an image or video. Meanwhile, an essential design factor for context creation is to direct viewers' attention to the regions of interest, such as buttons on web pages [118], products being advertised [119], or the storytelling events in a film [120]. Unlike the layout of an image, human attention is selective in nature [7, 8], concurrently influenced by high-level semantics, mid-level layouts, and low-level visual features [39, 40, 44, 121] (see examples in Figure 5.1), as well as spatial and temporal characteristics [122, 123]. Therefore, existing



(a) Saliency-guided image generation (b) A fixed layout

Figure 5.1: *Attention-controllable image generation with saliency guidance.* Given a text prompt and saliency map pair, GazeFusion generates images that not only contain the content as described by the text prompt but also attract viewers' attention toward the desired image regions as emphasized by the saliency map. As illustrated in the left panel, GazeFusion understands and exploits a diversity of factors inducing visual saliency, including low-level image features, such as color, contrast, frequency, orientation, and layout, and high-level semantic information, such as objects, texts, and faces. The right panel demonstrates that GazeFusion can flexibly manipulate viewers' visual attention within the generated images by purely adjusting the color and contrast of specific image content while precisely following the user-specified layout.

conditioning mechanisms do not adequately control a viewer's visual spatial attention.

This gap is especially important because saliency is not a single primitive that can be represented by a bounding box or a segmentation mask. What tends to attract attention depends jointly on image content, display context, and the viewing task. A text prompt can ask for a specific scene, but it does not directly specify which part of the scene should be visually emphasized. Conversely, a saliency map can encode desired attentional structure without dictating the exact object boundaries or layout. That makes saliency an attractive control signal for generative models, but also a harder one to learn because the conditioning signal is more abstract and more closely tied to human perception.

Our work aims to generate images and videos that guide viewers' visual attention toward specific regions of interest. To this end, we first analyze the spatial visual saliency [44] of a

large-scale image dataset [42]. The resulting image-saliency pairs are then leveraged to train a saliency-guided diffusion model, which we refer to as GazeFusion. We further extend our saliency conditioning from static images to temporally consistent video generation. The intent is not merely to add another spatial control channel, but to introduce a generative interface that directly reflects how people distribute attention over images.

Because eye-tracked saliency datasets are relatively small, this chapter relies on predicted saliency maps to scale training to a large corpus while still preserving the statistical structure of human attention. This choice makes it possible to learn from hundreds of thousands of image-saliency pairs rather than from a much smaller hand-collected set. The same idea then extends naturally to video, where temporal saliency prediction helps bridge the gap between viewing a single image and attending to a motion sequence. In both cases, the goal is not only to generate visually plausible content, but to generate content whose perceptual emphasis is deliberately shaped.

To evaluate the GazeFusion's effectiveness in guiding real users' visual attention, we conducted a user study with human observers naturally examining the generated images with 3,000 eye-tracked trials. A series of objective evaluations further demonstrate that our method consistently outperforms alternative control mechanisms in guiding user attention when viewing its generated visual content. We also showcase how GazeFusion can be applied in practical applications, including interactive designing saliency guidance, suppressing viewer attention in unwanted regions, and adapting generated content to various display and viewing conditions. This research serves as an important step toward human-centric control over generative models, focusing on the integration of typically implicit human design intentions within the content produced by these models. Our source code and pre-trained models are released at `https://github.com/NYU-ICL/saliency-guided-image-generation`.

The broader dissertation-level motivation is that generative graphics systems should eventually be controllable not only by what is depicted, but also by what the viewer is likely to notice first.

That perspective connects this chapter to the dissertation theme of perceptually guided graphics systems: once saliency can be modeled as an explicit conditioning signal, it becomes possible to ask more nuanced questions about attention-aware creation, rather than treating saliency as an after-the-fact evaluation metric.

To summarize, our main contributions include:

- introduce a saliency-guided image and video generation framework that incorporates the data priors of human visual attention mechanisms into the generation process;
- systematically study and validate the attention-directing performance of the proposed framework through an eye-tracked user study and a large-scale model-based saliency analysis;
- demonstrate the availability and flexibility of the proposed framework in practical applications, including interactive design of saliency guidance, viewer attention suppression in unwanted regions, and adaptive visual content generation for varied display and viewing conditions.

5.1 A Saliency-Guided Diffusion Model

Diffusion models define a Markov chain that iteratively adds Gaussian noise to samples from an empirical data distribution and gradually converts them into noisy samples from a standard Gaussian distribution. They then learn the reverse diffusion (or denoising) process to iteratively remove noise and generate new data samples from randomly sampled Gaussian noise. State-of-the-art image diffusion models are commonly trained on large-scale image datasets and are capable of synthesizing visually appealing and content-diverse images [115].

Recent efforts toward controlling these large models, such as ControlNet [37] and GLIGEN [36], have shown that it is possible to incorporate a variety of multimodal conditions into the generation

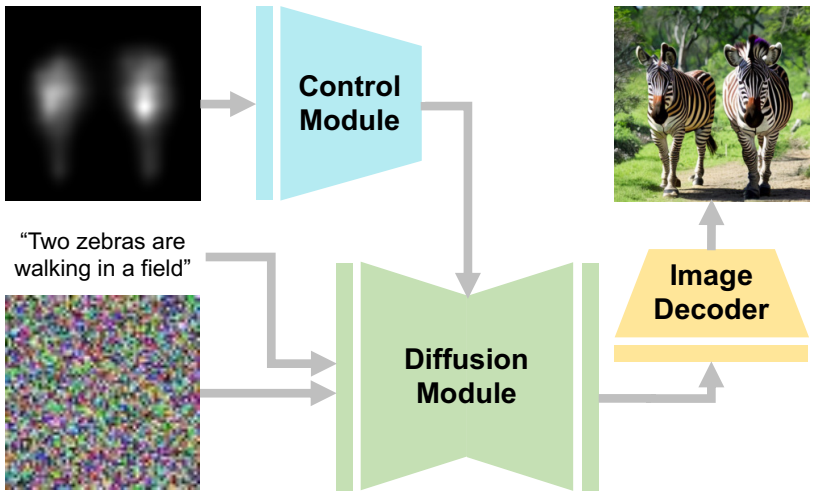


Figure 5.2: *Saliency-guided image generation with GazeFusion.* Starting from randomly sampled noisy latents, GazeFusion conditions the denoising process on user-specified saliency maps and text prompts such that the image features and semantic content in the generated images can trigger similarly distributed viewer attention.

process. This integration allows for the manipulation of semantic information and spatial layout in the content produced by these large models. Such conditional generation is typically achieved by first augmenting pre-trained image diffusion models with an adaptation module and then fine-tuning on a considerably smaller set of condition-image pairs. GazeFusion follows this line of work, but replaces explicit geometric or semantic control with an attention-distribution control signal.

To achieve attention-aware image generation, we first curate a dataset of saliency-image pairs. The scale of existing image datasets with paired eye-tracked human saliency data is commonly too limited ($< 10k$ images) to support generative learning. Therefore, we leverage the captioned MSCOCO dataset [124] for images and a learning-based saliency model, EMLNet [44], to predict their corresponding saliency maps. As visualized in Figure 5.2, we attach a ControlNet module to the encoder and middle blocks of a pre-trained Stable Diffusion (SD2.1) model [34] to inject saliency map conditions through zero convolutions and perform saliency-to-image translation. Particularly, the train, test, and unlabeled splits of MSCOCO 2017 (a total of 282k images) were

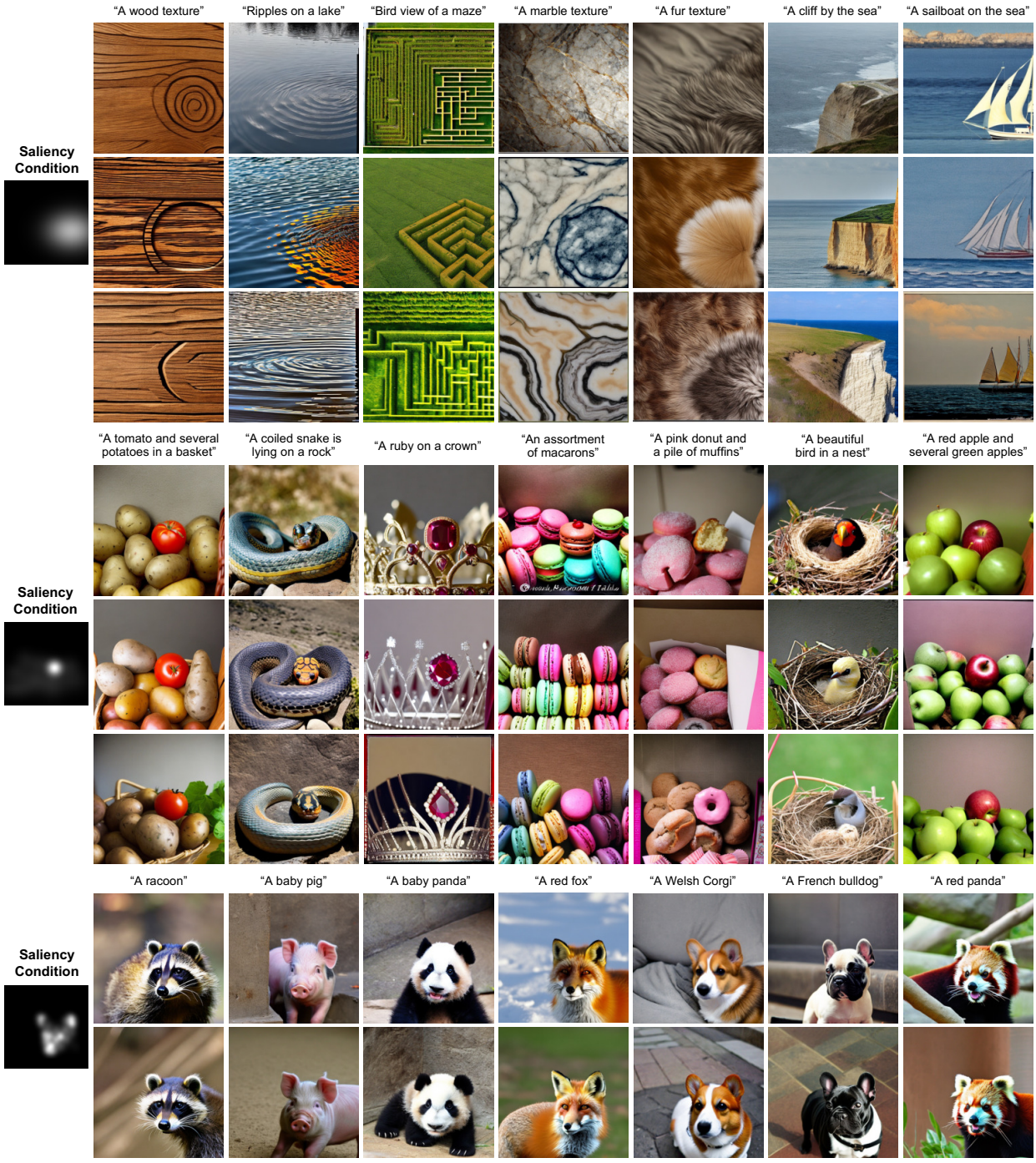


Figure 5.3: *Qualitative results of saliency-guided image generation with GazeFusion I.* GazeFusion leverages a variety of factors that affect visual saliency, such as color (e.g., the tomato and the apple scenes), frequency (e.g., the maze and the lake scenes), contrast (e.g., the fur and the marble texture scenes), orientation (e.g., the wood texture and the macaron scenes), layout (e.g., the last 3 rows), and high-level semantic information (e.g., the snake and the bird scenes).



Figure 5.4: *Qualitative results of saliency-guided image generation with GazeFusion II.* These examples demonstrate that the same saliency-conditioning mechanism can produce diverse outputs aligned with the text prompts while still following the intended attention layout.



Figure 5.5: *Ablation study on subtle changes in saliency guidance.* These examples show that visually insignificant components in the conditioning saliency map (highlighted region) can still have significant impact on the generation results. This sensitivity is useful for fine-grained design adjustments.

taken to construct our training set $\mathcal{D}$. The 5k validation images of MSCOCO 2017 were held out for evaluation. GazeFusion was initialized with the pre-trained SD2.1 model checkpoint and finetuned using an Adam optimizer (constant learning rate $1e^{-5}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$) [125] for $5e^5$ steps. Mathematically, our saliency-conditioned training process can be summarized as optimizing the denoising network in SD2.1, denoted as ϵ_θ, to predict the Gaussian noise added at each time step.

$$\mathcal{L} = \mathbb{E}_{z,t,c_t,c_s,\epsilon\sim\mathcal{N}(0,1)} \|\epsilon_\theta(z_t, t, c_t, c_s) - \epsilon\|_2^2. \tag{5.1}$$

where z, z_t, c_t, and c_s denote a sample from the latent image distribution, the corresponding noisy sample after adding Gaussian noise for t steps, the text prompt, and the conditioning saliency map, respectively. As sampled cases in Figures 5.1, 5.3 and 5.4 show, GazeFusion automatically incorporates a range of factors that lead to visual saliency, such as low-level image features (e.g. frequency, color, contrast, orientation, and layout) and high-level semantic information (e.g. objects, texts, and faces). The saliency perturbation study in Figure 5.5 further shows that even small changes in the conditioning map can shift the resulting attention layout in a meaningful way, which makes the interface useful for fine-grained design. Figure 5.3 highlights this behavior



Figure 5.6: *Qualitative results of saliency-guided video generation with GazeFusion.* By leveraging an off-the-shelf zero-shot video generation pipeline, GazeFusion can be extended to generate temporally consistent video clips with spatial-temporal saliency guidance.

across a range of prompts, while Figure 5.4 adds more examples demonstrating that the same guidance mechanism can still produce diverse outputs rather than collapsing to a rigid layout template.

5.1.1 Extension to Saliency-Guided Video Generation

We further explore the possibility of applying GazeFusion to saliency-guided video generation. Existing zero-shot (i.e. training-free) video generation pipelines extend the generation of individual frames to video clips by enforcing the temporal consistency across adjacent frames [116, 126]. Therefore, they are also adaptable to controlling modules with conditions such as body poses and edges.

However, a unique challenge for human perception is the domain gap between viewing individual frames with extended duration (spatial-only saliency, $\mathbf{S} = S_{\{1,2,\ldots,T\}}$, as used in Section 5.1) and watching them temporally composed as a video sequence (spatio-temporal saliency $\mathbf{V} = V_{\{1,2,\ldots,T\}}$) [122]. Here, S and V indicate the saliency maps on individual frames and videos of the same image sequence. This is due to various temporally induced factors, such as camera and object motions, influencing selective attention. Therefore, although it is possible to directly apply our control module to perform per-frame saliency-guided image generation $\mathbf{S}$ to compose

a temporally consistent video, the resulting user attention will differ from the target $\mathbf{V}$. To address this domain gap, we adopt the TASED-Net video saliency predictor [127] to approximate $\mathbf{V}$ and provide temporally consistent saliency sequences. Figure 5.6 shows two sequences of GazeFusion-generated video frames.

5.2 User Study: Evaluating Visual Attention in Generated Images

The central question of this chapter is whether saliency conditioning changes real human viewing behavior rather than only changing the output of a saliency predictor. To evaluate this directly, we conduct an eye-tracked user study in which participants freely view generated images while their gaze trajectories are recorded, yielding 3,000 eye-tracked trials in total.

The stimuli are generated from held-out prompts under three conditions: a text-only diffusion baseline, a bounding-box-guided baseline, and the saliency-guided model. Participants view the images sequentially while an eye tracker records their fixations. From these fixation traces, empirical saliency maps are computed and compared against the intended attention distributions. This design makes it possible to measure whether the generated images actually guide users' attention to the desired regions.

Participants. Twenty adults participated in the study (ages 23–57, 9 female). All of them have normal or corrected-to-normal vision, no history of visual deficiency, and no color blindness. None of them were aware of the hypothesis, the research, or the number of conditions. The research protocol was approved by the Institutional Review Board (IRB) at the host institution, and all subjects gave informed consent prior to the study.

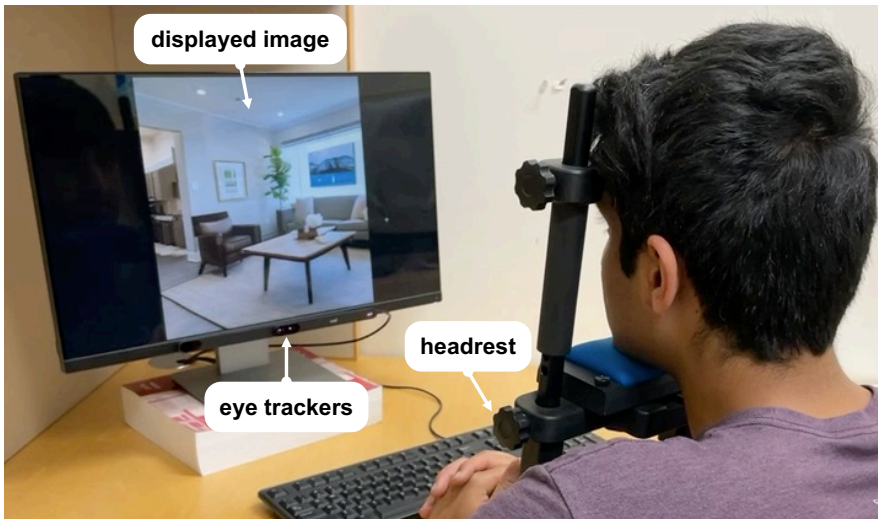


Figure 5.7: *Eye-tracked user study setup.* A Tobii Pro Spark eye tracker was used to record participants' eye-gaze directions while they viewed a sequence of generated images under controlled display conditions.

Setup and procedures. During the study, subjects remained seated in a well-lit room and viewed a 24-inch Dell monitor (Model No. S2415H, resolution 1920×1080, luminance 250 cd/m^2) binocularly from an SR Research headrest positioned 60 cm away. The effective field of view and resolution were $46^\circ \times 26.8^\circ$ and 40 pixels per degree of visual angle. A Tobii Pro Spark eye tracker was mounted to the bottom of the monitor to record their eye gaze at 60 FPS. A 5-point eye-tracking calibration was performed before each session began. Figure 5.7 shows the experimental setup of our user study.

Stimuli. We first sampled 50 images from the held-out validation set of MSCOCO 2017, where half of them have humans/animals as the main content and the rest show close-up shots of objects or nature/city scenes. These selected images were annotated using the BLIP-2 Image2Text model [87] for text prompt conditioning. Image saliency maps were extracted using the EML-Net saliency model [44] for visual saliency conditioning. We then fed the obtained paired text prompts and saliency maps to our saliency-guided model to generate 50 images of 512×512 resolution as the visual stimuli for the user study. The hypothesis is that these generated images should direct

viewers' attention toward the intended regions as depicted by the saliency maps while observing the text prompts and maintaining non-degraded image quality/diversity.

Conditions. For attention-directing performance comparison, we included two baseline conditions, the text-only model (**TEXT**), Stable Diffusion v2.1 [34], and the bounding-box-guided model (**BBOX**), GLIGEN [36]. All three conditions share the same input text prompts to control what visual content should be generated. Additionally, **BBOX** takes in text-annotated bounding boxes predicted by the Grounding DINO model [128], and **OURS** (i.e. GazeFusion) takes in saliency maps predicted by the EML-Net saliency model. Similar to **OURS**, 50 images were generated for **TEXT** and **BBOX**, respectively.

Task and duration. The total of 150 images generated for the three conditions was shuffled in random order and sequentially displayed to each subject, with a 5-second duration for each image and a 1-second pause between consecutive images. The complete study, including hardware setup/calibration, pre-study instructions, and breaks, took about 30 minutes per subject. Throughout the study, all subjects were instructed to keep their head stationary on the headrest and freely explore the displayed images by shifting their eye gaze. Their eye gaze patterns on each image were recorded to compute the corresponding empirical saliency map.

Metrics. To quantitatively evaluate each model's performance in directing users' attention to intended image regions, we adopted five saliency similarity metrics from the MIT/Tuebingen

Table 5.1: *Eye-tracked user study results.* GazeFusion outperforms the two baseline methods in directing viewers' visual attention toward the user-intended image regions. ↑/↓ indicates that higher/lower values are better.

	AUC ↑	NSS ↑	CC ↑	KL ↓	SIM ↑
Text-only	0.65	0.71	0.21	4.75	0.34
Bounding-box	0.78	1.21	0.47	2.67	0.48
Saliency-guided	**0.84**	**1.82**	**0.78**	**0.79**	**0.66**

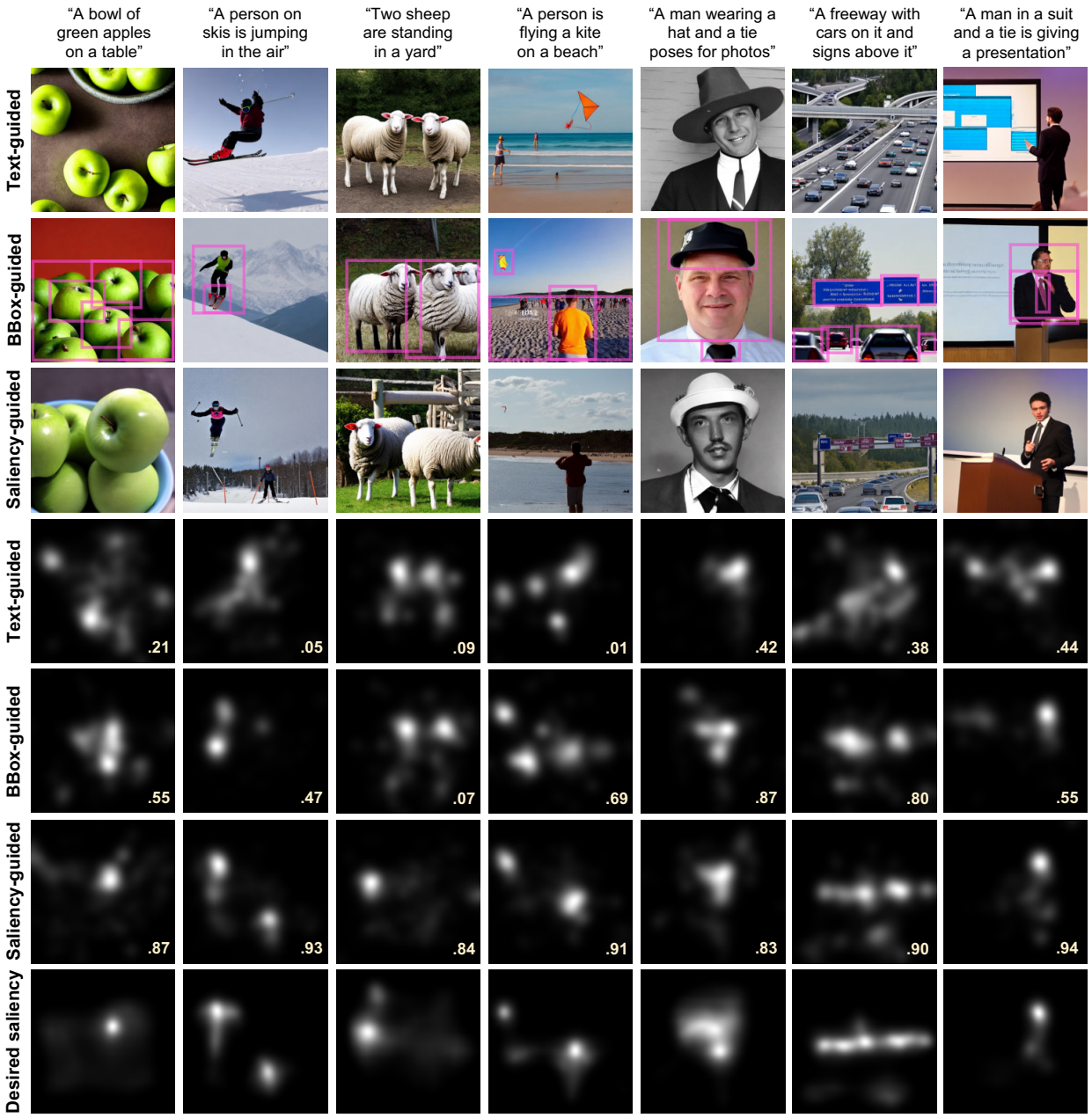


Figure 5.8: *Eye-tracked user study results.* Rows 1–3 show the generated images, rows 4–6 show the empirical saliency maps obtained by aggregating 20 users' eye gaze data, and row 7 shows the input conditioning saliency maps (i.e., the desired viewer attention distributions). The conditioning bounding boxes for **BBOX** are shown as pink overlays. The number associated with each empirical saliency map shows its correlation with the desired attention distribution. Compared to the two baseline methods, the images generated by GazeFusion (**OURS**) not only contain the exact content as described by the text prompts but also trigger viewer attention distributions that align better with the intended ones.

Saliency Benchmark [129]: Area Under ROC Curve (AUC) [8], Normalized Scanpath Saliency (NSS) [130], Correlation Coefficient (CC), Kullback–Leibler Divergence (KL), and Histogram Intersection (SIM). Notably, AUC and NSS take a saliency map and a sequence of eye fixations as inputs, while computing CC, KL, and SIM requires two saliency maps. To convert our collected eye-tracking data into empirical saliency maps, we followed the same post-processing procedures described in [9].

Results and discussion. As shown in Table 5.1, our saliency-guided approach consistently outperforms the two baselines in controlling and directing users' visual attention toward intended image regions across all five metrics. Figure 5.8 shows seven groups of generated images used in our user study, their corresponding text prompts and empirical saliency maps, as well as the input conditioning saliency maps. As can be observed, GazeFusion not only produces the content as depicted by the text prompts but also achieves viewer attention distributions that align well with the intended ones. These results strongly validate the attention-directing capability of GazeFusion and provide direct human evidence that saliency guidance is affecting actual gaze behavior rather than merely fooling an automatic saliency model.

5.3 Model-Based Saliency Analysis

The user study establishes that saliency-guided generation affects real viewers, but its scale is necessarily limited. To complement that experiment, this section evaluates the model on large sets of generated images and videos using off-the-shelf saliency predictors as analysis tools. The purpose is not to replace human evaluation, but to examine the robustness and generality of the learned control mechanism over many more prompts and attention patterns than are feasible in a laboratory study.

Image generation. Similar to the data preparation procedures in Section 5.2, we first computed the BLIP-2 captions, spatial saliency maps, and text-annotated bounding boxes on the held-out MSCOCO data (5K images) to condition the image generation process. Next, we applied GazeFusion to generate 5K images for **OURS** using the 5K paired captions and saliency maps. Using their respective model and input conditions, 5K images were also generated for **BBOX** and **TEXT**. Finally, we measured the discrepancy between the saliency maps used as input conditions (i.e., the desired attention distributions) and the saliency maps of the generated images per the EML-Net image saliency predictor (i.e., the achieved attention distributions).

Video generation. To generate a large set of video clips with GazeFusion, we first sampled 1K videos from the WebVid-10M dataset [131], trimmed them down to 4-second clips, and uniformized the frame rate to 8 FPS. Next, we took the TASED-Net video saliency predictor to extract the spatial-temporal saliency map sequence from each processed video clip. Leveraging the Text2Video-Zero pipeline [126], we then applied our GazeFusion model to generate 1K video clips of 4 seconds and 8 FPS based on the previously extracted saliency map sequences. Finally, similar to generated images, we measured the frame-wise discrepancy between the desired and achieved attention distributions for all generated video clips.

Results and discussion. In this large-scale evaluation, since we take advantage of ML-based saliency models to directly extract saliency maps from generated images and videos as proxies

Table 5.2: *Model-based saliency analysis.* The images and videos generated by GazeFusion achieve more aligned visual attention distributions with the input saliency maps according to EML-Net and TASED-Net. Note that the video results of **BBOX** are unavailable due to GLIGEN's incompatibility with the Text2Video-Zero pipeline.

	Image			Video		
	CC ↑	KL ↓	SIM ↑	CC ↑	KL ↓	SIM ↑
Text-only	0.22	7.27	0.35	0.21	5.68	0.34
Bounding-box	0.54	3.97	0.54	N/A	N/A	N/A
Saliency-guided	**0.84**	**0.75**	**0.75**	**0.79**	**0.85**	**0.68**

for ground truth and do not have access to eye fixation data, only CC, KL, and SIM are feasible and thus adopted. As shown in Table 5.2, GazeFusion significantly outperforms the two baselines across all three metrics on both image and video generation. The image examples in Figures 5.3 and 5.4 and the video examples in Figure 5.6 help interpret these metrics qualitatively: the model maintains prompt-consistent content while reshaping the attention layout in a controlled way. The analysis above not only validates the extendability of GazeFusion to saliency-guided video generation but also demonstrates its robustness and generality on a wide range of text prompts and attention distributions. These results lay the foundation for the practical applications discussed in Section 5.4.

Ablation Study on Image Quality and Diversity. To verify that our saliency conditioning does not result in any performance drop relative to the base SD2.1 model, we adopted two image quality and diversity metrics, inception score (IS, higher is better) and Fréchet inception distance (FID, lower is better), to evaluate how GazeFusion-generated images compare with those generated by the base model. To this end, we reused the two sets of images generated for the conditions **TEXT** (the base model with text guidance only) and **OURS** (our GazeFusion model). Note that both sets of images were generated using the same text prompts and are thus fair to compare. GazeFusion achieved FID = 16.43 and IS = 36.05, while SD2.1 achieved FID = 22.29 and IS = 34.81, indicating that our saliency-guided generation even improves upon the base model in terms of image quality and diversity. Such improvements are likely due to the similar image distribution of our training and evaluation sets, which were both taken from MSCOCO 2017. Our GazeFusion model was carefully fine-tuned on these photo-based realistic images to learn saliency guidance while SD2.1 was not specifically optimized to generate such images. These results strongly validate that we can effectively incorporate the data priors of human attentional behavior into the diffusion process without degrading the generation performance of the base diffusion model.

5.4 Interactive Design of Saliency Guidance

Saliency maps are expressive, but they can also be difficult to author manually. A designer may know roughly where attention should be concentrated without knowing exactly which saliency layout is compatible with a given prompt. Unlike other non-semantic controlling inputs, such as body poses, line sketches, and depth maps, prior research has observed a strong correlation between the content of an image and its incurred visual saliency [132]. Consequently, when the text prompt is incompatible with the saliency guidance provided by a designer, GazeFusion may generate unnaturally looking images with artifacts (e.g., distorted objects and human bodies, see the top row of Figure 5.9), as we have repeatedly observed in our experiments.

To this end, we developed an interactive saliency authoring system to ease designers' trial-and-error efforts in crafting prompt-compatible saliency guidance. Specifically, a user provides a text prompt p_{in} and generates a tentative saliency map by clicking over a black canvas. Each click creates a bivariate Gaussian $G_i(w_i, \mu_i, \Sigma_i)$ that can be further adjusted to compose the desired saliency guidance. Here, w_i, μ_i, and Σ_i denote the Gaussian's weight, mean, and covariance matrix. The resulting saliency map is represented as a Gaussian mixture (GM):

$$S_{\text{in}} \coloneqq \sum_{i=1}^{N_{\text{in}}} G_i\left(w_i, \mu_i, \Sigma_i\right), \tag{5.2}$$

Next, from our 282k text-saliency paired dataset $\mathcal{D}$ (Section 5.1), we search for the text prompt closest to the user-provided one within the CLIP-embedded language space [133], and retrieve its corresponding saliency map (i.e. the reference saliency map) as a starting point for optimization:

$$p_{\text{ref}}, S_{\text{ref}} = \underset{\{p,S\}\in\mathcal{D}}{\arg\min} \left\| \mathcal{E}(p) - \mathcal{E}(p_{\text{in}}) \right\|_2, \tag{5.3}$$

Here, p_{ref} and S_{ref} denote the retrieved reference text-saliency pair; $\mathcal{E}$ denotes the CLIP text encoder.

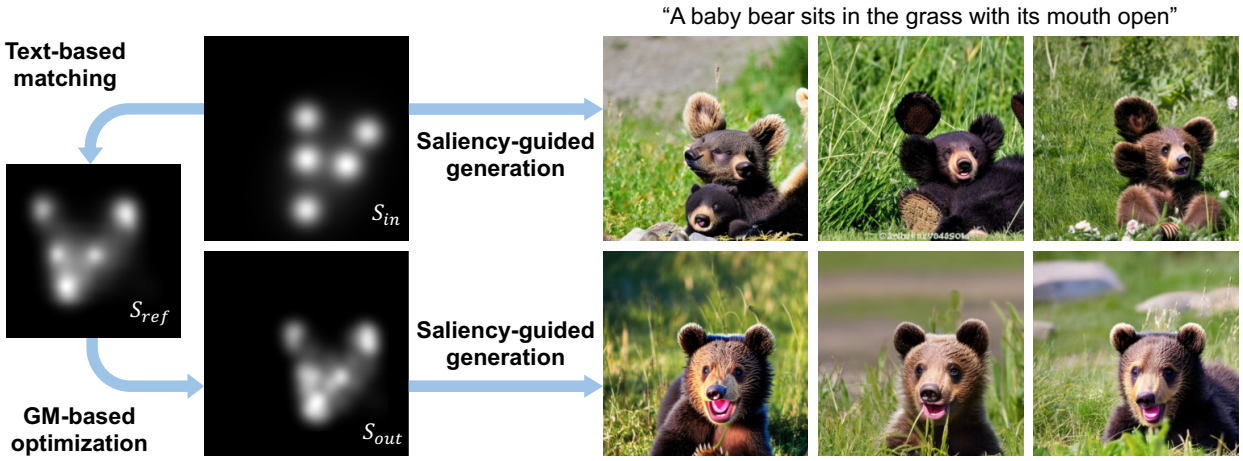


Figure 5.9: *Interactive design of saliency guidance.* Our interactive saliency design system eases users' burden of crafting high-quality, prompt-compatible saliency maps by proposing corrections when necessary. The top row shows a tentative user-created saliency map and its corresponding GazeFusion-generated images; the bottom row shows the artifact-free generation results obtained after optimizing the saliency guidance.

Finally, we convert S_{ref} into a Gaussian mixture $\hat{S}_{\text{ref}}$ and optimize an image-space transformation matrix $T \in \mathbb{R}^{2\times 2}$, which is composed of a translation vector $t \in \mathbb{R}^2$, a rotation angle $r \in [0, \pi]$, and a scaling vector $s \in \mathbb{R}^2_+$, to approximate the user-intended saliency guidance S_{in} as follows:

$$\hat{S}_{\text{ref}} \coloneqq \sum_{j=1}^{N_{\text{out}}} G_j\left(w_j, \mu_j, \Sigma_j\right), \tag{5.4}$$

$$\mathcal{L} = \left\| T \cdot \hat{S}_{\text{ref}} - S_{\text{in}} \right\|_2, \quad T_{\text{out}} = \arg\min_{T\in\mathbb{R}^{2\times 2}} \mathcal{L}, \quad S_{\text{out}} = T \cdot \hat{S}_{\text{ref}}, \tag{5.5}$$

Here, $\mathcal{L}$, T_{out}, and S_{out} denote the objective function, optimized transformation matrix, and corrected saliency map, respectively. Figure 5.9 shows an example where the initial user-specified saliency map S_{in} is incompatible with the accompanying text prompt due to inappropriate saliency intensity and layout. In the corresponding generated images, the baby bear exhibits a severely disfigured face and incorrect body anatomy. In comparison, the images generated based on the system-corrected saliency map S_{out} not only present a realistically looking baby bear without artifacts but also maintain the user-intended saliency layout. Notably, the whole generation process only involves a few mouse clicks from the user, with the correction automatically performed.



Figure 5.10: *Attention suppression with GazeFusion.* GazeFusion is capable of completely shifting viewers' attention away from specific image regions (absolute suppression) or reducing viewers' attention in less important regions (relative suppression). Brighter/darker colors indicate higher/lower saliency (or desired attention levels).

Using this interactive design interface, users can easily create complicated custom saliency maps to perform attention-controlled image generation; Figures 5.1 and 5.3 show several examples of such manually authored guidance.

5.5 Image Generation with Attention Suppression

Attention control is useful not only for making regions more salient, but also for reducing the prominence of regions that should not dominate the viewer's gaze. This section explores two forms of attention suppression using GazeFusion: 1) absolute suppression, where viewer attention is completely shifted away from the target regions; 2) relative suppression, where viewer attention is adaptively reduced in the less important regions compared to the more important ones, though not entirely removed.

As demonstrated in Figure 5.10, we crafted two pairs of saliency maps and fed them to

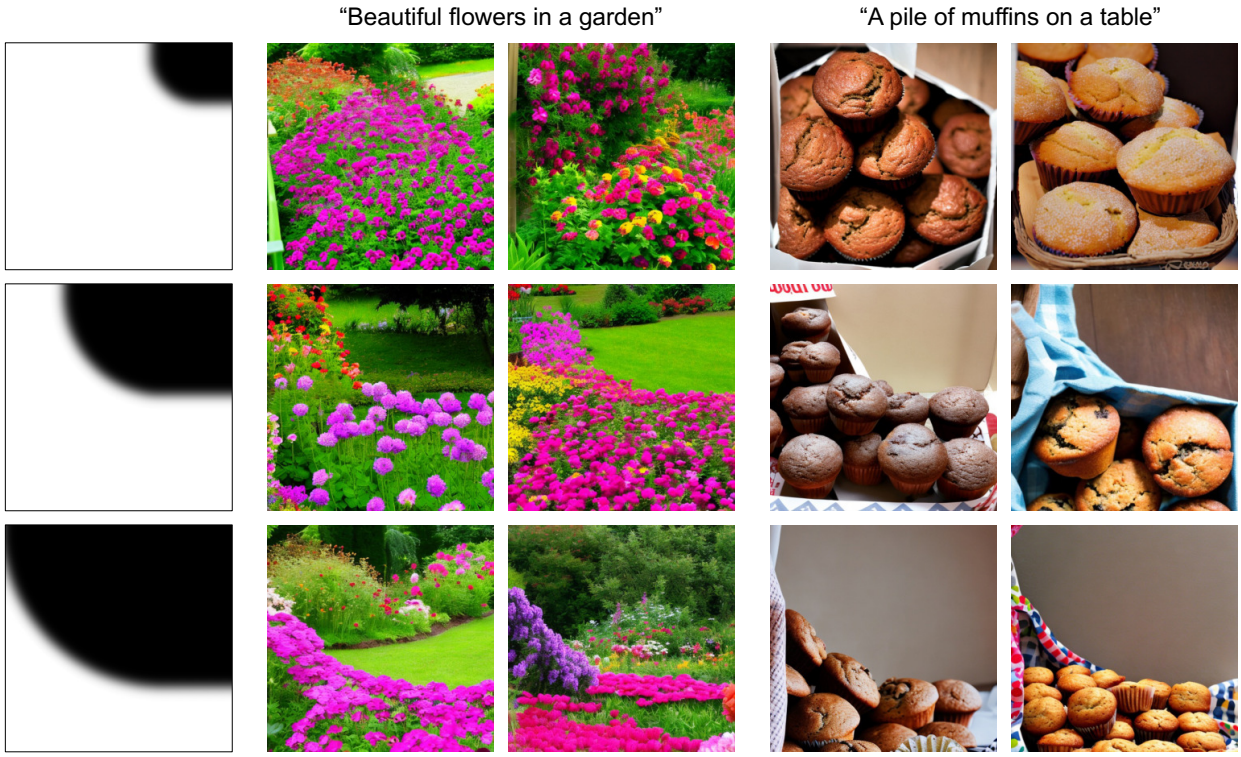


Figure 5.11: *Absolute attention suppression with GazeFusion.* Additional examples in which GazeFusion suppresses a target region almost entirely, causing the generated content to relocate elsewhere in the image while preserving the overall semantics of the prompt.

GazeFusion. In each pair, one saliency map is designed to demonstrate a type of attention suppression (**TEST**), and the other is crafted to be the same except in the suppressed regions (**CONTROL**). For 1) *absolute suppression*: **TEST** is strongly salient everywhere except for an oval-shaped region. The **TEST**-guided image features several arrays of donuts except for the suppressed region, while the **CONTROL**-guided image shows donuts everywhere. For 2) *relative suppression*, the text prompt describes a two-person chatting scenario. **CONTROL** shows two separate bright regions that are equally salient, and **TEST** shows the same two bright regions but with one of them being considerably brighter than the other. While the **CONTROL**-guided image depicts two persons at the two bright regions, the **TEST**-guided image, interestingly, not only positions the two persons to the two bright regions but also makes the person at the brighter one face the camera and the other person turn his back to the camera and out of focus. Additional

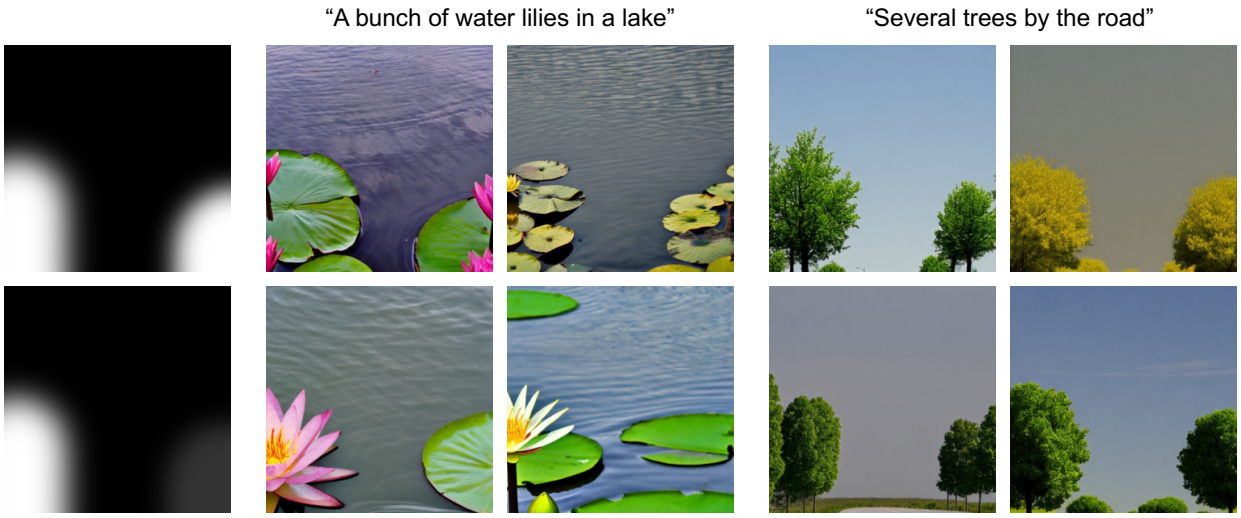


Figure 5.12: *Relative attention suppression with GazeFusion.* Additional examples in which GazeFusion reduces the prominence of one region relative to another, shifting viewer attention without fully removing the suppressed content from the scene.

examples are shown in Figures 5.11 and 5.12.

These examples illustrate that the saliency-guided model can accomplish both kinds of suppression. In some cases it removes salient content from a target region altogether; in others it preserves the content while modifying viewpoint, contrast, focus, or local prominence so that viewers are more likely to attend elsewhere first. This demonstrates that saliency guidance is not limited to simple attraction toward bright regions, but can support richer manipulations of visual hierarchy. Figures 5.11 and 5.12 further show that the same mechanism generalizes across prompts and compositions.

5.6 Display-Adaptive Image Generation

Images are increasingly viewed across displays with very different sizes, aspect ratios, and viewing distances. A composition that works well on a compact laptop screen may place critical content too far into the periphery on a wide display, making it harder to consume comfortably. Saliency-guided generation provides a way to adapt content to these changing viewing conditions before

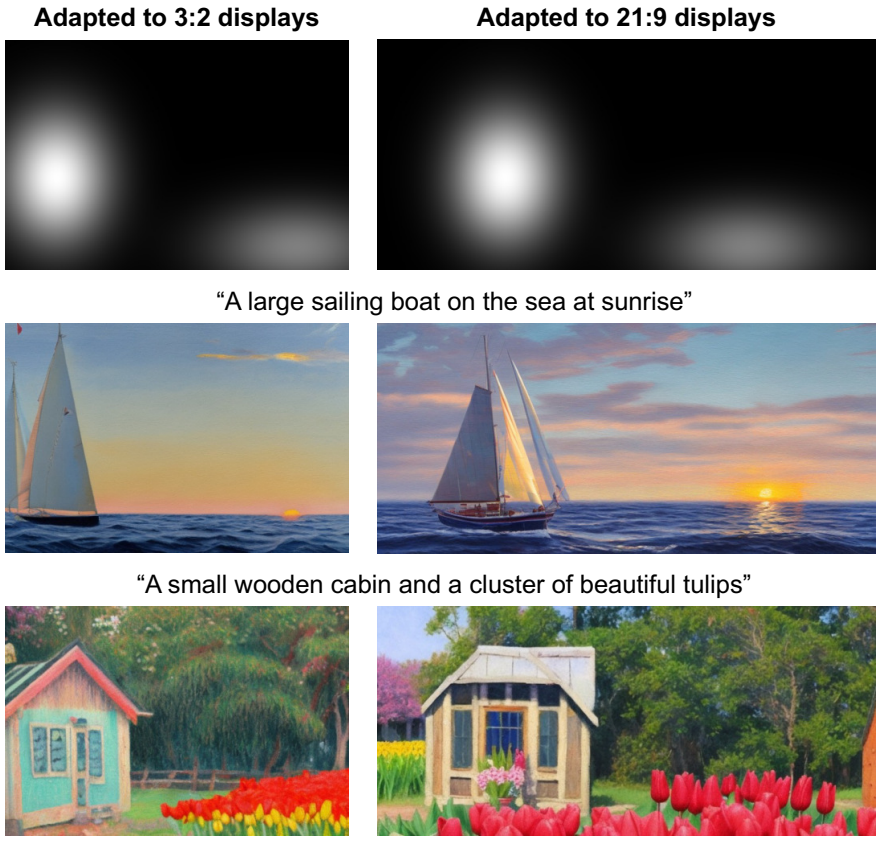


Figure 5.13: *Display-adaptive image generation.* The left/right column shows the target saliency distribution and the corresponding generated images for a narrow/wide field display. Notably, the spatial guidance for the wide-field display is designed to ensure that the main image content remains in a comfortable FoV for viewers. This can help improve content visibility and reduce excessive eye/head movements, countering the low visual acuity of peripheral vision.

the image is shown. Figure 5.13 shows how the same prompt can be steered toward different attention layouts to better match narrow and wide viewing conditions.

The increasing size of the latest displays has widened the field of view (FoV) they deliver. However, most visual content is created without considering the exact environment in which it will be displayed. In practice, this may lead to unpredictable and compromised viewing experiences. For instance, when viewing an image on a 13-inch MacBook, most of the image content will be located within the central part of the viewers' visual field; when viewing the same image on a 34-inch curved monitor with a 21:9 aspect ratio, however, a non-trivial part of the image will

be displayed in the viewers' low-acuity peripheral visual field [134, 135] and not be properly consumed. In addition, viewers also have to shift their eye gaze or rotate their necks more frequently, which may lead to ergonomic problems [136, 137].

Leveraging GazeFusion, artists and designers can readily customize their desired saliency guidance and generate graphics that are perceptually optimized to varying displays, including different aspect ratios, sizes, viewing distances, and viewing angles. This feature can help keep image content of particular interest and importance within a reasonable FoV when viewers move from one display to another. As demonstrated in Figure 5.13, GazeFusion-based adaptive generation shifts important image content toward the optimal viewing region for the display in use through saliency conditioning, effectively improving content visibility and reducing the need for frequent eye or head movements caused by peripheral viewing.

5.7 Discussion

This chapter showed that saliency maps can serve as an effective control interface for image generation. The resulting model guides human attention more reliably than text-only and bounding-box-guided baselines, remains competitive in image quality and diversity, and supports practical design workflows such as interactive saliency authoring, attention suppression, and display adaptation. More broadly, it demonstrates that generative models can be controlled in terms of perceptual outcomes rather than only semantic content and layout.

Several limitations remain. Saliency is a composite phenomenon influenced by low-level features, semantics, task context, and temporal factors, but the current model treats it as a single conditioning channel. A more structured formulation could separate different causes of attention and provide more interpretable control. This matters because the same visual saliency pattern can arise for different reasons, and those reasons may imply different design actions. The present chapter also uses predicted saliency maps for training and large-scale evaluation, which makes it

scalable, but still places a ceiling on how directly GazeFusion can model human viewing behavior in uncommon or highly task-specific settings.

In addition, the extension to video generation is currently evaluated only through model-based analysis rather than human eye tracking. Future work should therefore investigate richer decompositions of saliency and more direct human evaluation of temporally guided generation. A temporal model that explicitly reasons about motion, scene dynamics, and gaze shifts would likely make the video setting more faithful to real viewing behavior. Another promising direction is to connect saliency guidance with downstream editing or generation tasks beyond the ones demonstrated here, such as interface design, educational media, or attention-aware storyboarding. The interactive design application also reveals a practical limitation: when the supplied text prompt and desired saliency structure are strongly mismatched, the generated images may become less natural, which suggests that future work should jointly reason about saliency, prompt compatibility, and visual quality. Even so, this chapter provides a concrete step toward viewer-aware generative graphics systems.

A more fine-grained future treatment of saliency itself is also desired. Rather than treating all attention mechanisms as a single channel, one could integrate multiple saliency models that separate low-level image features, mid-level structures, and semantic labels under a probabilistic framework. In addition, eye-tracking-free attention assessment via crowdsourcing is a potential path toward scaling the video setting, since direct video gaze annotation is expensive. Finally, the interactive saliency-design workflow could be extended to jointly model prompt compatibility, saliency structure, and image quality, which would make the correction step more principled when a manually authored saliency map is difficult to reconcile with the text prompt.

Within the dissertation, this chapter plays a distinct role: it moves from representations and rendering to the generation of new content that is itself conditioned on human perceptual behavior. That shift matters because it shows that perceptual modeling is not only a way to optimize existing graphics pipelines, but also a way to shape the content those pipelines produce in the first place.

It also links naturally back to the preceding chapters: Chapters 3 and 4 allocate representation and rendering effort according to content and perceptual sensitivity, whereas GazeFusion allocates generative capacity according to anticipated visual attention.

Chapter 6

Visual-Auditory Perception and Reaction

Visual targets in immersive environments rarely appear in isolation. In games, simulators, and mixed-reality interfaces, users continuously integrate what they see with what they hear while deciding where to look next and how quickly to respond. That coupling matters because the success of many interactive systems depends not only on whether critical events are rendered, but also on whether those events can be perceived and acted upon in time. A warning light, a sound cue, or a jointly presented visual-auditory signal may all point to the same target, yet the time required to interpret those cues and initiate an eye movement can differ substantially across sensory modality, target eccentricity, and stimulus quality.

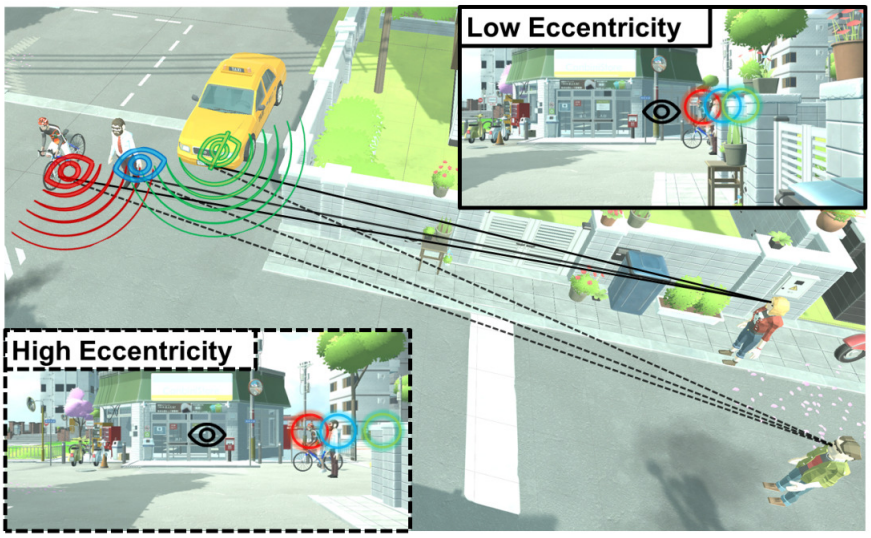
Previous work in graphics and human-computer interaction has established that gaze behavior is tightly related to perceptual bandwidth and user performance in immersive systems [6, 9, 54]. This line of research has motivated perceptually guided rendering, gaze-aware transmission, and user-centered display design. At the same time, reaction-latency studies in psychology and neuroscience have shown that both vision and audition contribute to target acquisition, and that their relative influence changes with stimulus quality and spatial configuration [10, 11, 138]. What remains missing is an operational model that connects these psychophysical observations to immersive graphics practice. If a target appears briefly in the periphery, should it be strengthened

visually, supplemented with audio, or relocated altogether? If a video is accelerated or an interface is densely populated, how likely is it that users will miss task-relevant events?

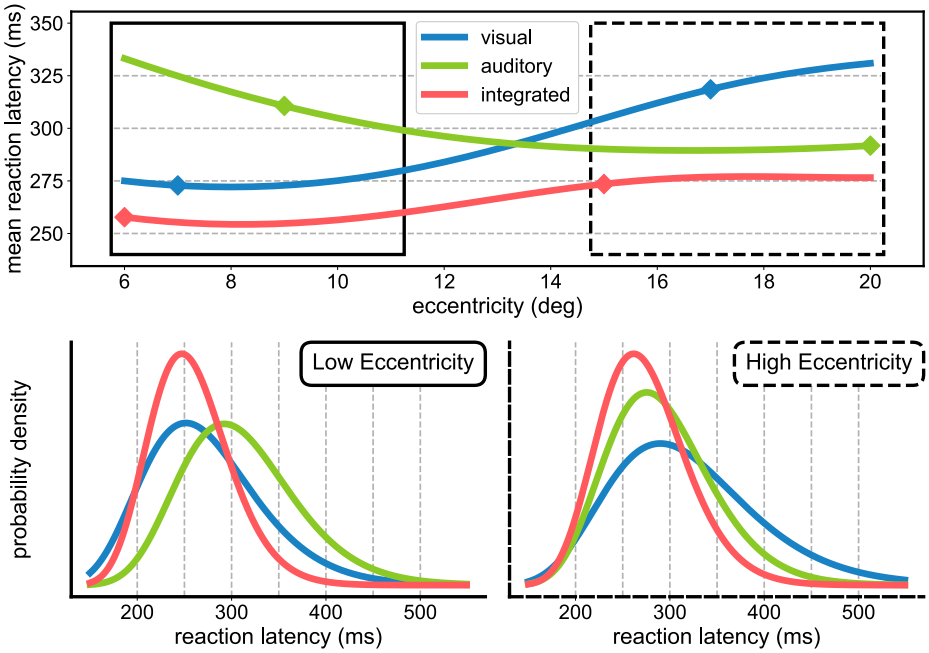
This chapter addresses that gap by studying visual-auditory reaction latency as a concrete human-centered performance signal for immersive environments. We first measure target-reaching latency in VR using saccadic eye movements as a precise and accessible behavioral marker. The pilot study spans visual-only, audio-only, and integrated visual-auditory stimuli across multiple eccentricities, contrast levels, and sound volumes. Figure 6.1 previews the central motivation: the same set of targets may produce markedly different reaction-time distributions depending on where they appear and which sensory cues are available. These differences suggest that reaction latency can serve as a useful intermediate quantity between low-level stimulus design and high-level interface performance.

The technical goal of this chapter is not to claim a complete neurological account of multisensory integration. Instead, we seek a compact, executable model that is faithful to measured human behavior and useful for downstream graphics applications. To that end, we build on drift-diffusion formulations of decision latency [139–141]. This framework provides a natural way to represent both the stochastic variability of human responses and the systematic effects of stimulus properties. We adapt it to jointly account for visual contrast, auditory volume, and eccentricity, and we fit its open parameters from the VR psychophysical data collected in our studies. The resulting model predicts not only a mean latency but a full reaction-time distribution, which is especially valuable for estimating the probability that a user misses a short-lived event.

This work also connects closely to the dissertation's broader theme of human-centered multimodal graphics. The previous chapters focused primarily on visual representation, perceptual rendering, and saliency-guided generation. Here we move from static or rendered content quality to time-sensitive multimodal behavior. Vision still matters, but it now interacts explicitly with hearing and with task dynamics. This chapter therefore serves as a bridge from visual perception alone toward richer models of multisensory interaction, where performance depends on how



(a) Perceiving visual-auditory targets at changing eccentricity



(b) Predicting visual-auditory reaction latency

Figure 6.1: *Predicting reaction latency under multisensory integration and changing eccentricity.* The top panel illustrates a virtual environment where two VR individuals perceive three targets: a biker ringing a bell (red, visual and audible), a pedestrian (blue, visual but silent), and a honking car behind the wall (green, audible but invisible). Their reaction time to each target differs based on their standing positions (different eccentricities) and sensory cues. The insets illustrate the first-person views assuming gazing at the center. The bottom panel shows our probabilistic model predicting the mean (marked with diamond dots) and the full reaction-time distribution for the two individuals. The reaction time for visual and auditory targets exhibits opposite trends as eccentricity changes.

sensory cues cooperate or compete under realistic time constraints.

This chapter proceeds in seven parts. We begin with a pilot study that quantifies reaction latency for visual-only, audio-only, and integrated stimuli under controlled variations in eccentricity, contrast, and volume. We then develop a multimodal perceptual model for reaction latency by adapting a drift-diffusion formulation to these experimental conditions. After establishing the model, we evaluate its accuracy and generalization on held-out data and compare its predictions with user behavior in more naturalistic immersive scenes. Finally, we demonstrate three application-facing analyses: assessing the perceptibility of visual-auditory content, deriving layout guidance for multisensory interfaces, and recommending video playback speed so that critical events remain perceivable.

Taken together, the studies in this chapter show that reaction latency can be measured, modeled, and used as a practical design signal for immersive graphics systems. Rather than treating perception as a passive afterthought, we study how the timing of multisensory evidence accumulation shapes what users notice and when they can act on it. That perspective opens a path toward interfaces and content pipelines that are not only visually compelling, but also tuned to the temporal limits of human multimodal perception.

6.1 Pilot Study: Measuring Visual-Auditory Reaction Latency

Setup and participants. As shown in Figure 6.2, the study was performed using a Varjo Aero VR headset (Table 6.1) with Sony XM5 noise-canceling headphones (Table 6.2). We recruited 6 participants (ages 23-29, 3 female) with normal or corrected-to-normal vision and hearing conditions. In this experiment, we included a smaller participant group for larger individual sample sizes to derive a probabilistic model in Section 6.2. For each participant, an eye-tracking



Figure 6.2: *Pilot study experimental setup.* The pilot study was conducted with a Varjo Aero headset, Sony XM5 noise-canceling headphones, and headset-provided eye tracking. Participants remained seated while perceiving visual and auditory targets in VR, allowing us to measure target-reaching latency through saccadic eye movements under controlled multisensory conditions.

calibration was applied before data collection. During the study, participants remained seated and perceived the visual and/or audio stimuli through the headset and headphones. Their eye movement data were recorded at 200 Hz with the gaze tracker provided by the headset. The study was approved by the Institutional Review Board (IRB).

Stimuli and conditions. As shown in Figure 6.3, the stimuli were a fixation-assisting indicator [142], a pair of visual targets, and/or a co-located spatial sound; all stimuli were synthesized in Unity. We characterize the target stimuli with three primary dimensions, eccentricity ($e \in \{5°, 10°, 15° 20°\}$), visual contrast ($c \in \{0, 0.1, 0.5, 4\}$), and auditory volume ($v \in$

Table 6.1: *Varjo Aero VR headset specifications.* We used the Varjo Aero VR headset for wide-field visual presentation and integrated eye tracking.

Display Resolution	2880 × 2720 pixels per eye
Display Refresh Rate	90Hz
Field Of View	Horizontal:115°, Diagonal: 134°
Eye Tracking Frequency	200Hz
Eye Tracking Accuracy	Sub-Degree

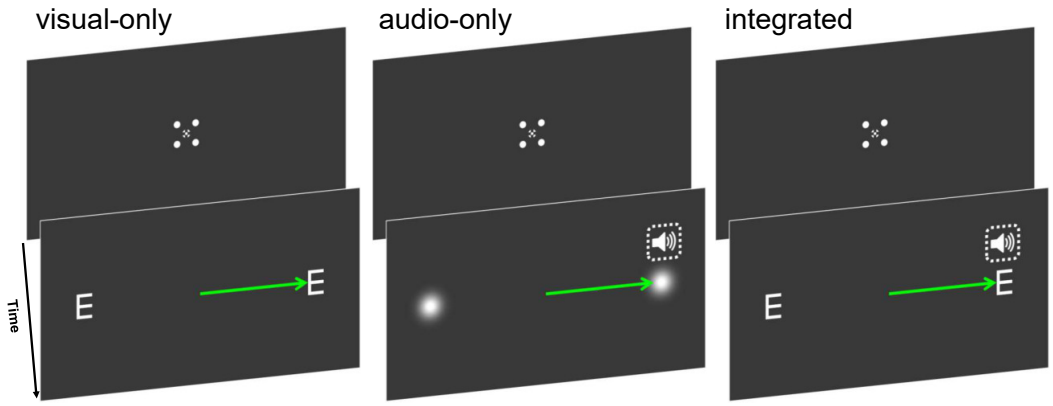


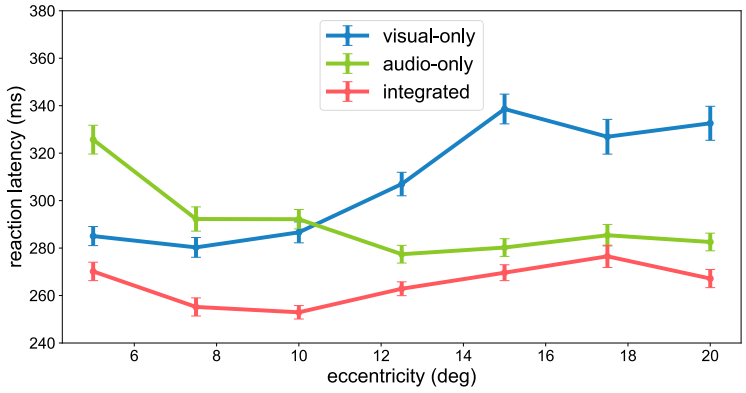
Figure 6.3: *Pilot study stimuli and task.* The task combined a fixation indicator, a pair of visual targets, and an optional co-located spatial sound. We studied visual-only (audio volume $v = 0$ dB), audio-only (visual contrast $c = 0$), and visual-auditory integrated conditions. Participants analyzed the presented cues and executed a saccade toward the perceived target side.

$\{0, 40$ dB, 52 dB, 66 dB$\}$) measured by a RISEPRO decibel meter. Note that $v = 0/c = 0$ indicates visual-only/audio-only stimuli. The sample points for each dimension were selected based on a preliminary study, visualized in Figure 6.4. The experiment was conducted with 12 visual-only, 12 audio-only, and 36 integrated conditions, for a total of 60 cases.

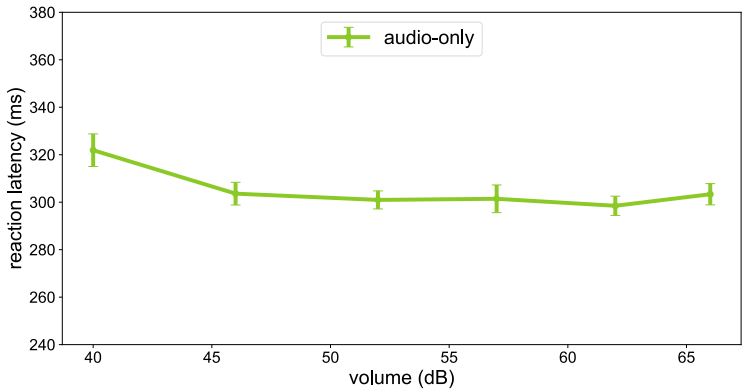
For $c \neq 0$ conditions, the visual targets were a pair of identical E letters, both toward the left or right. For the $c = 0$ audio-only condition, we substituted the "E" targets with a pair of identical Gaussian blobs. Participants had to rely only on their perception of the audio stimulus to perform the task without obtaining useful visual cues. To minimize any reaction-time influence from the visual placeholders, we intentionally designed the blobs to have high contrast ($c = 4$) across all trials. The audio stimulus was a 20 Hz-20K Hz white noise in all conditions.

Table 6.2: *Sony WH-1000XM5 headphone specifications.* We used the Sony WH-1000XM5 headphones for spatial audio presentation.

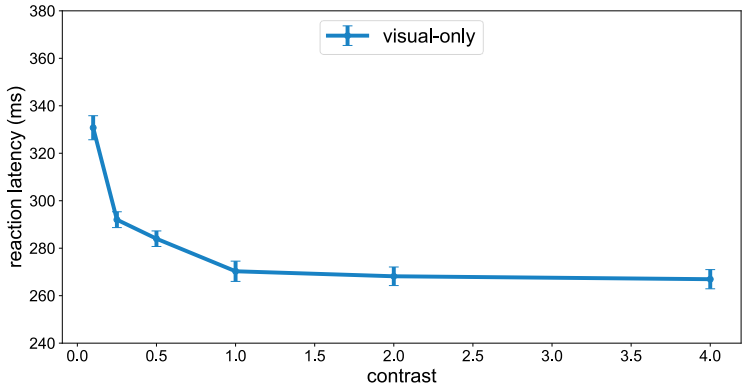
Frequency Response	20 Hz - 40000 Hz
Sensitivities	102 dB/mW (unit turned on)



(a) Effect of eccentricity



(b) Effect of volume



(c) Effect of contrast

Figure 6.4: *Preliminary study: visual-auditory reaction latency results.* We recruited four users (ages 23–29, one female) with normal or corrected-to-normal vision and hearing. The preliminary study used the same experimental setup and protocol as the pilot study in Section 6.1, but with denser sampling over eccentricity, volume, and contrast. This experiment helped us design the main pilot study by selecting the maximum, minimum, and neutrally challenging sample points for the stimulus characteristics while keeping the number of sample points balanced across the three dimensions to avoid excessive user fatigue.

Task and procedure. The task was target-reaching using saccadic eye movement. Specifically, the procedure was:

1. A fixation indicator appeared at the screen center.
2. After a successful fixation for 0.5 seconds, the indicator disappeared.
3. After a short delay randomly selected between 300ms and 500ms, the visual and/or audio stimuli appeared.
4. Participants then analyzed the presented cues and executed the corresponding saccade. In the visual-only and integrated conditions, they determined the orientation of the "E" targets and saccaded to the correct side based on that direction. In the audio-only condition, they relied on the perceived sound direction and saccaded to the Gaussian blob accompanied by the spatial audio cue.

For each participant, the experiment was performed with 6 repetitive sessions, each partitioned into 3 blocks (visual, audio, and integrated). The order of blocks in different sessions followed the balanced Latin square. Within each block, the trial conditions and the left/right placement of the targets were fully randomized to prevent performance bias or carry-over effects. The procedure took about 3 hours for each participant, including a short training session and breaks between sessions. With each condition repeated 36 times for each participant, we collected 12960 trials in total.

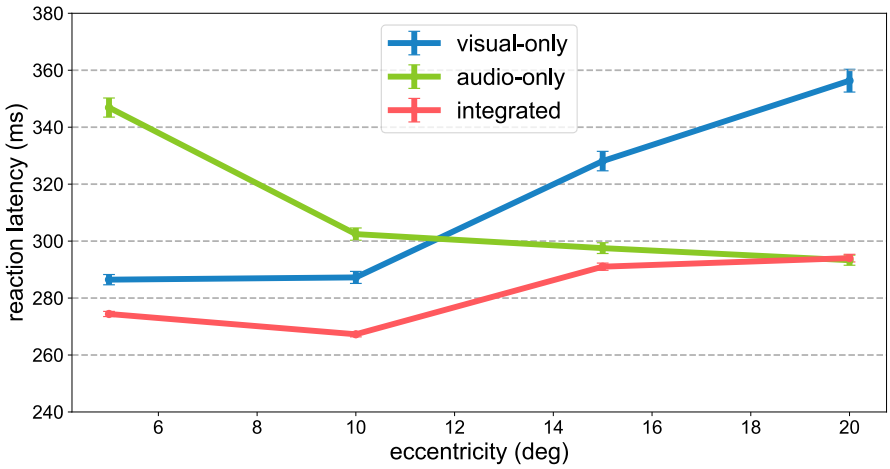
During the study, we considered saccades made in the wrong direction to indicate insufficient perception of the visual and/or auditory cues, so we rejected the corresponding trial and let users redo it. The redo was not executed immediately. Instead, the program randomly injected the rejected trial into the remaining unfinished trials so that users had no prior knowledge of the upcoming conditions.

Data processing. As in prior literature [143, 144], we measure users' reaction latency as the duration between target onset and the primary saccade. We use the statistical approach from Engbert and Mergenthaler [145] to detect the saccade onset. Studies have observed significant individual variances in reaction latency [146–148]. Therefore, similar to [141], a cross-subject calibration was performed using each participant's mean latency under a neutrally challenging condition ($c = 0.5, v = 52$ dB, $e = 10°$) to reduce inter-subject offsets before aggregation.

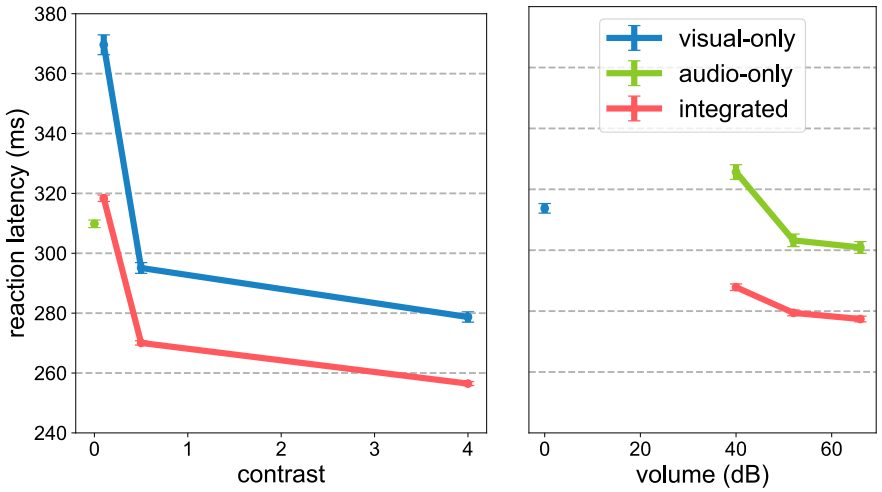
6.1.1 Results

The aggregated latency trends across eccentricity, contrast, and volume are summarized in Figure 6.5. As shown in Figure 6.5a, increasing eccentricity has opposite effects on the two unimodal cases: reaction latency increases in visual-only conditions (from 286.21 ± 1.78 ms at 5° to 355.76 ± 3.96 ms at 20°), but decreases in audio-only conditions (from 346.71 ± 3.35 ms at 5° to 293.01 ± 1.77 ms at 20°). We further compare our measured reaction time in visual-only conditions with prior work [141]. Our measured mean reaction time increases from 275.77 ms at $c = 0.5, e = 5°$ to 446.44 ms at $c = 0.1, e = 20°$. This is similar to the data measured under the same conditions in [141], which reported reaction times of 278.29 ms and 431.68 ms, respectively. We additionally use a statistical goodness-of-fit Kolmogorov–Smirnov (K-S) test [149] to compare the two distributions. The K-S test ($D = .10, p = 1.00$) failed to reject the null hypothesis that the datasets from our experiment and [141] are drawn from the same distribution, indicating their alignment.

A one-way repeated measures ANOVA shows a significant main effect of eccentricity on both the visual-only condition ($F_{3,2443} = 133.19, p < .0001$) and audio-only condition ($F_{3,2468} = 109.17, p < .0001$). Pearson coefficients validate the significant positive and negative correlations for the visual-only ($r(2445) = .36, p < .0001$) and auditory-only ($r(2470) = -.29, p < .0001$) conditions, respectively. We also observe their intersecting effect with eccentricity: visual-only



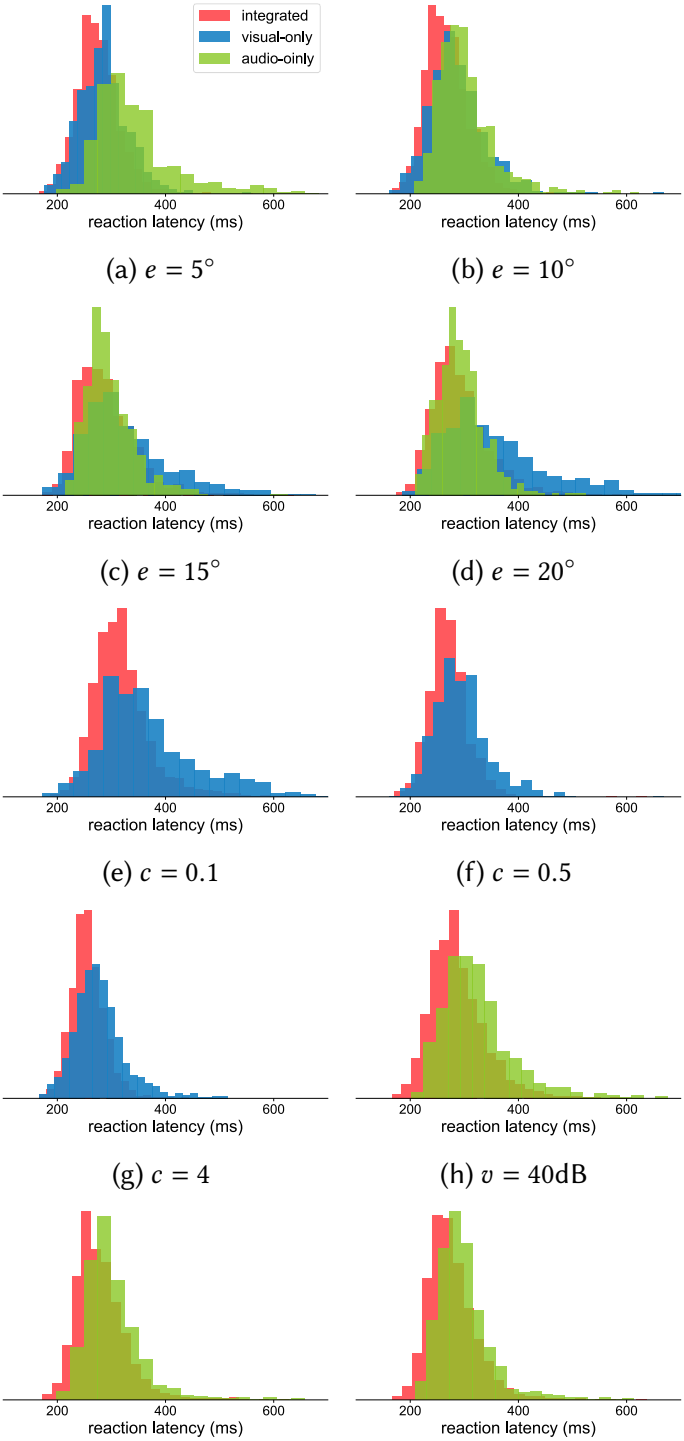
(a) Effect of eccentricity



(b) Effect of contrast

(c) Effect of volume

Figure 6.5: *Pilot study: visual-auditory reaction latency effects.* The plots summarize how reaction latency varies with eccentricity, visual contrast, and auditory volume across visual-only, audio-only, and integrated conditions. **Top:** eccentricity produces opposite trends for visual-only and audio-only targets, while integrated cues remain comparatively stable. **Bottom-left** and **bottom-right** show the latency distributions aggregated over contrast and volume, respectively. Here, $c = 0$ denotes audio-only conditions and $v = 0$ denotes visual-only conditions. Error bars represent standard error. Detailed histogram plots are provided in Figure 6.6.



(a) $e = 5°$ (b) $e = 10°$

(c) $e = 15°$ (d) $e = 20°$

(e) $c = 0.1$ (f) $c = 0.5$

(g) $c = 4$ (h) $v = 40$dB

(i) $v = 52$dB (j) $v = 66$dB

Figure 6.6: *Pilot study: visual-auditory reaction latency histograms.* The histograms show the detailed distribution of aggregated user behavior at representative eccentricity, contrast, and volume settings.

stimuli trigger faster reactions at low eccentricities ($\leq 10°$) than audio-only stimuli, but this trend is reversed as eccentricity further increases. In the integrated condition, the reaction latency is also influenced by eccentricity ($F_{3,7368} = 133.41, p < .0001$), but with less standard deviation (281.43 ± 11.1) compared to visual-only (314.20 ± 29.3) and audio-only (309.84 ± 21.5) conditions.

Figures 6.5b and 6.5c further show that higher contrast and volume generally lower reaction latency, although the volume effect is weaker than the contrast effect. Pearson coefficients indicate negative correlations between contrast and latency in visual-only ($r(2445) = -.36, p < .0001$) and integrated ($r(7370) = -.39, p < .0001$) conditions, and between volume and latency in auditory-only ($r(2470) = -.16, p < .0001$) and integrated ($r(7370) = -.08, p < .0001$) conditions. Figure 6.6 further visualizes the stochastic latency distributions for representative eccentricity, contrast, and volume settings, which motivates the probabilistic formulation introduced in the next section.

6.1.2 Discussion

The statistical analysis and visualization led us to several findings. First, we found that reactions are faster to audio-only cues at larger eccentricities, *contrary to* visual-only conditions where peripheral reactions are slower. Second, each sense is influenced by its specific attributes, such as visual contrast and auditory volume. Third, we observe that visual-only and auditory-only cues can approach the generally faster integrated cue at smaller and larger eccentricities, respectively, suggesting that audio-visual integration may be optimized for the more efficient and reliable sensory cue [12]. In an extreme case (e.g., $e = 20°$ and $c = 0.1$), reaction latency slightly increases in the integrated condition compared to auditory-only, indicating that a single sense can outperform the integrated condition when one sensory cue is prominent and the other is greatly degraded. Based on the comparison between our measured data and the existing visual-only reaction time model [141], our results also show statistical alignment with prior work. Those conclusions motivate us to develop a probabilistic computational model in Section 6.2.

6.2 A Computational Model for Visual-Auditory Reaction Latency

We develop a computational model using a widely applied computational framework, the drift-diffusion model (DDM, Section 6.2.1). In Section 6.2.2, we present a visual-auditory DDM-based formulation to depict the probability of users' reaction time given target contrast c (0 for audio-only), volume v (0 for visual-only), and eccentricity e. Finally, in Section 6.2.3, we fit the model with our data collection.

6.2.1 Drift-Diffusion Framework for Reaction Latency

Drift-diffusion model (DDM) [139, 150, 151] has been evidenced as an effective framework to model human decision-making over time in neuroscience, psychology, and economics [151–154], especially the accuracy and latency of human action performance [141, 155, 156]. DDM simulates a stochastic diffusion process to accumulate evidence with a non-zero drift while we examine the content. The cumulative level at time t is modeled as E_t whose trajectory is a Brownian motion with mean drift rate $\delta > 0$:

$$E(t;\delta) = \delta t + \varepsilon(t),\ t \geq 0, \tag{6.1}$$

where $\varepsilon(t)$ is a noise term depicting human behavioral uncertainty.

Once E reaches a threshold $\alpha > 0$, a decision or action is initiated with a reset. Therefore, the corresponding reaction latency $L(\alpha;\delta)$ can be formulated as:

$$L(\alpha,\delta) = \inf_{t>0} \{E\,(t;\delta) \geq \alpha\}\,. \tag{6.2}$$

Instancing $\varepsilon(t)$ as a Gaussian noise of standard deviation t derives:

$$L(\alpha, \delta) \sim \overline{\varepsilon}\left(\frac{\alpha}{\delta}, \alpha^2\right), \tag{6.3}$$

where $\overline{\varepsilon}$ is an inverse Gaussian distribution [157]. Next, we establish a DDM-based model that predicts users' reaction latency given the stimuli's visual-auditory characteristics.

6.2.2 Visual-Auditory Reaction Latency Model

We discuss and formulate how our investigated multisensory stimulus characteristics influence and determine these parameters in a DDM.

Threshold α. Intuitively, the evidence threshold α indicates how much evidence needs to be gathered, i.e., "how hard" a task is. In fact, prior literature has also suggested that it is primarily determined by the nature of a decision-making task [158, 159]. Interestingly, in a multisensory setting, the presence of each individual modality fundamentally changes the task nature. Therefore, we model the task based on which one(s) of the visual and auditory cues are present. That is

$$\alpha \coloneqq \alpha\left(\chi\left(c\right), \chi\left(v\right)\right), \tag{6.4}$$

where $\chi(x) \coloneqq \begin{cases} 0, x = 0 \\ 1, x > 0 \end{cases}$ is a Dirichlet function. Note that α is not correlated with the target eccentricity e, which does not alter the task nature. The sensory-dependent α plays a core role in integrating both uni- and multi-sensory tasks.

Drift rate δ. The difficulty of processing sensory inputs is usually modeled to determine the drift rate [141, 158]. Our pilot study analysis in Section 6.1.2 also shows the significance of the stimulus characteristics, including visual contrast c, audio volume v, and eccentricity e. Motivated by

these observations, we model visual-only, auditory-only, and visual-auditory-integrated evidence accumulation rates as $\delta \coloneqq \delta(c, v, e)$.

Integration. The computational model shall obtain the explicit representation of $\alpha\,(c, v)$ and $\delta(c, v, e)$ as presented above. There has been extensive neurological literature and hypothesis on the process of multimodal integration, such as time-windowing [160], or averaging [161]. While weighing individual modalities, performance optimality (minimal variances) have been considered [12, 162], while in visual-audio tasks, both optimal and sub-optimal patterns might be observed [161, 163, 164]. However, instead of seeking a neurological explanation, we aim to develop a characterized and executable algorithm to assist downstream graphics applications. Therefore, we simplify the integrated model as a pure computational regression from our acquired data, such that both $\alpha\,(\cdot)$ and $\delta\,(\cdot)$ are formulated through learned functions with the fitting process detailed below.

6.2.3 Fitting Model Parameters: δ and α

Using the psychophysical data collected in Section 6.1, we compute and fit the DDM parameters $\{\alpha, \delta\}$ by noting their relationships with the mean and variance of the latency distributions L, which are modeled as inverse Gaussians:

$$\mathbb{E}[L] = \frac{\alpha}{\delta},\ \ \mathbb{V}[L] = \frac{\alpha}{\delta^3}. \tag{6.5}$$

Therefore, for each presented sensory identified by $\{\hat{\chi}_c, \hat{\chi}_v\}$ and the corresponding reaction latency data $\{\hat{L}\}$, we can approximate the evidence threshold α as:

$$\alpha\,(\hat{\chi}_c, \hat{\chi}_v) = \sqrt{\mathbb{E}[\hat{L}]^3 / \mathbb{V}[\hat{L}]}. \tag{6.6}$$

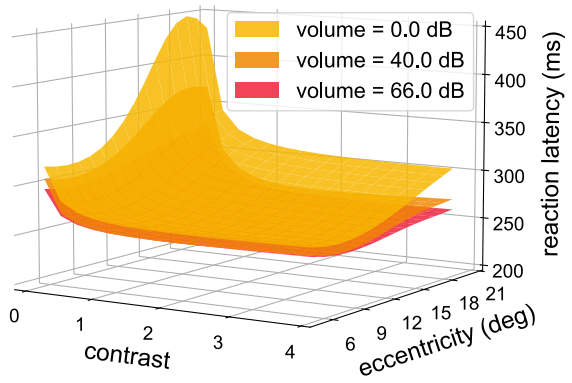
Similarly, we can approximate the drift rates by sampling the subset $\{\hat{L}\}$ from our entire dataset as only those trials from condition $\hat{c}, \hat{v}, \hat{e}$:

$$\delta(\hat{c}, \hat{v}, \hat{e}) = \sqrt{\mathbb{E}[\hat{L}]/\mathbb{V}[\hat{L}]}. \tag{6.7}$$

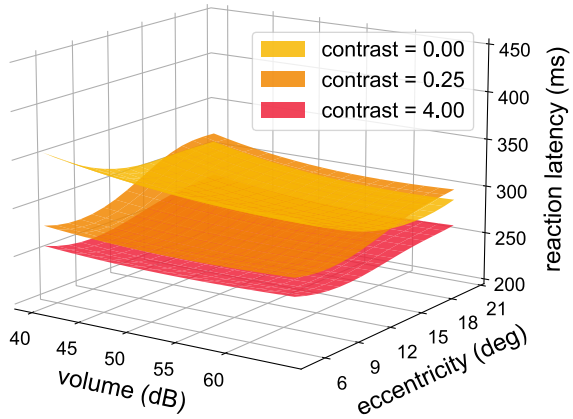
However, there is a remarkable distinction in learning α and δ. All α can be directly approximated from our data given which sensory inputs are presented. However, δ analytically depends on the exact values of each sensory cue. To ensure local smoothness, we formulate the underlying $\delta(c, v, e)$ as a Radial Basis Function Neural Network (RBFNN) and optimize the parameters therein using sample-computed target values. The final model is visualized in Figure 6.7. The drift rate component $\delta(\cdot, \cdot, \cdot)$ of our visual-auditory-integrated DDM, which is a function of contrast c, volume v, and eccentricity e, was modeled using a Radial Basis Function Neural Network. Specifically, we modeled our function as:

$$\delta(c, v, e) = \sum_{i=1}^{N} \lambda_i \rho \left(\left\| \begin{bmatrix} \begin{cases} 0 & \text{if } c = 0 \\ \log_{10}(c) & \text{if } c \neq 0 \end{cases} \\ v \\ e \end{bmatrix} - b_i \right\|, \sigma_i \right), \tag{6.8}$$

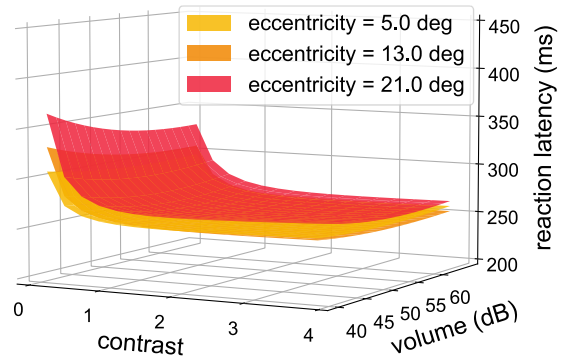
where ρ is the Gaussian basis function and b_i/σ_i is the radial basis center/Gaussian deviation for the ith neuron. Since we observed a fast drop-down of reaction latency in response to the contrast change from 0.1 to 0.5, we applied a log transform to the contrast value to stabilize the nonlinear relationship. We chose $N = 25$, an L_2 loss, and the Adam optimizer to train the RBFNN for 5k epochs. The learning rate started at .01, and a 10× learning-rate decay was performed at epoch 2.5k. Since our dataset of $\{(c, v, e), \hat{\delta}(c, v, e)\}$ pairs is fairly small, we performed full-batch training instead of mini-batch training, i.e. the whole dataset was used for each parameter update.



(a) Contrast-Eccentricity



(b) Volume-Eccentricity



(c) Contrast-Volume

Figure 6.7: *3D visualization of the visual-auditory integrated DDM.* Each subfigure visualizes a 2D slice of the mean value derived from the 3D latency distribution $L \coloneqq L\left(\alpha\left(c,v\right),\delta\left(c,v,e\right)\right)$ over the variables (c,v,e).

For visual-only and auditory-only conditions, we set $v = 0$ or $c = 0$ as the input to our model. The training was conducted using a single Nvidia RTX 4090 GPU and took less than 1 minute to complete.

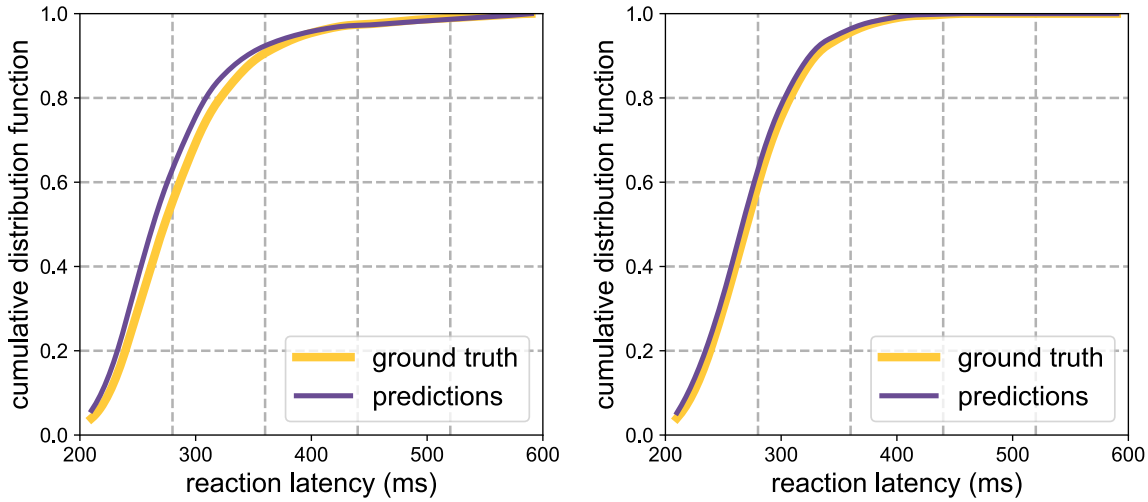
6.3 Model Accuracy and Generalization

Dataset. We split the reaction time data from Section 6.1 into non-overlapping training-testing partitions for cyclic cross-validation. We first randomly sample 1/6 of the entire dataset to compose the evaluation set used to measure the model's prediction accuracy, and repeat this procedure six times ($\mathcal{R}_{1,2,...,6}$). We also split the data by participants ($\mathcal{P}_{1,2,...,6}$) to evaluate the model's generalizability to new users. Specifically, each of the 6 testing datasets contains all participant trials, with the remaining reserved to re-train our model.

Results and discussion. When comparing between a testing dataset and model predictions, we adopt two metrics, the K-S test and Wasserstein distance (WD) [165]. WD provides a continuous-domain metric on the distinction between the two samples. Figure 6.8 visualizes the cumulative distribution function (CDF) of measured data and model predictions for both random ($\mathcal{R}$) and

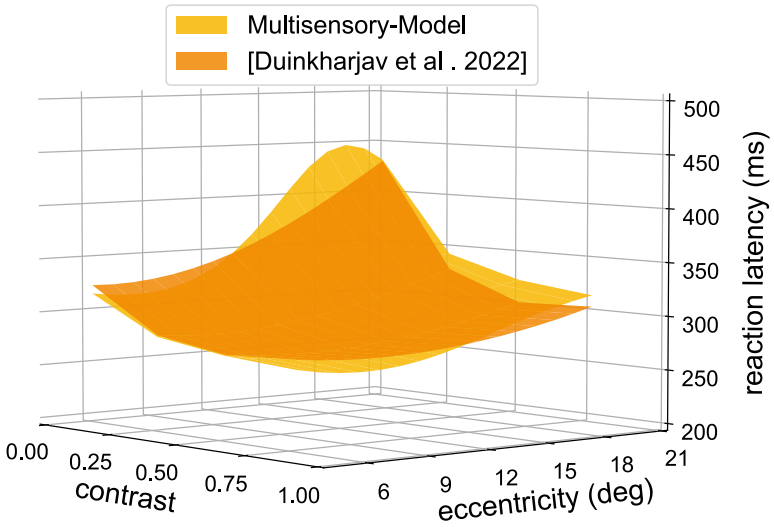
Table 6.3: *Generalization to unseen users.* D and p denote the distance metric and its corresponding p-value for the K-S test, respectively. The null hypothesis cannot be rejected in the K-S test due to $p \gg .05$. With the mean reaction latency across all pilot study trials being 293ms, a Wasserstein distance(WD) of 8.88 indicates a 3.03% mean deviation.

Statistics \ Test Set	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$
K-S test D	.06	.10	.10	.13	.10	.06
K-S test p	1.00	.99	.99	.95	.99	1.00
Wasserstein Dist	4.14	6.46	10.75	13.26	20.08	4.40
Statistics \ Test Set	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$	$\mathcal{R}_6$
K-S test D	.06	.06	.10	.10	.06	.10
K-S test p	1.00	1.00	.99	.99	1.00	.99
Wasserstein Dist	9.75	6.09	8.37	8.21	6.58	8.52



(a) Results for a random partition

(b) Results for a random subject



(c) Model comparison with prior work

Figure 6.8: *Cross-validation results and comparison with prior work.* The top-left panel shows the results for one randomly sampled held-out partition, $\mathcal{R}_1$. The top-right panel shows the testing results on $\mathcal{P}_1$, with the model trained on the other five participants. The bottom panel compares the visual-only predictions of our model with those from prior work [141].

participant ($\mathcal{P}$)-based partitions. Table 6.3 shows all numerical analysis results.

In summary, all K-S tests on the 12 partitions show $p \geq .95$, failing to reject the null hypothesis that the model prediction and the test data are drawn from the same distribution. The WD for random/participant-based partitions is $7.92 \pm 1.23/9.85 \pm 5.63$. Given the mean latency of 293 ms from the entire dataset, the model prediction may be approximated as exhibiting $< 2.70\%/3.36\%$ deviation. These results validate the distributional alignment between test data and our model's predictions. Specifically, the analysis with random partitions confirms our model's prediction accuracy. Moreover, the analysis with participant-based partitions demonstrates the model's generalizability to new users in practical applications.

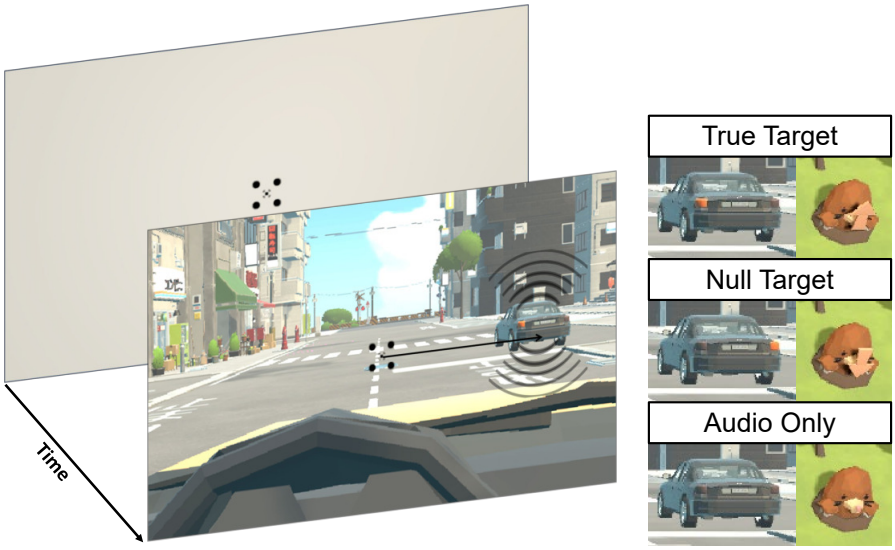
Comparison with prior work. As in Figure 6.8c, we further compare our model with prior open-source literature from Duinkharjav et al. [141]. We test with a median $f = 1.5$ spatial frequency (of their range $\{.5, 1, 2, 4\}$) as the frequency input for their model, while maintaining consistent contrast and eccentricity to evaluate and compare the reaction-time predictions of both models. Since the previous model only predicts the visual-only condition, we set $v = 0$ for our model. Our model aligns with [141] across various contrasts and eccentricities ($D = .10, p = 1.00$) from a K-S test.

6.4 Visual-Auditory Reaction Latency in Immersive Environments

From the pilot study (Section 6.1.2) and our model (Section 6.2), we observed how the visual-audio interplay significantly influences reaction time. This study evaluates the model's applicability to varied target-identification tasks in realistic and representative scenarios: driving and gaming.

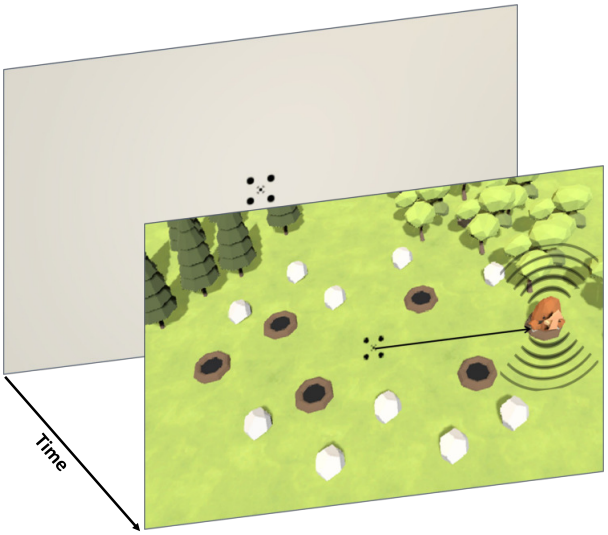
Participants and tasks. Twelve adults (ages 21-46, 3 females) with normal or corrected-to-normal vision and hearing participated in the study. The appearance orders of the three conditions and the two scenes were randomized and counter-balanced across participants. Rather than examining a pair of identical targets in Section 6.1, we aim to validate the model with a different go/no-go target-reaching task [166, 167]. At the beginning of each trial, participants were shown a grey background with a fixation indicator at the center of the display. After a successful fixation, the scene with the task object appeared (a car or a mole). Then, after a randomized .75-1 second delay, an additional task-relevant stimulus (turning light/arrow with or without a spatial sound) appeared on the object to indicate true or null targets (see the scenes-and-setup paragraph below for details). Participants were directed to make a saccade towards the true target or to maintain the gaze fixed if perceiving a null target. Similar to other go/no-go paradigms, we included null trials to ensure that participants were perceptually analyzing the targets instead of blindly responding to any changes.

Scenes and setup. As summarized in Figure 6.9, and detailed in Figures 6.9a to 6.9c, we designed two virtual environments to simulate realistic multimodal scenarios. In the *driving scene*, the true target was the car with an overtaking turn signal (left/right-side car turning right/left light on) and/or with a spatially co-located car horn sound (60 dB); the null target was the car with the opposite turn signal and/or a spatially misaligned sound. In the *whack-a-mole gaming scene*, the true target was the mole with an up arrow and/or a spatially co-located rustled sound (55 dB); the null target was the mole with a down arrow and/or a spatially misaligned rustled sound. Visual contrasts were approximated as the averaged Weber contrast from a three-layer Laplacian pyramid. Auditory volume was physically measured with the same device as Section 6.1. The hardware remained the same as Section 6.1.



(a) Simulated driving

(b) Targets



(c) Whack-a-mole gaming

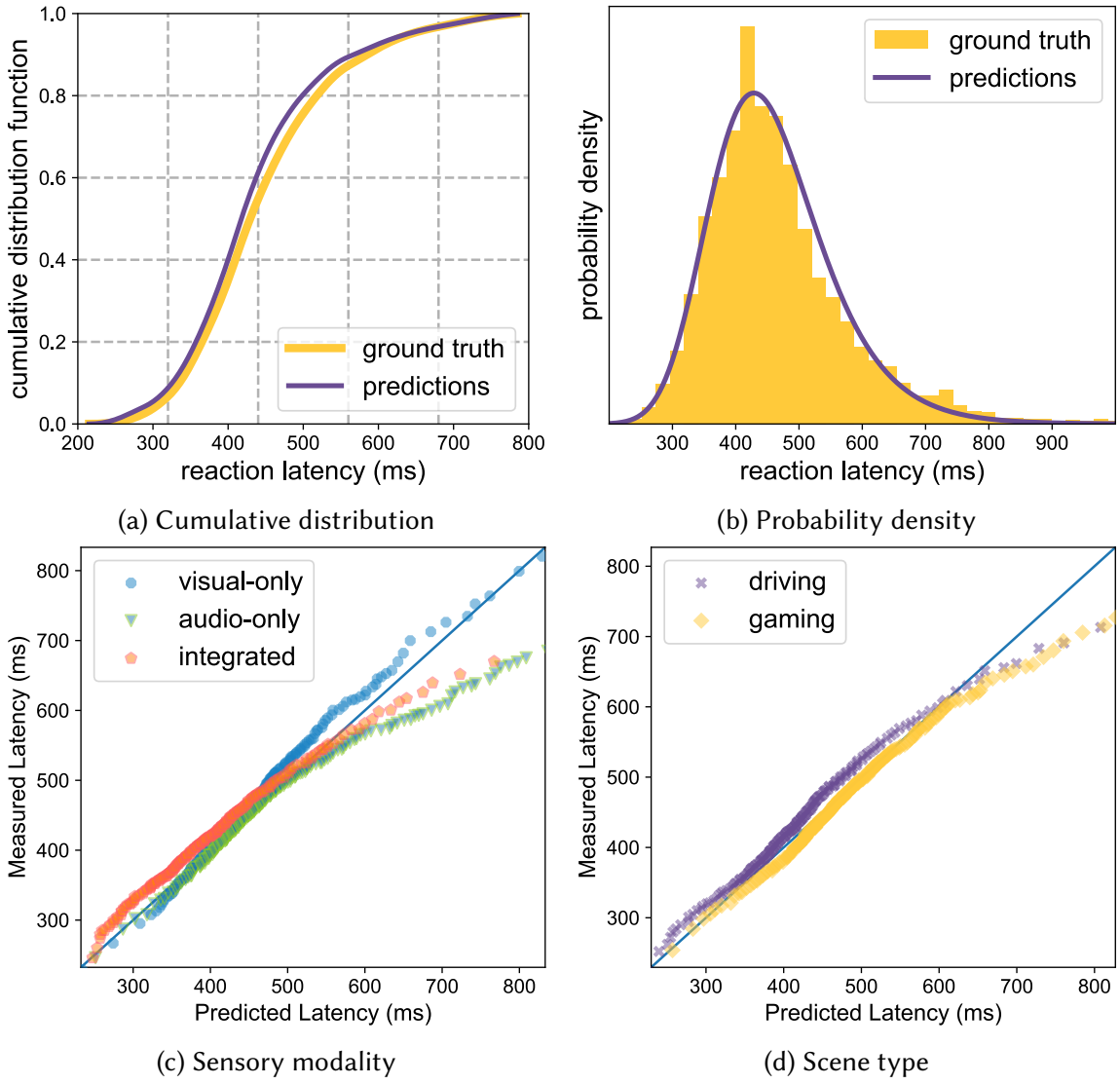
Figure 6.9: *Validation study scenes and target cues.* Our validation study uses two representative multimodal scenarios: simulated driving and whack-a-mole gaming. The targets panel summarizes the true-target, null-target, and audio-only cue configurations used across both scenes, enabling evaluation of the reaction-latency model in more naturalistic go/no-go target-identification tasks.

Conditions. For each scene, we also experimented with three sensory modalities (visual-only, audio-only, and integrated). In the audio-only condition, all targets were designed as visually unidentifiable shown in Figure 6.9b. For example, the mole didn't hold any arrow in its hand, and the car didn't turn on its turn signal. Therefore, the true target could only be identified by whether the auditory direction is on the same side as its visual appearance. Aiming to evaluate our model's generalizability in unseen scenarios, we intentionally selected different eccentricity levels from the pilot study (Section 6.1). Within each condition, we experimented with three horizontal eccentricities not presented in the pilot study (7°, 14° and 21°). We also introduced vertical eccentricities from 1° to 4° to enable validation of the model's generalization capability. Each participant performed 16 repeats × 3 eccentricities × 2 scenes × 3 sensory modalities, resulting in 288 × 12 participants = 3456 trials. They also completed 3456 × 50% = 1728 null trials. Overall, we collected 3456 + 1728 = 5184 trials in total.

Results and discussion. The overall validation outcome is summarized in Figure 6.10. In comparison with our model prediction and evaluation data, Figures 6.10a and 6.10b visualize the aggregated cumulative distribution and probability density of the measured data and the model predictions. In addition, Figures 6.10c and 6.10d present quantile-quantile (Q-Q) plots across different sensory modalities and scene types; samples lying close to the diagonal indicate strong distributional agreement between the model and the measured data. We also statistically validate the model with user data across modalities and scenes using goodness-of-fit (K-S) and

Table 6.4: *Generalization to natural scenes.* This table reports K-S test statistics and Wasserstein distances across sensory modalities and scene types.

Statistics \ Test Set	All	Visual	Auditory	Integrated	Driving	Gaming
K-S D	.06	.10	.10	.10	.13	.06
K-S p	1.00	1.00	1.00	1.00	.96	1.00
WD	21.41	15.94	41.86	23.69	22.85	15.01
Relative WD	4.56%	3.34%	8.41%	5.48%	5.05%	3.07%



(a) Cumulative distribution

(b) Probability density

(c) Sensory modality

(d) Scene type

Figure 6.10: *Validation study results vs model predictions.* These plots compare measured target-reaching latencies from the validation study against the model predictions. The top row shows the aggregated cumulative distribution and probability density, while the bottom row shows quantile-quantile comparisons after grouping the data by sensory modality and by scene type. Samples lying close to the diagonal indicate strong agreement between the model and the measured data.

distribution-distance metrics (WD), as detailed in Table 6.4. For the entire dataset, the K-S test failed to reject the null hypothesis, with $D = .06$, $p = 1.00$, and WD = 21.41. This analysis demonstrates our model's applicability in naturalistic scenes, maintaining consistent predictions with variations in scene types or novel target characteristics.

6.5 Assessment of Visual-Auditory Perceptibility

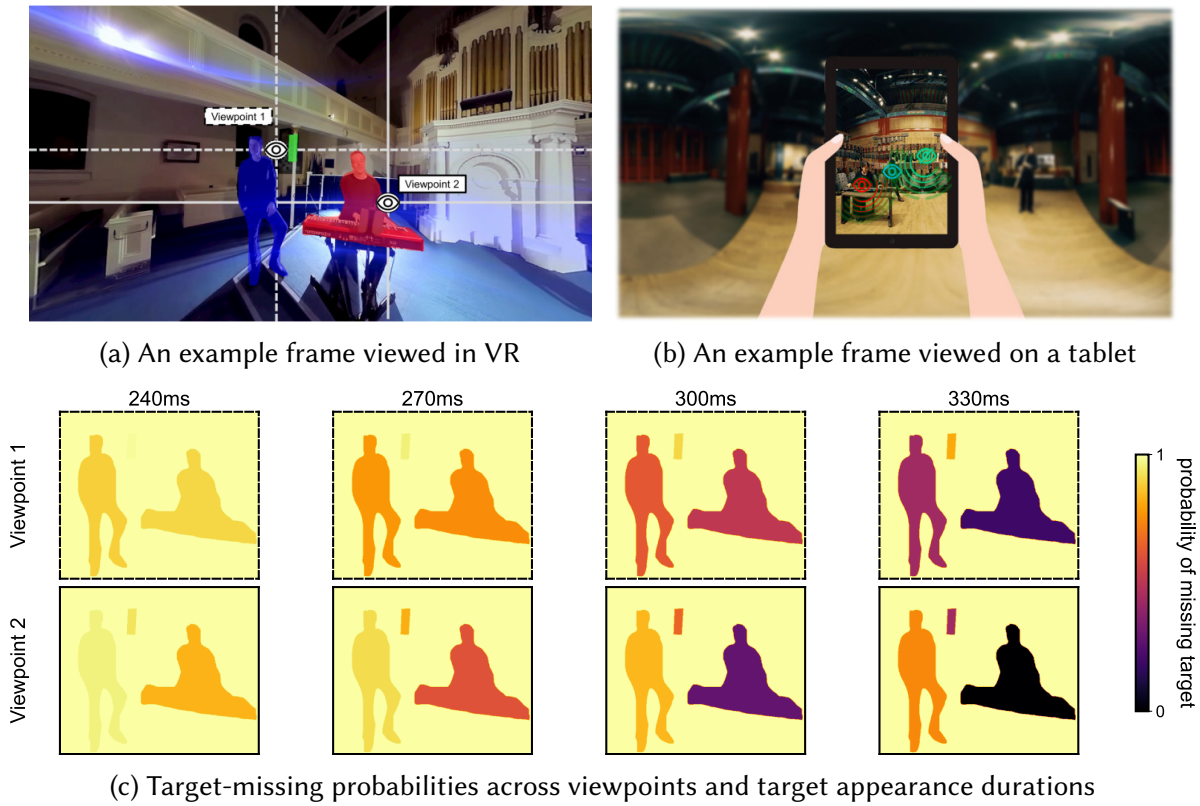
In scenarios like VR games (e.g., Beat Saber), immersive simulators (e.g., VR driving simulation), or content creation (e.g., commercial video), observers may overlook a quickly appearing target, such as a gaming block, a virtual pedestrian, or a promotional product. Our model can approximate the likelihood of users reacting to visual and/or auditory events in time, thereby assessing the perceptibility of multisensory content. The assessment can offer guidance for the design of VR interactive applications and video creation.

Perceptibility approximation. Our model predicts the probabilistic distribution of users' reaction latency for each target. We then approximate the likelihood of missing a target as the accumulated probability that the reaction time exceeds the target appearance duration. That is, for each target defined with $\{e, c, v\}$ and appearance duration d, the probability of missing a target is:

$$\begin{aligned} P_{\text{miss}}(e,c,v,d) &\coloneqq P\left(L\left(\alpha(c,v), \delta(e,c,v)\right) > d\right) \\ &= 1 - \Phi\left(\sqrt{\frac{\alpha^2}{d}}\left(\frac{\delta d}{\alpha} - 1\right)\right) - e^{2\alpha\delta}\Phi\left(-\sqrt{\frac{\alpha^2}{d}}\left(\frac{\delta d}{\alpha} + 1\right)\right), \end{aligned} \tag{6.9}$$

where Φ is the cumulative distribution function of a Gaussian.

Dataset. Here, we demonstrate the proof of concept with a public video dataset for virtual reality applications. We experiment with a visual-audio omnidirectional video dataset from [168]. It contains 17 omnidirectional footage clips (2K-4K resolution @ 24-60 FPS) with first-order B-format



(a) An example frame viewed in VR

(b) An example frame viewed on a tablet

(c) Target-missing probabilities across viewpoints and target appearance durations

Figure 6.11: *Assessing target-missing probability during video playback.* The top-left panel shows an example frame viewed in VR. The blue, green, and red masks indicate visual-only, audio-only, and visual-auditory integrated targets, respectively. The top-right panel shows an example panoramic video frame viewed on a tablet with a 40° effective FoV. The blue, green, and red icons indicate a visual-only character, an audio-only organ, and a visual-auditory integrated piano player. The bottom panel visualizes the predicted target-missing probabilities for the VR example frame (Figure 6.11a) across different viewpoints and target appearance durations.

ambisonic surround sound @ 4, 000 Hz.

Experiment and results. We rendered the omnidirectional videos in VR ($120° \times 90°$). Figure 6.11 summarizes the perceptibility-analysis pipeline and representative outputs. We extracted the targets via instance segmentation with MMDetection [169] and estimated the audio-source directions from the ambisonics (see Figure 6.11a). We then visualized the probability of missing targets under different viewpoints and target appearance times, as shown in Figure 6.11c. The visualization indicates that across visual-only, audio-only, and integrated conditions, both viewpoint and target

appearance time (240 ms to 330 ms) substantially influence target-miss likelihood, with respective shifts of 17.3%/21.0%/8.3% and 42.9%/44.8%/62.0%.

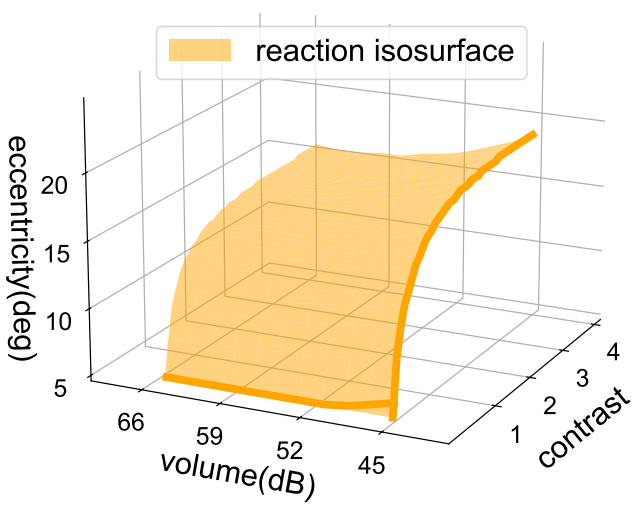
6.6 Layout Design for Visual-Auditory Content

XR interfaces can be designed with not only visual but also auditory cues [170, 171]. So far, we have little quantitative guidance on how such multisensory cues should be spatially designed so that users react faster to the individual elements.

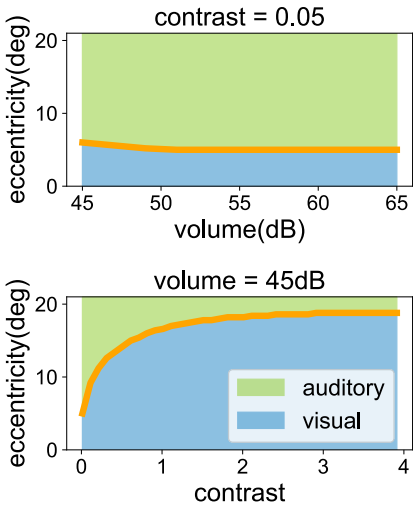
The overall design analysis is summarized in Figure 6.12. As shown in Figure 6.12a, we leverage our model to suggest whether an audio cue or a visual cue, if not both, is more beneficial for triggering faster user reactions at a given eccentricity. Specifically, the model compares the mean reaction time under visual-only and audio-only cues for a given contrast-volume pair and determines which sensory modality is predicted to be faster across eccentricity ranges. The resulting decision boundary is visualized as an isosurface in Figure 6.12b. For eccentricities above the surface, audio cues are predicted to trigger faster responses; below the surface, visual cues are predicted to be faster. Two representative 2D slices from this 3D isosurface are shown in Figure 6.12c. The audio-only cue triggers faster reactions over a larger area when $c = 0.05$, but over a smaller area when $v = 45$dB. These slices make the design implication explicit: the preferred modality depends jointly on eccentricity and cue quality. A low-contrast visual cue benefits more from auditory support in the periphery, whereas a sufficiently visible target can remain visually dominant over a larger central region. This kind of model-derived guidance can inform where multimodal cues should be placed in an AR interface before the interface is deployed to users.



(a) Example visual-auditory content layout in AR



(b) Reaction latency isosurface



(c) 2D slices from the isosurface

Figure 6.12: *Visual-auditory AR content layout design.* The top panel shows an example AR content layout in which designers may choose to place visual and auditory cues at different eccentricities. Our model provides quantitative guidance for placing those cues so as to trigger faster user reactions. The bottom-left panel shows the model-derived isosurface on which visual-only and audio-only cues are predicted to yield identical mean reaction latency. The X-, Y-, and Z-axes represent audio volume, visual contrast, and eccentricity, respectively. Visual-only stimuli are predicted to trigger faster/slower reactions than audio-only stimuli below/above the surface. The bottom-right panel visualizes two representative 2D slices from the isosurface. The blue and green regions indicate where visual or auditory cues are expected to trigger faster reactions. We selected a visual-driven slice (v = 45dB) and an audio-driven slice (c = 0.05).

6.7 Video Playback Speed Recommendations

For faster content consumption, today's video streaming platforms commonly support accelerating playback speed (e.g., 0.25× to 2× on YouTube). However, we still lack a comprehensive guideline that captures the experiential consequences of these adjustments. For instance, if a video is accelerated, is there a risk of crucial events being missed before users can react to them? In fact, studies have identified drawbacks of accelerated video play on learning effectiveness [172]. Here, by leveraging the perceptibility assessment from Section 6.5, we can predict users' reaction time to a certain target during video watching. Higher playback speed reduces the target appearance time and makes it more likely to be overlooked. Therefore, our model can suggest a maximal playback speed so that the video is not over-accelerated and observers do not miss crucial events.

Dataset preprocessing. We reused the dataset in Section 6.5, standardized the footage, and annotated the directions of visual and auditory sources. Since the intended targets may be arbitrary depending on individuals, we simplify by identifying the visual, auditory, and integrated targets as described below. For consistency in both spatial and temporal resolutions across all frames, we normalized all footage by downsampling to 320×240 @ 10 Hz.

- Identifying visual targets: without loss of generality, we assume the potential visual targets to be human characters. Specifically, in each plane frame, we detect the two characters with the highest confidence scores with MMDetection [169]. The visual eccentricities are directly obtained from the detection. Their contrasts were approximated similarly to Section 6.4.

- Identifying auditory targets: the original ambisonic audio in the dataset is sampled at 48,000 Hz, where each audio sample encodes a primary sound source [173]. To robustly determine the direction of the audio targets, we uniformly downsampled the audio to 150 samples per video frame and estimated the direction by averaging their horizontal eccentricities [174]. We approximate the audio source's volume by mapping the ambisonic W-channel to the

scale of 44 dB-63 dB.

- Identifying visual-auditory integrated targets: we define a pair of visual and auditory targets as integrated if their spherical angular distance is less than 12°; otherwise, they are treated as visual-only or audio-only.

- Approximating target appearance duration: regardless of the video content, eye rotations change our retinal image and thus target visual eccentricities. Therefore, we approximate the duration of target appearance (defined by unchaged {e, c, v }) as an averaged fixation duration of approximately 400 ms [175]. While performing content-aware studies with eye-tracked datasets such as [9] is an interesting future avenue, it falls beyond our main focus in this research, as discussed in Section 6.8. Nonetheless, our model and the analysis framework are still applicable to target-adaptive durations.

Experiment and results. To simulate viewing with an iPad Pro tablet (22 cm×28 cm) in portrait mode (3:4 aspect ratio), we uniformly re-framed 9 plane videos ($40° \times 53°$) along the equator (0° latitude) of each panoramic video (as illustrated in Figure 6.11b) and obtained 1648/931/389 frames with visual-only/audio-only/integrated targets. For fair cross-comparisons, we randomly sampled 350 frames for each sensory condition. As a proof-of-concept analysis, we approximate the user's gaze at a fixed display position before each target-reaching event. Therefore, we assume each user's current eye gaze is at the display center with a 30 cm viewing distance. Section 6.7 shows the overall statistics of the dataset. The predicted target-missing probabilities (P_{miss}) across the dataset are visualized in Section 6.7. Without loss of generality, we analyze with 40% chance of missing as an "acceptable" threshold, i.e., $P_{\text{miss}} \leq 0.4$. As in Section 6.7, we observe that 1.35 ± .06×/1.28 ± .04×/1.46 ± .04× to be the maximal playback speed for visual-only/audio-only/integrated targets.

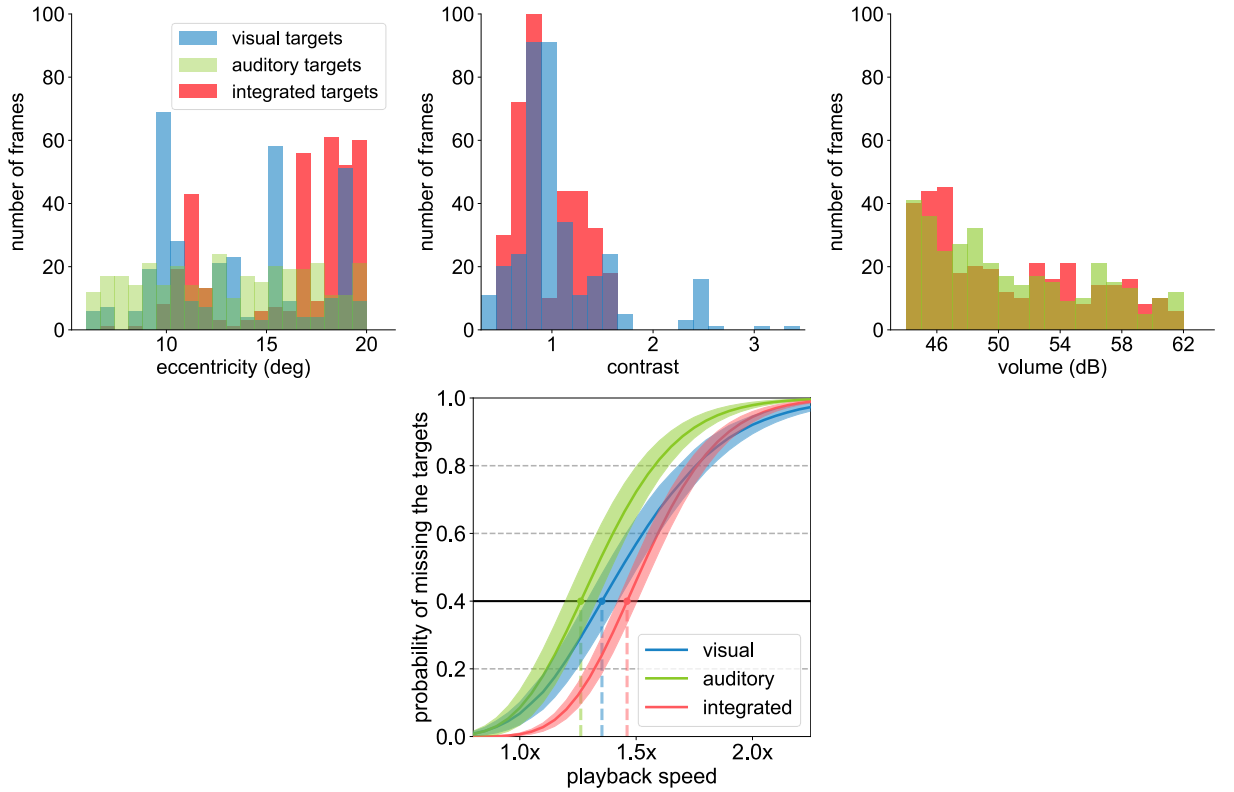


Figure 6.13: *Video playback speed recommendations.* The top panel summarizes the content characteristics of our collected video dataset. The bottom panel shows the mean and standard deviation of the predicted target-missing probabilities as playback speed varies, assuming a viewing distance of 30cm.

6.8 Discussion

This chapter establishes visual-auditory reaction latency as a practical human-centered signal for immersive graphics systems. Through controlled VR psychophysics, we measured how target eccentricity, visual contrast, and auditory volume jointly influence the time required to identify a target and initiate an eye movement. Those measurements then supported a probabilistic model that generalizes to novel conditions and produces application-relevant quantities such as target-miss probability and modality-dependent reaction advantage. Beyond fitting the experimental data, our model proved useful in naturalistic validation tasks and in three downstream analyses related to perceptibility assessment, interface layout, and video playback speed.

Several limitations remain. First, our model is intentionally built around a restricted set of

stimulus factors: eccentricity, contrast, and volume. Those variables capture major aspects of visual-auditory target detectability, but they do not exhaust the perceptual factors that matter in practice. Object motion, target size, texture, temporal modulation, auditory frequency content, and higher-level scene context can all affect response speed [176, 177]. Extending the model to account for these factors would improve realism, but it would also substantially increase the dimensionality of the psychophysical sampling problem.

Second, reaction latency in this work is measured using saccade onset. This choice is well motivated for gaze-mediated target acquisition and is consistent with prior visual reaction-time modeling, but it is not the only behavior of interest in immersive systems. Many tasks ultimately depend on hand, head, or whole-body responses rather than eye movements alone [143, 178]. Future work could combine gaze with controller motion, EMG, or hand tracking to study how multisensory evidence propagates from attention shifts to overt action.

Third, the experiments and model assume a relatively clean target-identification setting in which the relevant target is known and distractors are treated as background. Real scenes are often more cluttered. Multiple targets may compete for attention, auditory sources may overlap, and semantic saliency may reshape how evidence is accumulated. Integrating reaction-latency modeling with saliency prediction or scene-level attention models would help close the gap between controlled psychophysics and naturalistic content analysis [9, 179].

Finally, the application analyses in this chapter should be understood as model-driven guidance rather than complete end-to-end user studies for each downstream use case. The content-perceptibility and playback-speed examples illustrate how the model can provide quantitative support for design decisions, but they rely on simplifying assumptions about current gaze position, target annotation, and event duration. A natural next step is to embed the model into closed-loop systems that personalize content using eye tracking or predictive attention estimates in real-time.

Even with these limitations, this chapter demonstrates that multisensory reaction latency can be operationalized in a way that is both scientifically grounded and useful for graphics design. We

can measure how fast users respond to visual-auditory events, predict how that response changes across conditions, and use those predictions to reason about whether immersive content is likely to be noticed in time. In the context of this dissertation, that result broadens the role of perception from rendering quality and visual attention to the timing of multimodal interaction itself.

Chapter 7

Force-Aware Interfaces for XR Interaction

Immersive systems have become increasingly effective at presenting convincing visual and auditory content, yet physical interaction in those systems is still often coarse. Users can see and hear rich virtual events, but the force they exert with their own hands is frequently reduced to controller triggers, collision-based heuristics, or indirect kinematic proxies. That mismatch matters most in tasks that depend on dexterity. Pressing, pinching, tapping, and manipulating objects are not simply positional gestures; they are forceful actions whose meaning depends on how the hand engages with the environment.

This chapter studies how finger-level force can be sensed and used as a direct interaction modality for immersive computing. The key observation is biological: many of the muscles that control finger force are located in the forearm, and their activations can be recorded non-invasively with surface electromyography (sEMG). If those signals can be decoded robustly and in real-time, immersive systems gain access to a richer description of user intent without requiring bulky gloves or dedicated on-hand force sensors. In that sense, the chapter extends the dissertation's human-centered perspective from visual and auditory perception to active motor behavior and haptic intention.

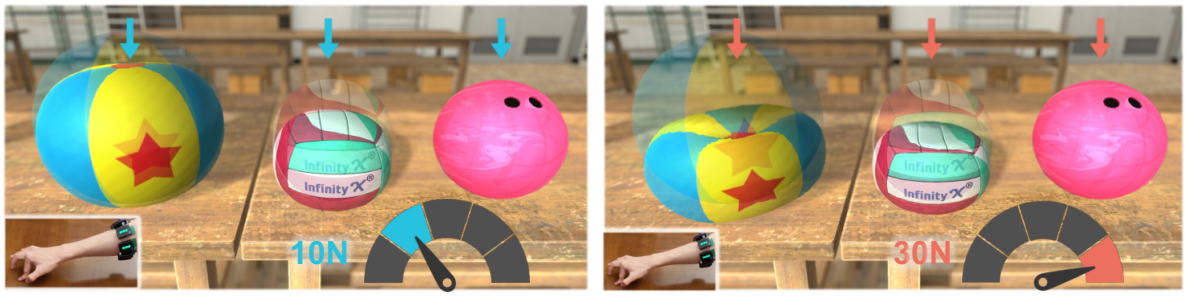
Force-aware interaction has long been recognized as essential for realistic virtual experience.

Haptic rendering, tactile displays, and force-sensing wearables have all pushed the field toward richer contact sensations [45, 46, 50, 180]. At the same time, hand tracking and controller-based interfaces have improved the spatial side of manipulation. What remains difficult is closing the full loop between what users physically do and how virtual objects respond. Existing force-sensing interfaces can be accurate, but they often constrain dexterity precisely because they must be attached to the hand [49, 181]. Camera-based alternatives reduce instrumentation, yet they are susceptible to occlusion, latency, and ambiguous visual evidence when fine-grained forces must be inferred from appearance alone [182].

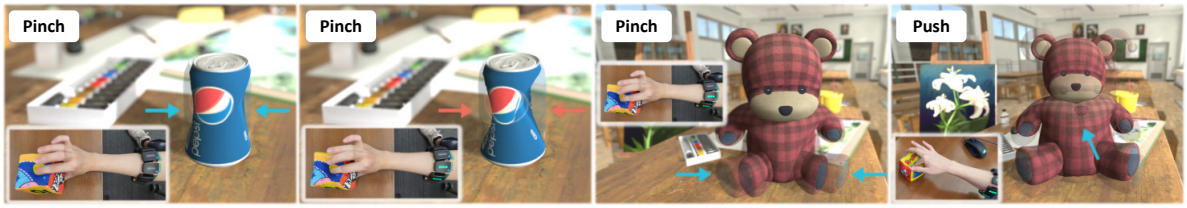
Electromyography offers a different route. Recent work has shown that sEMG can support gesture recognition, hand-pose inference, and other low-friction human-computer interfaces [51, 52, 183]. Prior work also demonstrated that forearm sEMG can be used to estimate finger-exerted forces, although most existing studies were limited in interaction variety, realism of use conditions, or application-level validation [53, 184–186]. This chapter builds on that foundation and asks a broader question: how can force decoding contribute to more natural, physically grounded, and multimodal immersive interaction?

We answer that question in two stages. The first part of the chapter develops an sEMG-based force interface around a large sEMG-force dataset, a frequency-domain autoencoder model, ForceSense1, and a series of evaluations on decoding accuracy, generalization, and system performance. Figure 7.1 illustrates the overall motivation: decoded finger force can drive materially meaningful deformation and tactile interaction in virtual environments. The early sections of the chapter therefore focus on how force information can be captured, decoded, and validated as a reliable interaction signal.

The second part extends that foundation toward a more capable and application-ready system. We revisit the sEMG-to-force problem with a transformer model, ForceSense2, that improves robustness and real-time performance for dexterous tasks. We then study how decoded force influences haptic perception and task execution in immersive environments, including elemental



(a) ForceSense1 enables users to better perceive the physical properties of virtual objects

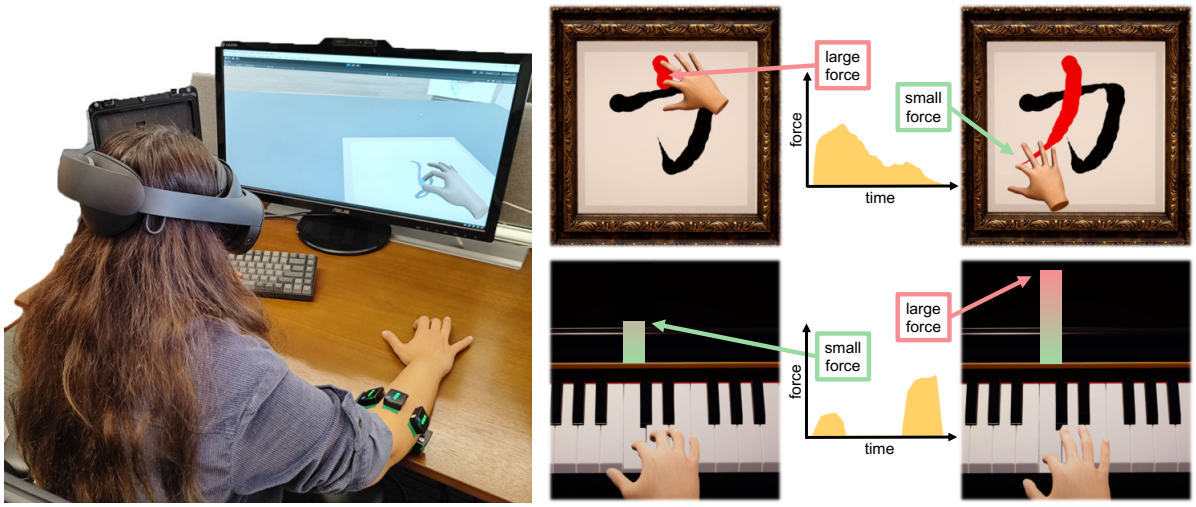


(b) Force-driven virtual interactions with ForceSense1

Figure 7.1: *Force-aware XR interface via sEMG-based force decoding.* ForceSense1 enables XR users to perform force-driven virtual interactions by decoding their forearm muscle activations into finger-level forces. The top panel illustrates the deformation of virtual balls under pressing forces of varying intensities. The beach ball is softer than the volleyball and thus exhibits larger deformation under the same force level, while the bowling ball is rigid and barely deforms within the force range of finger pressing. ForceSense1 helps XR users better perceive and distinguish the physical properties of virtual objects in a manner similar to real-world interactions. The bottom panel shows force-driven virtual interactions with ForceSense1. 3D asset credits: SbbUtutuya, Virtual Method, and TGameAssets (Unity); and TankStorm (Sketchfab).

object interactions and multimodal task learning through piano playing and calligraphy practice, as illustrated in Figure 7.2. Across these studies, the emphasis shifts from force prediction accuracy alone to the experiential consequences of force-aware interfaces: whether users can better discriminate physical workload, interact more naturally, and learn high-finesse motor tasks more effectively.

This chapter also occupies an important structural role in the dissertation. The previous chapter modeled how users react to multimodal perceptual cues. Here we move from passive perception to active manipulation. Vision and audition remain important, but they are no longer the only channels of interest; tactile perception, motor control, and physical effort become central. By decoding physical forces from muscular activities, we study how immersive systems can more



(a) An XR user operating ForceSense2 (b) ForceSense2 supports high-finesse tasks

Figure 7.2: *Force-driven virtual interactions with ForceSense2.* The left panel shows a user operating ForceSense2 to practice calligraphy in XR. The right panel demonstrates that ForceSense2 supports XR manipulation tasks that require high-finesse skills, such as calligraphy (top) and piano playing (bottom).

faithfully capture what users intend to do with their hands and how that additional modality enhances the quality of interaction.

The chapter is organized around nine sections. We begin with the biomechanical basis of finger-level force and the data-collection procedures used to capture synchronized sEMG and force signals. We then present ForceSense1, an autoencoder model for finger-force decoding, followed by evaluations of decoding accuracy, generalization, and system performance. The next two sections examine user studies on material perception and ubiquitous control via finger tapping. We then introduce ForceSense2, a transformer model that strengthens the decoding pipeline for real-time immersive interaction, and conclude with user studies on force-driven virtual object interaction and multimodal task learning. Together, these sections demonstrate how muscle-sensed force can become a practical interface primitive for physically grounded immersive computing.

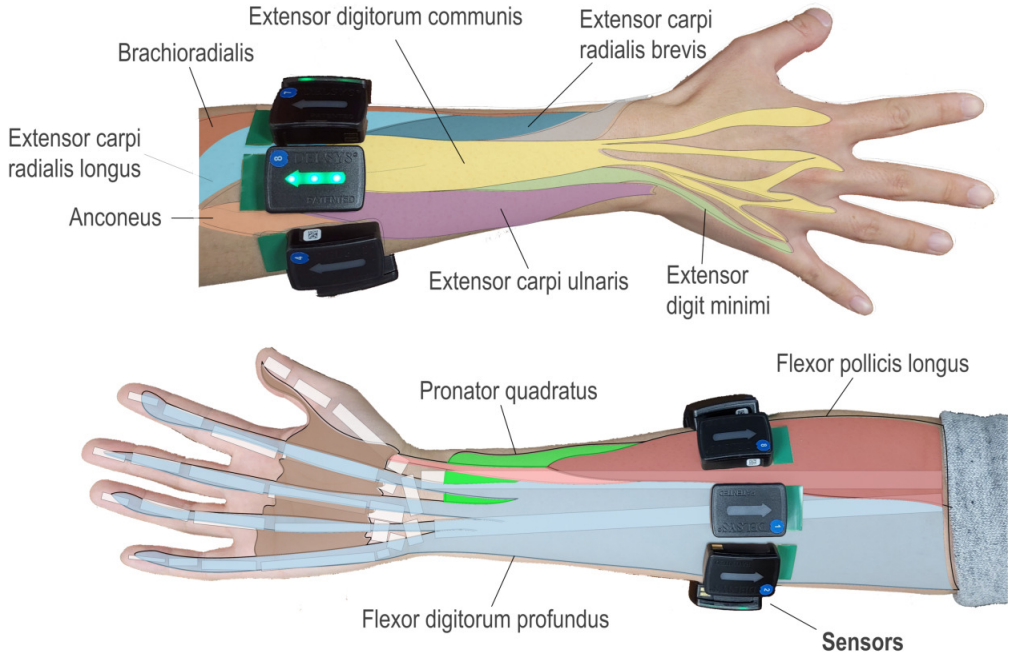


Figure 7.3: *Illustration of forearm muscles responsible for finger flexion and extension.* ForceSense1 decodes finger-level forces from forearm muscle activations via sEMG sensing, preserving the dexterity needed for high-finesse hand interactions in XR.

7.1 Human Hand and Forearm Biomechanics

To bypass the limitations of passively measuring hand-exerted interaction forces using cumbersome (and interfering) sensors such as gloves, we argue that such information can be actively decoded at the finger level from the bioelectric signals reflecting forearm muscle activations, which wireless sEMG sensors can in turn capture.

Hand-forearm biomechanical mechanism. As shown in Figure 7.3 (bottom), hands, the most dexterous limbs on the human body, exhibit high degree-of-freedom (DOF) articulations through a large number of finger joints, allowing us to perform complex and subtle interactions with the surroundings. The muscles driving this delicate articulated structure are: 1) extrinsic muscles spread over the anterior and posterior compartments of the forearm; 2) intrinsic muscles located right in the hand.

From the perspective of XR applications, users most often interact with their surroundings through bare-hand touching, pressing, pinching, and gripping [187]. The major contributing muscles in these interactions include flexor digitorum superficialis, flexor digitorum profundus, and flexor pollicis longus, all residing in the anterior compartment of the forearm. In particular, flexor digitorum superficialis controls the flexion of PIP and MCP joints for the four fingers and the wrist; flexor digitorum profundus controls the flexion of DIP joints for the four fingers as well as the flexion of MCP joints for the four fingers and the wrist; flexor pollicis longus controls the flexion of IP and MCP joints for the thumb. Figure 7.3 illustrates the anatomical structure of these forearm flexor muscles. By investigating the signals passed when invoking these biomechanics, we reason that it becomes possible to *sense and learn* hand operations from the connected forearm.

Bioelectric mechanism of muscle activation. Muscles are composed of constituent elements called motor units, and the contraction of each single muscle is managed by a specific group of motor units. On the other hand, motor units are made from more fundamental units called muscle fibers. When activated by our brain, muscle fibers within the same motor unit fire together and generate a propagating electrical potential called motor unit action potential (MUAP) via the elevation of Ca^{2+} in the sarcoplasm [188]. When placed on our skin, the sEMG sensors record such electrical signals in real-time. The various hand-object interactions that we can perform are results of varying activation patterns of the involved muscle fibers, which are themselves reflected by the recorded signals. However, how analytically or numerically the electrical signals are coupled with mechanical forces remains an open challenge, especially given the inevitable sensing noise. In the following section, we discuss our attempts toward a robust electric-mechanic signal decoding in the frequency domain.

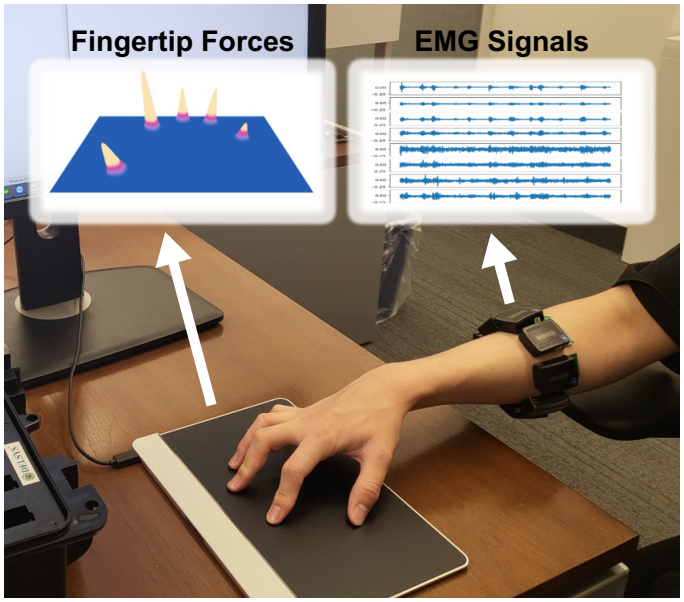


Figure 7.4: *Experimental setup for synchronized sEMG and force data collection.* During data collection, participants interacted with a Morph Sensel trackpad through a range of pressing and pinching actions while wearing eight sEMG sensors on their right forearms.

7.2 Joint sEMG and Force Data Collection

To support the two stages of this chapter, we first establish a data-collection pipeline that synchronizes forearm sEMG with finger-level force measurements. This dataset is designed to capture common pressing and pinching interactions at sufficient temporal resolution for real-time force decoding, providing the empirical foundation for both ForceSense1 and ForceSense2.

Data Collection. To collect sEMG electrical signals, we adopted 8 Delsys Trigno sEMG sensors and overlayed them on the forearm in a way that all muscles of interest are monitored. All 8 sEMG sensors are wirelessly synchronized at 2000 Hz. The captured sEMG data is streamed to a desktop computer over WiFi in real-time. To collect finger-level force data, we employed a Morph Sensel trackpad with pressure sensors arranged into a dense array. The data collection setup is illustrated in Figure 7.4. In particular, we divide the tracking area into 5 non-overlapping regions so that each fingertip only taps onto its dedicated partition throughout the data collection

process. Detected contact points with force information can then be correctly attributed to the corresponding fingers. Note that this design choice is only adopted to ease force labeling efforts and we do not assume any specific wrist/finger poses during either training or testing. Also, the dividing strategy is user- and motion-specific to accommodate the hand size and personal habit of different users. System timestamps are used for time synchronization.

Data pre-processing. As a typical bio-electric sensor, sEMG signals suffer from a certain level of measurement errors including powerline noise and other electromagnetic artifacts. Existing sEMG processing approaches typically extract and learn from statistical features in the time domain, such as mean absolute value, average amplitude change, interquartile range, etc [189]. Consequently, subtle noise or distortion may cause significant feature-space error [190].

Drawing inspirations from audio research, we compute the spectrograms of sEMG signals using short-time Fourier transform (STFT) so that high-frequency additive noise may be more distinctly isolated. Another computational advantage of learning with frequency-domain representation is that the sEMG signal from each electrode, or channel, is now a 2D array instead of a 1D time series and that we can seamlessly take advantage of powerful convolutional neural network (CNN) models for better parameter efficiency and generalization capability. Specifically, we adopt a Hanning window of size 256 sample points, which corresponds to a duration of 128ms, with hop length set to 32, to obtain 129 frequency bins. In addition, a resampling step is needed to temporally align raw force data (the sampling frequency of Morph Sensel is around 125Hz) with computed sEMG spectrograms. A nearest-neighbor-based interpolation is adopted for this purpose.

sEMG-force dataset. When users perform hand-object interactions, whether in the physical or virtual world, pressing, pushing, pinching, and holding are arguably among the most frequent actions [191]. These actions allow users to not only better perceive surrounding objects, especially their physical properties, but also pick them up for further interactions. Based on their respective

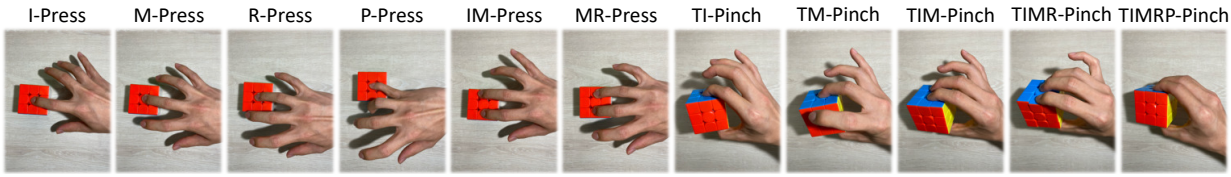


Figure 7.5: *Hand–object interaction types used to construct the sEMG–force dataset.* The capital letters before the hyphen, T, I, M, R, and P, denote the thumb, index, middle, ring, and pinky fingers, respectively.

force exertion mechanism, we partitioned these actions into two representative groups: pressing/pushing and pinching/holding. Eleven common finger combinations were selected for data collection purposes, with six for the former and five for the latter. This set of actions, which we call action set $\mathcal{A}$, is summarized in Figure 7.5.

To build our sEMG-force dataset, we recruited 9 participants (ages $21-30$, 5 females, 4 males). Following the data collection and pre-processing pipeline, we conducted three collection sessions with each subject, capturing 30 seconds of data for each action during each session. In total, each participant contributed 990-seconds of time-synchronized sEMG and force data. For each subject, 2 randomly selected sessions (out of 3) were earmarked for the construction of the training set. The remaining session was withheld and only used for evaluation. When performing pinching actions during data collection, participants were instructed to keep their four fingers over the trackpad and their thumbs below the table, so that they could pinch the ensemble of trackpad and table in a natural manner. Besides, they kept the resultant force stable and balanced (i.e., $\sim$ 0N). The ground-truth forces for the four fingers were directly recorded by the trackpad, and the force for the thumb was derived as the additive inverse. With the importance of data coverage in mind, all subjects were instructed to randomize their force intensity level within the natural range of each action. In addition, a random spacing in time was enforced between adjacent interactions, so that neural network models do not over-fit to unintended temporal features. The sEMG-force dataset contains light touch less than 1N and firm press up to 30N, covering the typical functional force range of human fingers [192]. In particular, the maximum force for the five fingers in Newton, from the thumb to the pinky finger, are 29.8, 24.4, 25.6, 20.4, and

14.7. The mean/standard deviation/interquartile range are 10.1/8.1/13.9, 5.8/4.6/7.4, 5.7/4.7/7.6, 3.4/3.1/4.0, and 2.6/2.5/3.2. Our CNN model predicts force values in $[0, 30]$, which is configured through the force upperbound $F_{\max}$.

7.3 ForceSense1: An Autoencoder Model for sEMG-to-Force Decoding

A main roadblock for sEMG sensors is the well-known challenge of aligning the electrodes exactly on muscles. For instance, as seen in Figure 7.3, sensors may commonly cross-ride on or fall in the gap between the underlying interwoven muscle bundles. As a result, although the activation information of all target muscles are captured by sEMG sensors, directly assigning the electric signals to individual muscle-group and joints becomes unrealistic. To robustly recover finger-wise force information from raw sEMG data, we resort to the data-driven paradigm and adopt powerful neural network models to learn this highly non-linear correlation between forearm sEMG signal and finger-level forces.

Model architecture. While recurrent neural network (RNN) has been a common practice for sequential data learning, recent advancements in audio learning have shown that deep CNN models with properly processed input data are capable of delivering better performance in some cases, thanks to their highly efficient parameter usage which allows for very deep design [193]. We are inspired to exploit convolutional filters to extract deep features from the 2D spectrograms.

While our input data points live in a high-dimensional space (129), those features containing the semantic information of finger-wise forces are embedded in a subspace of much lower dimension. To efficiently extract relevant information and mitigate overfitting to training data, we employ an encoder-decoder architecture to enforce a low-dimensional latent space. Also, we only feed the 64 low-frequency components from the spectrograms to the encoder model to remove high-

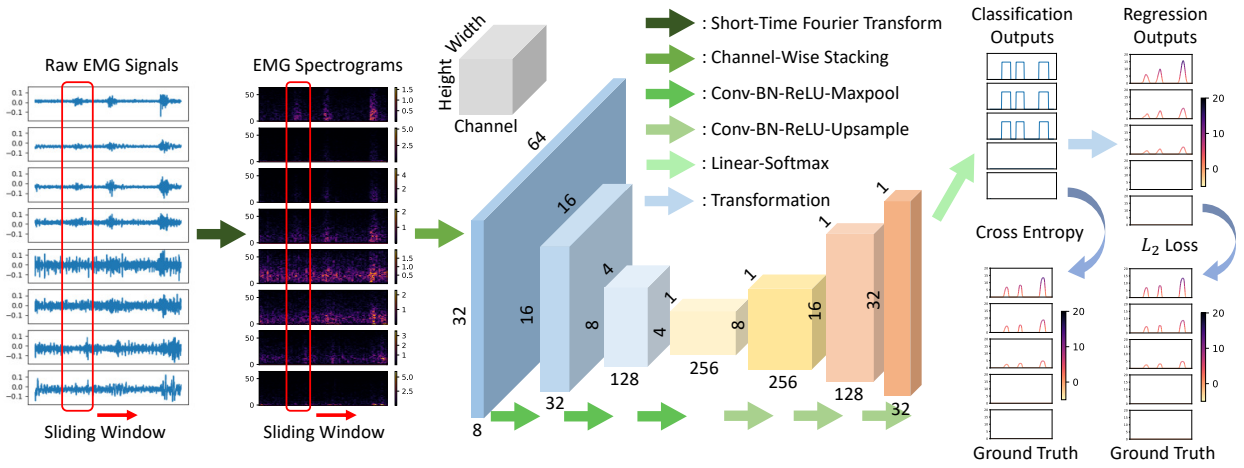


Figure 7.6: *Model architecture of ForceSense1.* To optimize for parameter efficiency and resilience to data noise, we transform raw sEMG signals into frequency domain via STFT, treat the resulting spectrograms as multi-channel images, and employ a lightweight CNN model with bottleneck design. Training is performed using a customized classification-regression joint loss tailored to the task of force estimation. When put into action, ForceSense1 uses a fixed-size sliding window to retrieve the latest frames from wirelessly streamed sEMG signals and decodes finger-level forces in real-time.

frequency data noise and allow for accelerated performance. In addition, we provide the model with sequential data from a long time interval (32 consecutive frames in the spectrograms, which correspond to 624ms raw sEMG data) instead of a single frame to let it exploit information from previous frames and better satisfy temporal constraints. The input data size is thus (N, C, L, S), where N, $C = 8$, $L = 32$, and $S = 64$ denote the batch size, number of sEMG channels, input sequence length, and number of sEMG frequency components, respectively. The encoder model consists of repeating Convolution-BatchNorm-ReLU blocks, with each block followed by a 2×2 Maxpool layer for downsampling in time and frequency dimensions. Similarly, the decoder model also consists of repeating Convolution-BatchNorm-ReLU blocks, with each block followed by a 2×2 bilinear Upsample layer for increasing time dimension. After that, the decoder output is transformed by a linear layer in the channel dimension to match desired force outputs, e.g. 5 values for 5 finger-wise forces. The output data size is thus $(N, F, L, 1)$, or (N, F, L) with the fake frequency dimension squeezed out, where F denotes the pre-defined number of force components. All convolutional layers have kernel size 3×3. When putting the model in action, spectrograms

of streamed sEMG signals are computed on the fly and a sliding window of length $L = 32$ feeds the latest data to the model for real-time inference. Detailed model architecture, data flow and input/output dimension at each layer are illustrated in Figure 7.6.

Joint classification and regression. Estimating continuous finger-wise forces, by its nature, is a regression problem, and it is natural to adopt common regression losses, such as L_1 or L_2 loss, as the objective function. However, the muscle-generated forces in interactive scenarios have two unique patterns: humans apply forces only sparsely in the real-world; and the variance of force levels is commonly high, ranging from light touches to hard pushes. In practice, regression-based learning oftentimes tends to predict non-zero values (false positive when we do not generate forces) or over-smooth low-amplitude values (false negative for light forces). For our targeted XR applications, this seemingly small estimation error can lead to visually noticeable artifacts and largely compromise user experience (e.g., causing constant vibrations on objects or producing no reaction on low-force touches).

A naïve solution to this problem is to set a cut-off threshold such that the estimated values below it are treated as zero. Although this modification enables zero-value output, tweaking the threshold can be unworkable in practice and the performance is still barely satisfactory as will be shown in Section 7.4.1. To address this issue, we introduce a classification loss to better differentiate between sEMG sequences with and without forces. Specifically, for each time frame t and each finger i, the model outputs a value $p_i^t \in [0, 1]$ indicating the probability of that finger applying force at that time frame. A cross entropy loss L_c is employed to train the model for this force/no-force binary classification task. On top of p_i^t, we compute the predicted force as $\hat{F}_i^t = 2F_{\max} \cdot \max(0, p_i^t - 0.5)$, where $F_{\max}$ denotes the force upper-bound and defines the predicted force range. A L_2 loss is then employed to train the model for force regression.

$$L_c = \frac{1}{T \cdot I} \sum_{t=1}^{T} \sum_{i=1}^{I} y_i^t \cdot \log p_i^t + (1 - y_i^t) \cdot \log(1 - p_i^t) \tag{7.1}$$

$$L_r = \frac{1}{T \cdot I} \sum_{t=1}^{T} \sum_{i=1}^{I} \|\hat{F}_i^t - F_i^t\|^2 \tag{7.2}$$

where y_i^t and F_i^t denote the ground-truth force label and value, respectively, for finger i at time frame t.

A hyper-parameter λ is introduced to balance between classification and regression, and the overall loss L takes the form:

$$L = L_c + \lambda \cdot L_r \tag{7.3}$$

The joint loss above is designed such that, for finger i at time frame t: when $p_i^t \in [0, 0.5]$, we have $\hat{F}_i^t = 0$, only L_c takes effect and the model focuses on correcting wrong classifications; when $p_i^t \in (0.5, 1.0]$, we have $\hat{F}_i^t = 2F_{\max} \cdot (p_i^t - 0.5) \in (0, F_{\max}]$, L_c and L_r together push the model towards the joint classification-regression goal. As a result, we are able to not only get zero-value outputs when there is no force, but also prioritize classification over regression at the beginning stage of training, since the estimated force value will be useless if the predicted class is wrong in the first place. Note that p_i^t is exploited to both differentiate between no-force and force (classification with threshold $p_i^t = 0.5$) and compute the predicted forces $\hat{F}_i^t$ (regression).

7.4 Finger-Level Force Decoding with ForceSense1

7.4.1 Force Decoding Performance

Experimental setup. Before considering how our scheme applies to specific or new users, we first evaluate the performance of our model in a user-independent setting, where a single model is trained and shared by all users who contributed data. In particular, the entire training set was used to optimize the model against the joint loss defined in Equation (7.3) for 30 epochs. An Adam optimizer [125] with constant learning rate of $1e-4$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ was adopted. A weight decay factor of $1e-4$ was enforced to mitigate over-fitting. As a post-processing

step, we applied a Gaussian filter of window size 10 to the sequence of predicted force values for temporal smoothing. For ablation purposes, we also trained the model using regular L_1 or L_2 regression loss only, and cut off predicted force values below a small pre-defined threshold. We used PyTorch [194] to implement all our models as well as to perform training and evaluation.

Evaluation metrics. To assess the performance of our CNN model in estimating fingertip forces, we adopted three quantitative metrics: (1) Classification Accuracy; (2) Normalized Root Mean Squared Error (NRMSE); (3) Coefficient of Determination, R^2. The model's performance in determining whether a finger exerts force or not at a particular time frame is evaluated by the classification metric, and we only count the model's predictions for a time frame as correct if all five fingers are correctly classified. Using the same notations from Section 7.3, we have:

$$\text{NRMSE} = \frac{1}{F_{\max}} \sqrt{\frac{1}{T \cdot I} \sum_{t=1}^{T} \sum_{i=1}^{I} (F_i^t - \hat{F}_i^t)^2} \tag{7.4}$$

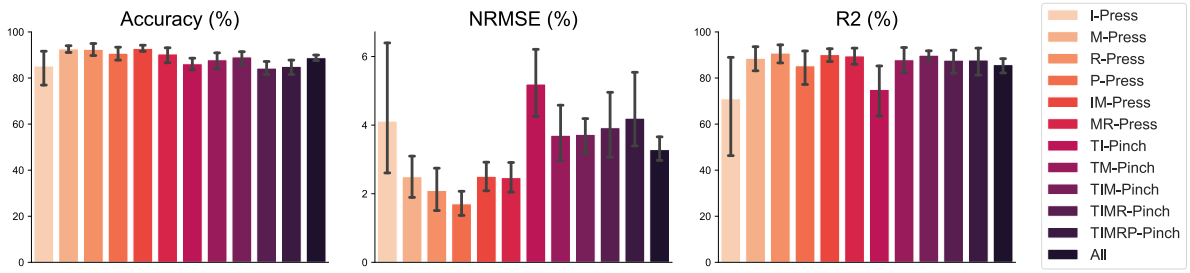
$$R^2 = 1 - \frac{\sum_{t=1}^{T} \sum_{i=1}^{I} (F_i^t - \hat{F}_i^t)^2}{\sum_{t=1}^{T} \sum_{i=1}^{I} (F_i^t - \bar{F}_i^t)^2} \tag{7.5}$$

where $\bar{F}_i^t$ gives the mean of F_i^t.

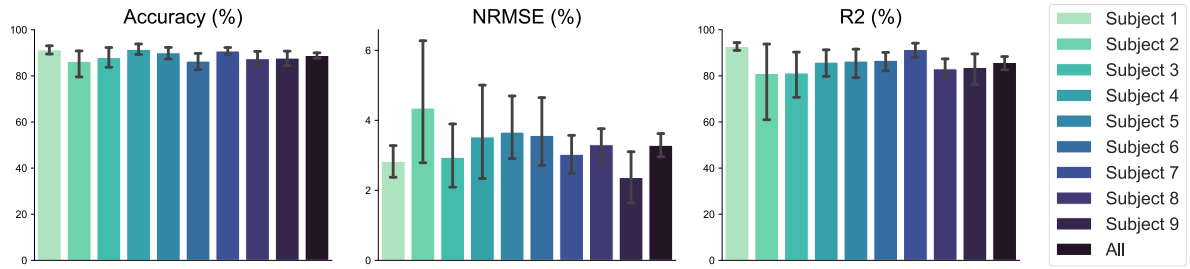
Results. The action-wise and subject-wise performance of the user-independent model is summarized in Figure 7.7. The overall accuracy, NRMSE, and R^2 are 88.83±6.13%, 3.29±1.76%, and 85.82±14.96%, respectively. On the action side, the model has the highest performance for

Table 7.1: *Ablation study on the loss function and temporal smoothing.* Comparison with regular L_1 and L_2 losses and a no-smoothing variant.

Metric	L_1/L_2 regression	No Smoothing	Ours
Accuracy	85.68% / 85.12%	88.83%	**88.83%**
NRMSE	4.56% / 4.34%	4.02%	**3.29%**
R^2	81.89% / 82.21%	83.59%	**85.82%**



(a) Action-wise and overall performance of the user-independent model



(b) Subject-wise and overall performance of the user-independent model

Figure 7.7: *Quantitative force decoding evaluation of the user-independent ForceSense1 model.* The top panel shows the action-wise performance of the user-independent ForceSense1 model in decoding finger-level forces, with 95% confidence intervals overlaid. The bottom panel shows the subject-wise performance across the nine participants.

ring finger pressing, with 92.47±3.89% accuracy, 2.10±0.99% NRMSE, 90.84±6.53% R^2, and the lowest for index finger pressing, with 85.21±11.37% accuracy, 4.12±3.04% NRMSE, 70.99±34.95% R^2. On the subject side, one-way repeated measures ANOVA gives $F_{1,8} = 1.28$, $p = 0.26$ for accuracy, $F_{1,8} = 1.18$, $p = 0.32$ for NRMSE, and $F_{1,8} = 0.82$, $p = 0.59$ for R^2, indicating minor utility discrepancy among subjects. Furthermore, the results of our ablation study are shown in Table 7.1, validating the effectiveness of the proposed joint classification-regression loss and temporal smoothing.

Discussion. The results above demonstrate the feasibility of accurately decoding finger-level forces from forearm sEMG signals and building robust predictive models that can be shared by multiple users. The statistical significance also suggests that the proposed scheme has the potential of being extended beyond an experimental setting and to more general application scenarios.

In addition, it is worth noting that such performance holds under the existence of real-world challenges, such as variations across users in sensor positioning, forearm muscle size, forearm hair thickness, etc. The model is resilient to various discrepancies among users, capable of capturing generalizable sEMG-to-force patterns, and achieves utility fairness for users, as evidenced by the ANOVA analysis above, all the while maintaining favorable overall performance.

While these results are statistically rewarding, a remarkably large amount of data is required from each user to support satisfactory performance in practice. Specifically, each participant contributed 22 sEMG-force joint sequences to the training set for the experiment above, which amount to 11 minutes of data. Consider also that there are other inevitable preparations, such as device setup, session break, data pre-processing, and model training. Such delay may become a roadblock for many XR applications in practice. To deploy our neural interface in consumer-level applications, more time-efficient training is essential. This aspect will be addressed in Section 7.4.3.

7.4.2 Comparison with Vision-Based Methods

Prior work in the literature have explored vision-based solutions to body force estimation, such as inferring contact forces from the dynamics of hand-object interactions using RGB videos [195–199] and predicting finger-level forces from the color changes in fingernail imaging [200–206]. Similar to our approach, a major benefit of vision-based solution is to bypass on-hand force sensing units. Among these solutions, those based on fingernail imaging also have the potential of delivering accurate and flexible per-finger force estimation for XR applications involving complex hand-object interactions. In this experiment, we compare the accuracy and robustness between our method and three recent vision-based methods [204–206].

Experimental setup. Due to the challenges of reproducing the identical hardware prototype of data acquisition as in [204–206], we evaluated our method under the setting adopted by them and compared with their reported performance metrics. In particular, [204–206] only considered

single-finger grasping actions and evaluated their method using the index, middle, and ring fingers. To accommodate their evaluation setting, we separated out the partition corresponding to these three single-finger actions from our sEMG-force dataset, i.e., 3x30 seconds (three sessions) of time-synchronized sEMG and force data for each subject and each of the three fingers. Two randomly selected sessions were used to train a single user-independent model, while the remaining session was used to evaluate the trained model.

Results. The finger-level and overall performance of force estimation, as evaluated by NRMSE, is shown in Table 7.2. Our method outperforms the three vision-based baselines by 2.5%, 2.2%, and 1.2% NRMSE on average, respectively. The advantage is especially noticeable in performance for the ring finger, with our method's NRMSE being less than a quarter of [204] and a third of [205, 206].

Discussion. Compared to our sEMG-based solution, which actively decodes finger-level forces from the controlling muscles' activities, vision-based methods rely on passive observations and are thus highly sensitive to the variations in external factors, such as ambient occlusions, viewing angles, and lighting conditions. Such degrading effects are more problematic for applications involving complex hand-object interactions or real-wild scenarios. On the contrary, our sEMG-based solution has shown high robustness to such mentioned issues by its nature. Besides the vulnerability to environmental factors, fingernail-imaging-based methods are also limited in their functional force range, since the variations in fingernail color get less and less detectable as the

Table 7.2: *Force decoding performance comparison with vision-based methods.* Three fingernail-imaging approaches are adopted as baselines [204–206].

Method	Index	Middle	Ring	Mean
[204]	6.1%	5.3%	10.1%	6.2%
[206]	6.1%	5.4%	9.0%	5.9%
[205]	5.7%	4.2%	8.2%	4.9%
Ours	**4.7%**	**3.7%**	**2.4%**	**3.7%**

force intensity increases. The typical functional range for these methods, as evaluated in [204–206], is around 10N. By contrast, our sEMG-based solution is more scalable in terms of force intensity and can robustly estimate forces up to 30N.

7.4.3 Individualization and Generalization

The analysis of user-independent training in Section 7.4.1 reveals the need for *time-efficient generalization*. In this section, we investigate how this goal is achievable by extending a pre-trained model for new users using only minimal amounts of data from them. This procedure is commonly referred to as calibration or individualization in human research. Calibration is crucial for machine learning on sEMG data since large natural variations exist among different people's muscle-to-sEMG patterns. Such discrepancies lead to the well-known generalization challenge that a machine-learned model trained on sEMG data commonly fails if directly applied to an unseen user [51]. Therefore, we decide to perform transfer learning to reduce the data requirements for deploying our model to a new user while maintaining satisfactory prediction performance.

Experimental setup. To evaluate time-efficient generalization via transfer learning, we adopted an experimental setup similar to cross-validation practices for machine learning tasks. Specifically, we first treated subject 1 (S1) as the new user and optimized the model using two sessions of data from each of the other eight subjects. Next, we fine-tuned the resulting model using a portion of data randomly selected from S1's first session, varying from 10% to 50%, to calibrate it into a user-specific model dedicated to S1. Note that 10% session corresponds to 33-second data. Adam optimizer [125] with constant learning rate of $5e-5$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ was adopted. A weight decay factor of $1e-4$ was enforced to mitigate over-fitting. We cycled through the role of new user with each subject to complete the experiment.

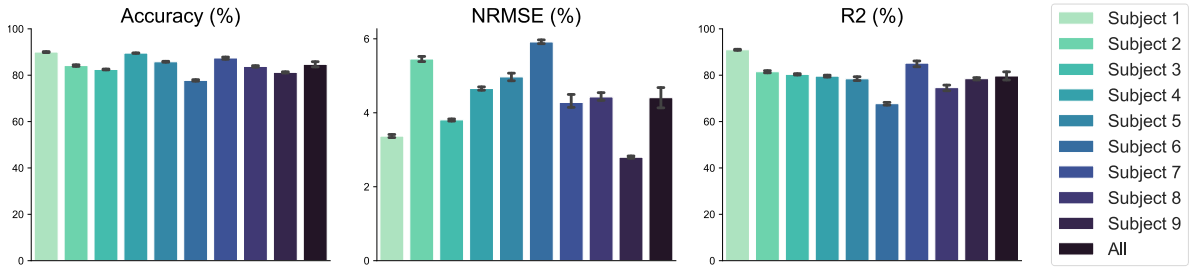


Figure 7.8: *Quantitative force decoding evaluation of user-specific ForceSense1 models.* Each user-specific model is calibrated using 165-second sEMG-force data, demonstrating that lightweight calibration is sufficient to obtain strong personalized performance.

Results. The performance of a user-specific model transferred using 165-second data (50% of each subject's first session), as measured by classification accuracy, NRMSE, and R^2, is summarized in Figure 7.8. At least 81.23% accuracy was consistently observed for all subjects' calibrated models except S6, whose model showed 77.77% accuracy. S1/S4's models achieved over 89.54% accuracy, surpassing the overall performance of the user-independent model with much less training data. The NRMSE metric revealed larger gaps across subjects, which ranged from 2.80% to 5.91%. R^2 is mostly above 75%, with the exception of S6's model yielding 67.73%.

To investigate the minimal amount of data required for calibrating the model towards reasonable utility in practical applications, we analyzed the trade-off between data volume for calibration and resulting model's performance. Figure 7.9 visualizes the calibrated models' average performance gain as a function of sEMG-force sequences' total length in time. All subjects' models demonstrated rapid improvements as the calibration kicked off and attained 83.33±4.47% accuracy, 4.74±1.09% NRMSE, and 76.55±7.88% R^2 with 66-second data only. The growth rate of mean performance then slowed down, and accuracy/NRMSE gradually plateaued when the duration of data exceeded 150 seconds. Remarkably, these results verify the data efficiency of transfer-learning-based individualization, as evidenced by various accuracy metrics. Taking S8's calibrated model (using 99-second data) as an example, we show a visual comparison between model-predicted and hardware-sensed force values for a randomly selected sEMG sequence from their evaluation

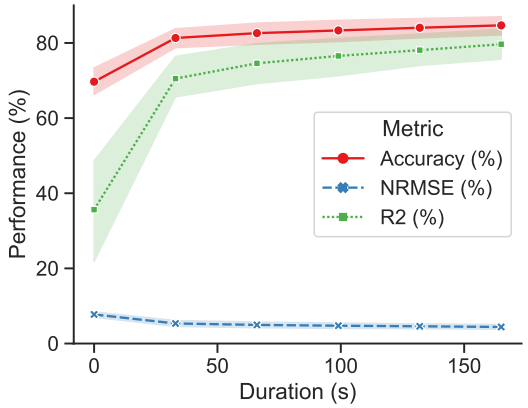


Figure 7.9: *Data-efficient calibration of user-specific models via transfer learning.* The performance of the user-specific model, measured by classification accuracy, NRMSE, and R^2, is plotted as a function of the amount of data collected from a new user for transfer learning. The translucent band around each curve indicates the 95% confidence interval.

session in Figure 7.10. Despite the variations in force intensity and temporal spacing, predicted force values generally aligned well with the ground truth, except for a few slight false positives for the ring finger.

Discussion. Exploiting transfer learning techniques, we demonstrated the feasibility of effectively adapting existing models into dedicated ones for previously unseen users using very limited data from them. Further, as indicated by the above analysis and visualized in Figure 7.9, the high-precision generalization is achieved with simple and rapid (less than 2 minutes) calibration for new users.

These findings also circle back to our goal of improving physicality for XR interactions in two significant ways. First, our system can harness users' natural abilities and predilections for manipulating things they encounter, thereby expanding the space for developers to create XR experiences that map to real-world scenarios and user behaviors. Importantly, we show that this is achievable for any user, with minimal retooling. Second, as shown in the following section, our system is responsive with low latency. This is crucial, in particular, if we consider that the visual

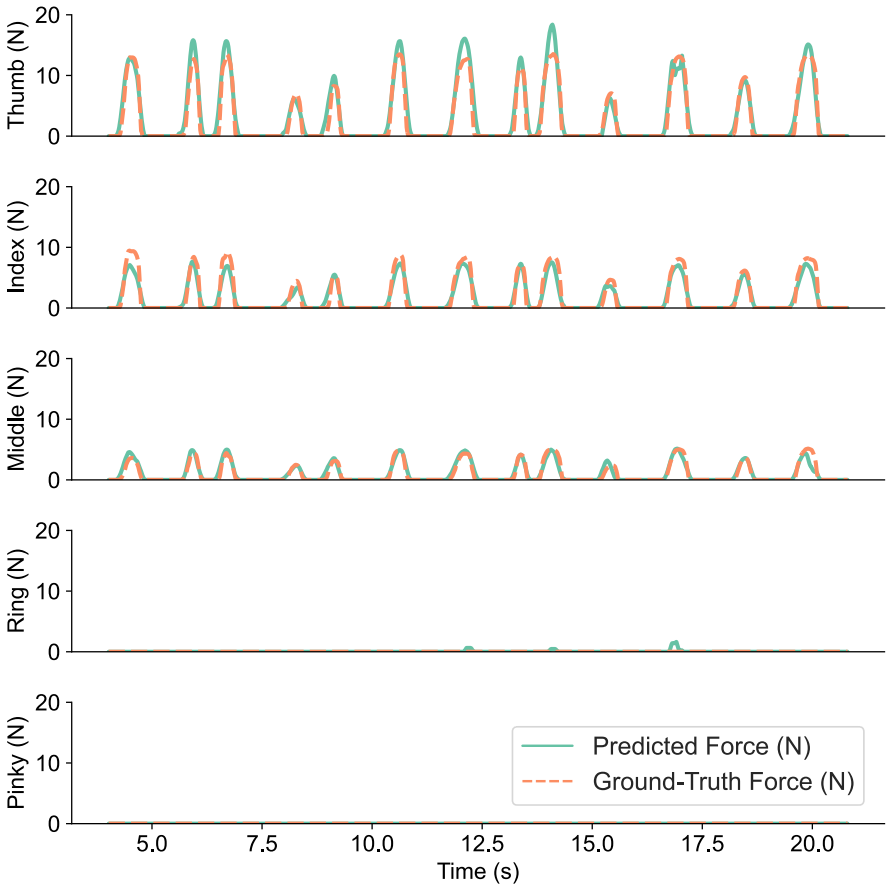


Figure 7.10: *Comparison between predicted and ground-truth finger-level forces.* The user-independent ForceSense1 model is calibrated for subject 8 using 165 seconds of data from one of their training sessions and evaluated on a randomly selected sequence from their evaluation session.

components in XR now routinely refresh at 90 Hz. Any system designed to communicate bodily forces with the virtual environment needs to be agile in timing and nimble in response to data.

7.4.4 System Performance

Thanks to the moderate scale and bottleneck design of our CNN model, our system's storage and computing requirements are minor for modern PC hardware. The compact model only contains 1.26M 32-bit floating point parameters (≃5MB memory). For the sEMG sequence streamed at a particular time frame, i.e., 1248 eight-channel sEMG samples, our model only generates around

29.19M Multiply-Accumulate Operations (MACs). Note that while our model requires an sEMG sequence of 1248 samples (624ms) as input, these data are only retrieved from history to make predictions for the current time step. When applied in practice, our system's latency performance has two important interfering dimensions: 1) *runtime speed*, i.e., the time needed to complete the force predictions on an sEMG sequence of 1248 samples; 2) *reaction latency*, i.e., the duration between sEMG data generation and force prediction.

As an implementation detail, we performed GPU parallelization with two consecutive sEMG sequences that differ by 32 samples and achieved ≃1.2-1.4ms inference time using a GTX TITAN XP GPU, thus obtaining an approximate 0.7ms latency (i.e. over 1000FPS). Note that this result characterizes the system itself rather than the force output, which is also determined by the spectrogram frame rate. With this setting, the system shall wait for both sequences to arrive, introducing an additional 16ms (32 sEMG samples) idle time. That is, although GPU parallelization accelerates runtime speed, it also introduces extra reaction latency. Together with the wireless sEMG data transmission latency (2ms) and model inference (0.7ms), our system achieves an overall reaction latency of ≃18.7ms, sufficient for most real-time XR applications [207].

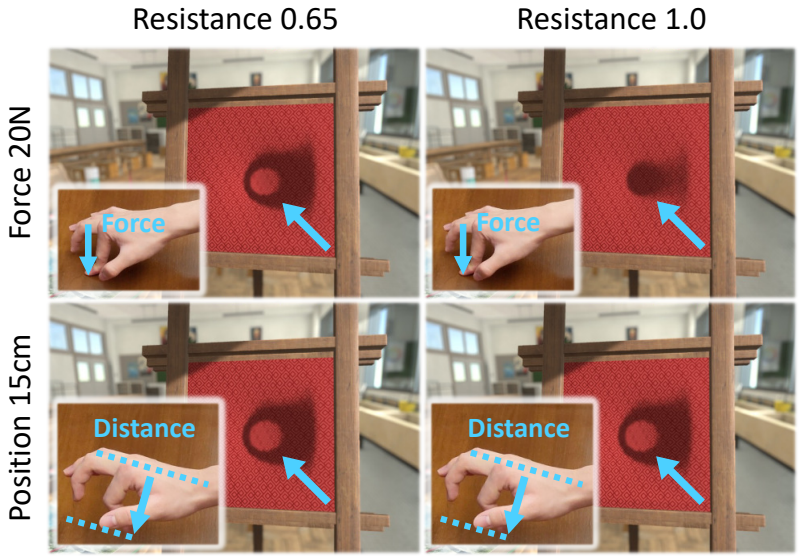
7.5 User Study: Enhancing Material Perception in Virtual Environments

A key aim of XR is to create virtual experiences for users as if they were in a physical environment. When interacting with objects in the physical world, we perceive their material properties, such as elasticity and stiffness, through a combination of haptic and visual feedback [208]. To recreate such perceptual realism in virtual environments, it is essential to precisely drive virtual objects' motions and deformations via users' muscular forces. We hypothesize that interfaces with such capability may significantly enhance human perception of virtual objects' material properties. In

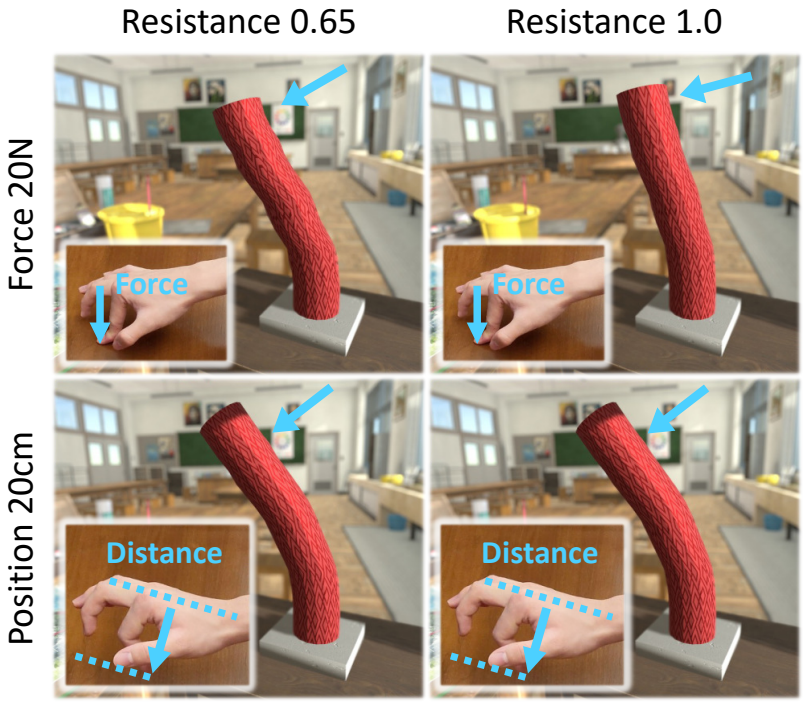
this study, we evaluate to what extent our system, as a real-time force-aware interface, advances toward this goal.

Participants, setup, and calibration. We recruited 12 subjects (ages 20-35, 6 female) to participate in the study. A calibration procedure was performed for each subject before starting the experiment by collecting 1 minute of sEMG-force data from him/her to customize the pre-trained user-independent model (as described in Section 7.4.3). This calibrated model was then used to estimate finger-wise forces on the sEMG signals sensed from that subject in real-time. Estimated force values were communicated to a Unity program via the ZeroMQ library. This pipeline allows for direct application of estimated forces to virtual objects in real-time via physical simulation. During the study, the subjects, wearing an Oculus Quest 2 head-mounted display (HMD), remained seated and were free to observe a virtual scene. They interacted with virtual objects in their field of view through unconstrained movements of their forearms, hands, and fingers.

Stimuli. As shown in Figure 7.11, the visual stimuli were two geometric primitives (elastic sheet and rod) that are soft and deformable. To enable real-time softbody simulation for low-latency interaction on portable XR headsets, we employed an efficient XPBD [209, 210] implementation by *Virtual Method Studio* [211] and only used low-resolution particle models of the virtual objects. It should be noted that simulators' efficiency is orthogonal to the accuracy of model-predicted muscular forces. Therefore, our model can be readily incorporated into *any* simulation system. We adopted deformation resistance of elastic materials as the proxy to represent stiffness with range $[0, 1]$. Deformation resistance measures a physical material's ability to resist externally loaded forces. In this study, we aim to identify participants' discriminative thresholds of virtual objects' stiffness under varying conditions. To avoid visual cues biasing the results, all objects were rendered with identical material and texture, regardless of their physical properties.



(a) Elastic sheet with **FORCE** (top) and **POSITION** (bottom)

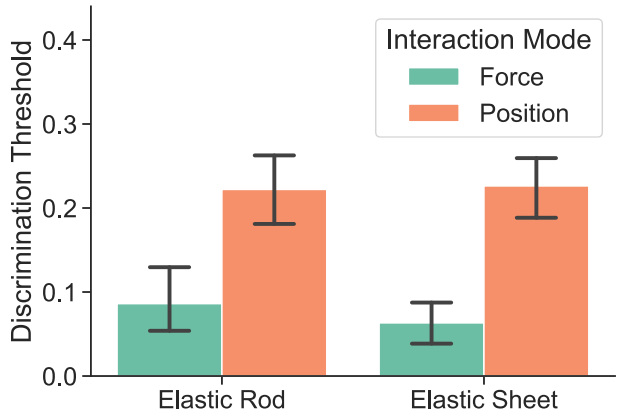


(b) Elastic rod with **FORCE** (top) and **POSITION** (bottom)

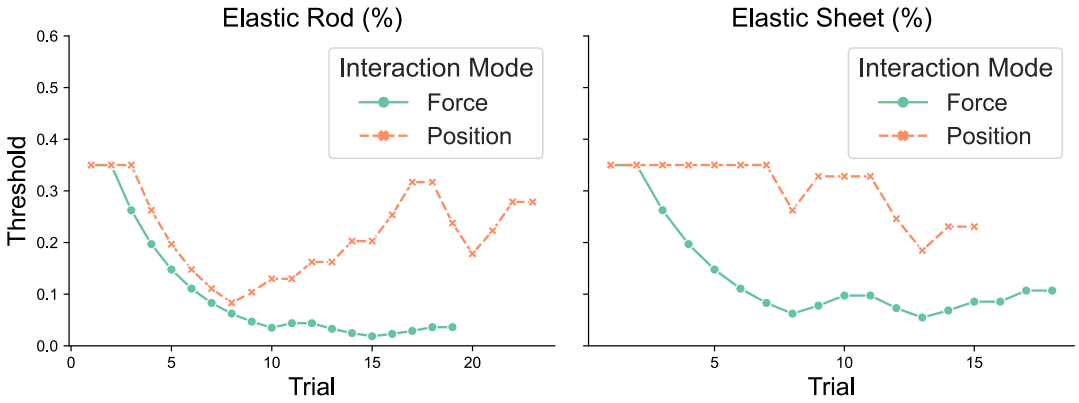
Figure 7.11: *Stimuli used in the XR material perception user study.* The top panel illustrates the responses of two visually identical elastic sheets with different material stiffness to inputs from **FORCE** (top) and **POSITION** (bottom). Compared with **POSITION**, the sheets exhibit more natural and pronounced differences under the same inputs from **FORCE**. The bottom panel shows similar results for an elastic rod. 3D asset credits: SbbUtutuya and Virtual Method (Unity).

Conditions. For evaluation purposes, all subjects were instructed to employ two interaction methods in sequence during the study (the order was random). Besides our data-driven system for force-enabled interaction (**FORCE**), we also included position-based interaction (**POSITION**) for comparison. Specifically, **POSITION** is a commonly adopted solution in commercial XR systems that lacks force information. It allows users to modify virtual objects' position, orientation, and shape by colliding their hands with the objects. For **FORCE**, users interacted with virtual objects by manipulating physical proxies while our system estimated their muscular forces. These forces were then used to deform the two virtual objects, including indenting the sheet's center and bending the rod's top. For **POSITION**, we leverage the hand tracking capability of Oculus Quest 2 to estimate the flexion level of users' index finger (distance from the index fingertip to the palm plane) and use it as input for the identical interaction as **FORCE**. Specifically, this value determines the indentation depth for the sheet's center and the bending level for the rod's top. Figure 7.11 illustrates these interactions. Notably, while physical proxies that resemble the virtual objects will enhance users' experience with **FORCE**, **POSITION** does not benefit from them. To avoid users favoring **FORCE** due to irrelevant features of the physical proxies, we intentionally used a hard and flat table.

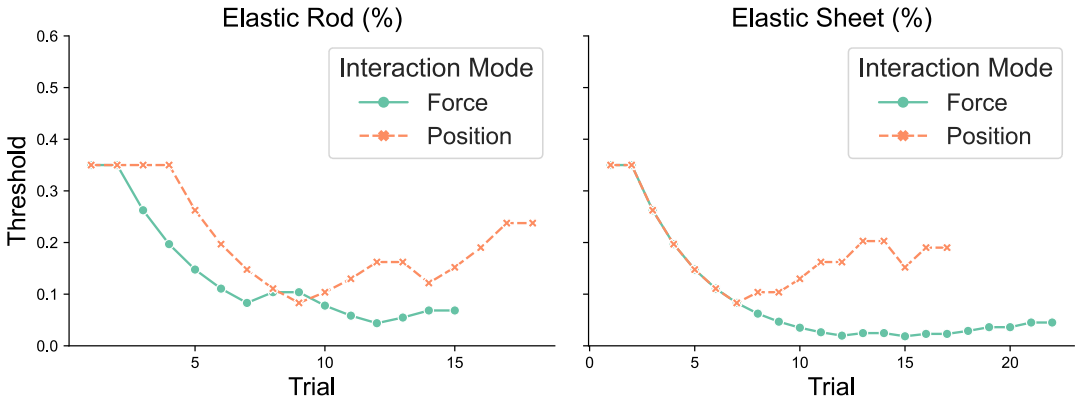
Task. To measure participants' (perceptual) discriminative threshold of material stiffness, we employed a psychophysical task as a two-interval forced choice (2IFC) with a 1-up-2-down staircase procedure (5-reversal to confirm convergence thus termination). For softer and stiffer objects, the experiment started with deformation resistance equal to 0.65 and 1.0, i.e. a threshold of 0.35. Each time the threshold got updated, it was incremented or decremented by a quarter of the current threshold, and the two deformation resistance values were updated such that their mean was unchanged. Specifically, the two interaction conditions (**FORCE**/**POSITION**) were sequentially presented to the user for consideration (with a random and counter-balanced order). During the experiment, the participants were instructed to freely interact with the corresponding stimuli and



(a) Discrimination thresholds



(b) An example 2IFC trial from subject 8



(c) An example 2IFC trial from subject 10

Figure 7.12: *XR material perception user study results.* The top panel shows the aggregated discrimination thresholds across 12 subjects for **FORCE** and **POSITION**. Error bars indicate 95% confidence intervals. The middle and bottom panels illustrate example 2IFC trials from two subjects. The consistent decreasing trend under **FORCE** indicates a higher confidence level in their 2IFC decisions.

then indicate (using a keyboard) which one of the two stimuli appeared stiffer. They observed the deformation pattern along with the proactive intervention. After each trial, the participants chose one of the two stimuli that appeared stiffer. Each 2IFC trial took 5 seconds. A warm-up session was first performed to allow each individual user to familiarize with the stimuli and the interaction design. For each participant, the entire experiment took about half an hour. The number of trials (ranging from 25 to 38) depended on the speed of the staircase convergence.

Metrics and results. Figure 7.12 visualizes our statistical results. The mean discriminative thresholds of **FORCE**/**POSITION** were 0.09±0.07, 0.22±0.07 for the elastic rod, and 0.06±0.04, 0.23±0.07 for the elastic sheet, indicating 61.3% and 72.1% improvements with **FORCE**, respectively. One-way repeated measures ANOVA shows that the effects of interaction method are statistically significant: $F_{1,11} = 20.99$ $p = 1.46e^{-4}$ for the elastic rod and $F_{1,11} = 46.79$ $p = 7.16e^{-7}$ for the elastic sheet.

Discussion. We designed our psychophysical experiment to test whether users could quantitatively perceive realistic soft objects and their material properties in XR. We used primitive geometries and their natural articulated abilities to explore, examine, and assess them through force-based interaction with the forearm, hand, and fingers. In other words, we wished to test whether users could simply enter an XR scene, start to prod and poke things in that scene, and leave with a sense that the objects responded with realistic physics.

The results showed a statistically significantly lower discrimination threshold while participants interacted with virtual objects with **FORCE**. That is, the force-visual correlated interaction facilitated significantly more realistic perception of virtual objects' physical characteristics when users were engaged in free manipulation within the virtual world. We regard this as a significant proof of concept for our approach. Consider that, in the real world, humans spend much of their infancy working out how to muster the forces available to them in their arms, hands, and fingers,

through ongoing trial and error with the things that they encounter. In essence, we capture the small electrical signals that human muscles cast as they put their skills to use, and we are able to use these signals as indices for machine-learning what that might mean in physics. Asking and answering how our human users believe that the physical response our system yields are realistic-seeming establishes the perceptual foundation of various new possibilities of interfaces.

7.6 User Study: Ubiquitous Control via Finger Tapping

Traditional computers are typically equipped with dedicated control devices such as mouse and keyboard. However, the ultimate goal of XR platforms is a transformative natural and ubiquitous control. To this end, researchers have recently attempted to infer user intention from natural modalities, such as vision [212–214], acoustics [215–217], radar [218, 219], and Wi-Fi [220, 221]. Despite their support for eyes-free control, these methods are susceptible to environmental interference, such as occlusions and noise, and may suffer from performance decay in complex environments. By contrast, sEMG-based solutions electronically tracks users' forearms as input devices. In this experiment, we validate our method's performance while being applied to detect "click" actions, which are identified as any finger's tapping.

Experimental setup. All ten fingers from both hands are included. One male subject participated in the study. Following the data collection and pre-processing pipeline (Section 7.2), we conducted four collection sessions. During each of the first three sessions, the subject performed single-finger tapping actions and captured 30 seconds of tapping data for each of the ten fingers. During the last session, the subject performed random multi-finger tapping actions and captured 150 seconds of tapping data for each hand. The subject was instructed to tap a trackpad in a natural unconstrained manner in all four sessions. In total, the subject contributed 1200-second of time-synchronized sEMG and force data. The force data was subsequently converted into $\{0, 1\}$

labels, i.e., "tap" and "no-tap". Two randomly selected sessions (out of the first three sessions) were used to construct the training set. The remaining two sessions were withheld and only used for evaluation. The CNN model was trained using Adam optimizer and cross-entropy loss for the task of per-frame finger-wise "tap" or "no-tap" classification for 20 epochs. The learning rate started at $1e-3$ and dropped to $1e-4$ at epoch 10. A weight decay factor of $1e-4$ was enforced to mitigate over-fitting. As a post-processing step, we applied a mean filter of window size 10 to the sequence of predicted tapping probabilities for temporal smoothing. The predicted tapping probability for each time frame was compared with a threshold of 0.3 to determine if it is "tap".

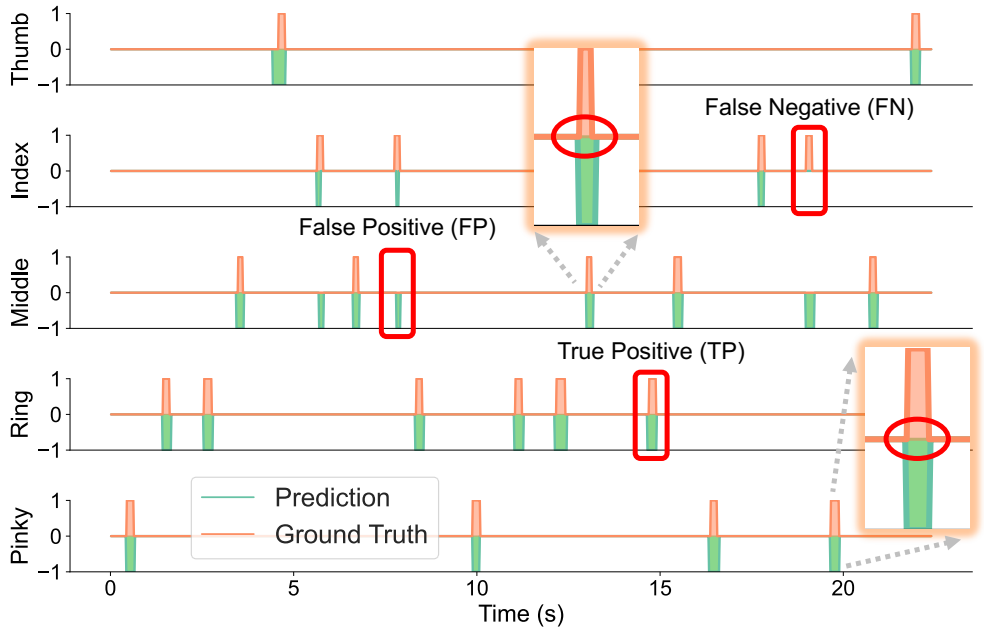
Evaluation metrics. We adapted two evaluation metrics from machine learning, precision (P) and recall (R), to accommodate our experimental setting. Here, P/R denotes the proportion of correctly detected tapping among all detected/ground-truth tapping. Note that both metrics are evaluated in a finger-wise manner.

$$\mathrm{P}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}; \quad \mathrm{R}_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}, \tag{7.6}$$

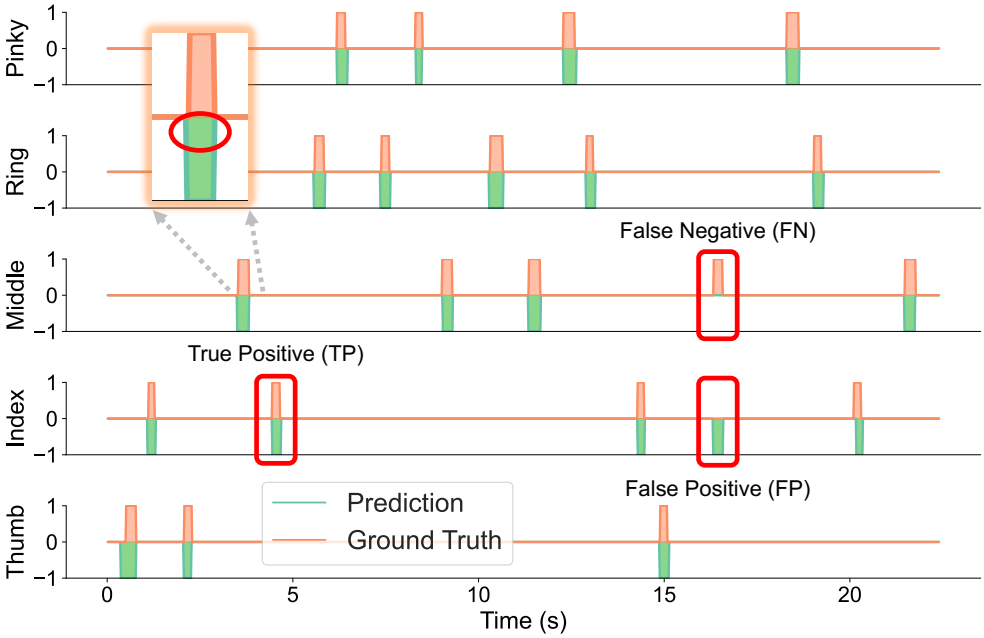
Here, TP (true positive), FP (false positive), and FN (false negative) denote the number of correctly detected tapping, falsely detected tapping, and undetected ground-truth tapping, respectively. Fingers are indexed by $i \in \{1, 2, 3, 4, 5\}$. In particular, a sequence of consecutive "tap" predictions of duration longer than 0.1 second is considered as a detected tapping. We consider correct detection if the temporal Intersection over Union (IoU) between its interval and the ground-truth is greater than 0.5, and falsely detected otherwise.

Table 7.3: *Ubiquitous control via finger tapping.* Precision and recall are reported for each finger and averaged across all ten fingers.

Metric	Thumb	Index	Middle	Ring	Pinky	Mean
Precision	93.3%	88.8%	84.9%	98.4%	94.4%	92.0%
Recall	94.2%	78.4%	94.7%	100.0%	96.2%	92.7%



(a) Right hand results



(b) Left hand results

Figure 7.13: *Ubiquitous control via finger tapping.* These panels visualize the ground-truth (orange) and model-predicted (green) results for two randomly sampled sequences from the held-out data. Bar width denotes the tapping duration.

Results. As shown in Table 7.3, our CNN model achieves mean precision of 92.0% and mean recall of 92.7% using only 10-minute data for training. The detection quality for the ring finger is the best, with 98.4% precision and 100.0% recall. The index finger showed lower recall 78.4%, while the middle finger showed lower precision 84.9%. Figure 7.13 visualizes the ground-truth (orange) and model-detected tappings (green) for two randomly-sampled sequences of multi-finger tapping from the last session, one for each hand. As we can see, most tapping actions are correctly detected by the model, with few false positives and false negatives. Accurate tapping duration and detection latency can also be observed.

Discussion. The above results demonstrate our method's applicability to effectively detect finger tapping as an ubiquitous interface. The definitions of precision and recall inherently establish a trade-off between the two metrics: the higher a model achieves in terms of recall, the lower it gets for precision, and vice-versa. Such trade-off can be translated into the context of this specific task as: the more sensitive the model is in detecting finger tapping (more positive predictions), the more actual finger tapping made by the user it will detect (higher recall), and the more mistakes it will make (lower precision). Conveniently, we can increase or decrease the detection threshold for "tap" to prioritize over precision or recall, depending on the demand in the actual application scenario. For instance, if we are considering an application where the accuracy of control signals outweighs the response rate, we might consider lowering the model's sensitivity to sacrifice a bit of recall for better precision.

7.7 ForceSense2: A Transformer Model for sEMG-to-Force Decoding

To establish a robust sEMG-to-force mapping for a more application-ready system, we combine a CNN-based feature extraction module with a transformer model [222]. CNNs are effective at

identifying patterns and extracting features from structured inputs such as the 2D spectrograms used in our pipeline. Transformer models, in contrast, are well suited for sequential data, where self-attention mechanisms capture contextual information and long-range temporal dependencies. By integrating these architectures, the model leverages their complementary strengths—CNNs for spatial feature extraction and transformers for temporal modeling—enabling more accurate interpretation of sEMG spectrogram sequences for finger-level force decoding.

Model architecture. The inputs to ForceSense2 are 4D tensors of shape (B, C, F, T), where B, $C = 8$, $F = 128$, and $T = 32$ denote the batch size, number of sEMG channels, number of retained frequency components, and spectrogram sequence length, respectively. We discard the highest-frequency component so that the number of frequency bins becomes a power of two. At the bottom of ForceSense2, we employ a CNN to extract deep features from the multi-channel sEMG spectrogram sequences. The CNN consists of four repeating Convolution-BatchNorm-ReLU blocks. The first three blocks are each followed by a 4×1 max-pooling layer, while the final block is followed by a 2×1 max-pooling layer. As a result, the frequency dimension is progressively downsampled from 128 to 1 and subsequently squeezed out. All convolutional layers use a 3×3 kernel.

Given the CNN outputs, a 3D tensor of shape (B, C, T), we employ a transformer layer to model temporal dynamics and relationships among the CNN-extracted features and to estimate the finger forces corresponding to the input sEMG signals. The transformer comprises ten encoder blocks, each with eight attention heads. The output is a 3D tensor of shape (B, J, T), where B, $J = 5$, and $T = 32$ denote the batch size, number of force components (i.e., fingers), and output sequence length, respectively. Notably, ForceSense2 processes sEMG spectrogram sequences that are 32 frames long (equivalent to 624 ms), allowing the model to exploit muscle-activity information from the recent past. Only the predicted force values in the most recent frame of the 32-frame output are retained and used throughout our experiments. At inference time, a sliding window is

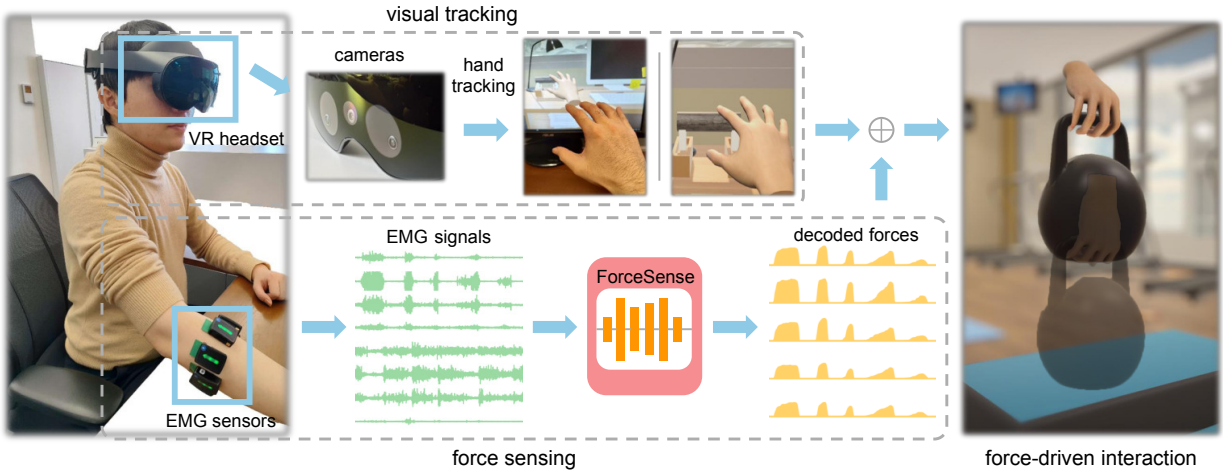


Figure 7.14: *Workflow of ForceSense2.* ForceSense2 provides fast, robust, and accurate sEMG-to-force decoding that, together with visual tracking, enables XR users to perform dexterous interaction tasks in virtual environments.

used to extract the most recent segment of the sEMG data stream and feed it into ForceSense2 for real-time prediction. Figure 7.14 illustrates the overall workflow of ForceSense2.

Experimental setup. We compare ForceSense2 with ForceSense1 on the same dataset (Section 7.2). We implemented ForceSense2 using the PyTorch deep learning framework [194] and trained it with the Adam optimizer [125] for 100 epochs. A batch size of 256 was used. The learning rate was initialized at 1×10^{-4} and decayed by a factor of 0.7 every 10 epochs. All hyperparameters were selected via cross-validation.

Results. As shown in Table 7.4, ForceSense2 substantially outperforms ForceSense1 across all three evaluation metrics, achieving an average accuracy, NRMSE, and R^2 of 91.35%, 3.28%, and

Table 7.4: *sEMG-to-force decoding performance comparison.* The ↑/↓ symbols indicate that higher/lower values correspond to better performance.

Method	Accuracy↑	NRMSE↓	R^2 ↑
ForceSense1	88.83±6.13%	3.29±1.76%	85.82±14.96%
ForceSense2	**91.35±1.83%**	**3.28±0.58%**	**89.59±2.60%**

89.59%, respectively. Notably, this predictive performance is obtained with a relatively modest model size of only 25.2M trainable parameters. Given an 8-channel sEMG data stream sampled at 2000 Hz, ForceSense2 requires 793.98M MACs (multiply–accumulate operations) to perform a single inference with batch size 1. On a desktop PC equipped with an Nvidia RTX 4090 GPU, this computation takes 5.50 ms on average, corresponding to an effective throughput of approximately 180 FPS.

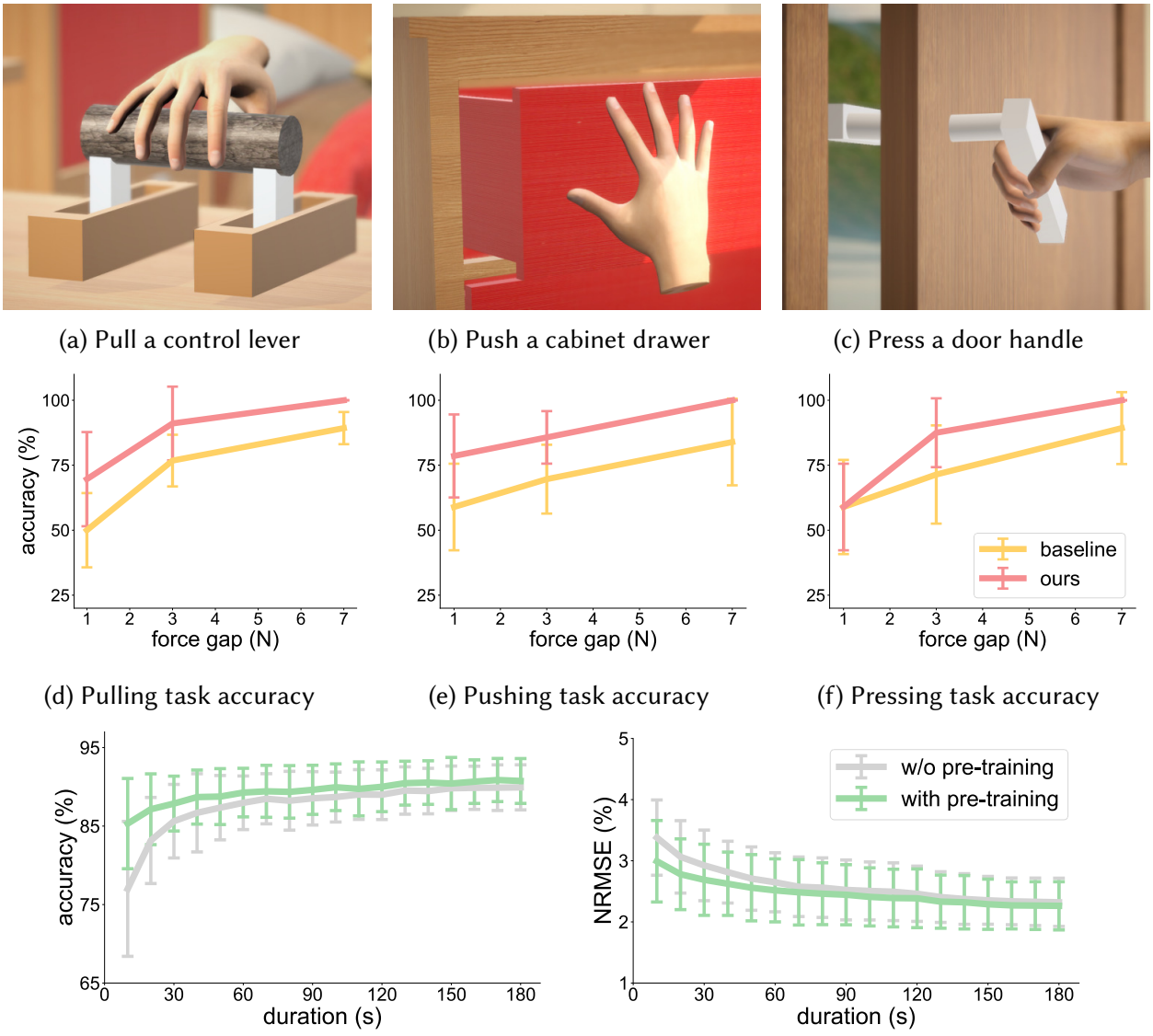
Discussion. Prior literature suggests that low system latency and high refresh rates are essential for building effective XR applications. In particular, users' motor performance, sense of agency, and simultaneity perception can deteriorate when system response delays exceed 50 ms [223, 224]. Empirical studies examining the effect of refresh rate on XR users' task performance and simulator sickness indicate that 120 FPS is an important threshold, beyond which no adverse effects from insufficient refresh rate are typically observed [107, 225]. Notably, ForceSense2 operates at approximately 180 FPS with a system delay of 7 ms, exceeding the requirements suggested by prior work by a substantial margin. In addition, the average force decoding error of ForceSense2, measured by NRMSE, is 3.28%, corresponding to an absolute error of only 0.98 N. These results demonstrate the practicality of ForceSense2 for real-time XR applications.

7.8 User Study: Force-Driven Interaction with Virtual Objects

Leveraging ForceSense2's force decoding capability, we designed a user study comprising three hand-object interaction scenarios commonly encountered in XR. Figure 7.15 summarizes the study configuration, while Figures 7.15a to 7.15c illustrate the three interaction tasks individually. The goal of the study is to examine the extent to which ForceSense2 improves users' perception of the *physical workload* required by different virtual interaction tasks.

Participants. We recruited eight participants (ages 22-26; 3 female) for the study. All had normal or corrected-to-normal vision, no history of auditory impairment, and no finger, hand, or arm impairments. Among the participants, four had no prior experience with XR, two had used XR equipment five times or fewer, and two were familiar with and had regular access to XR devices. None were aware of the study hypothesis, the research goals, or the number of task difficulty levels. The study protocol was approved by the Institutional Review Board (IRB) at the host institution, and all participants provided written informed consent prior to the study.

Calibration. We conducted a pre-study calibration to build a user-specific ForceSense2 model for each participant. Following the procedure described in Section 7.2, we recorded two minutes of forearm sEMG and finger force data for each participant while they performed the three elemental interactions shown in Figures 7.15a to 7.15c. A rigid plastic cuboid box ($35,\text{cm} \times 25,\text{cm} \times 15,\text{cm}$), fixed on a desk, served as the proxy object for force exertion. Participants were seated in front of the proxy box with their elbows in a natural resting posture, aligned in height with the top surface of the box. They were instructed to pull the back face, push the front face, and press the top face of the box by naturally extending their arms to realize the three elemental interactions. The Sensel Morph force trackpad was attached to the corresponding face of the box to measure finger-level forces. The resulting six minutes of synchronized sEMG-force data were used to fine-tune the pre-trained ForceSense2 model described in Section 7.3 to obtain a personalized model for each participant. As shown in Figures 7.15g and 7.15h, pre-training user-specific ForceSense2 models on the dataset from Section 7.2 substantially reduces the amount of data required for calibration. Our decision to collect only two minutes of data per interaction type was further motivated by the observation that model performance gradually plateaus beyond this duration, as illustrated in Figures 7.15g and 7.15h.



(a) Pull a control lever (b) Push a cabinet drawer (c) Press a door handle

(d) Pulling task accuracy (e) Pushing task accuracy (f) Pressing task accuracy

(g) Calibrated performance: accuracy (h) Calibrated performance: NRMSE

Figure 7.15: *Force-driven interaction with virtual objects via ForceSense2.* The top row illustrates the three interaction scenarios used in the user study: pulling a control lever, pushing a cabinet drawer, and pressing a door handle. The middle row reports the corresponding discrimination performance, showing that ForceSense2 improves users' ability to perceive the physical workload required by each interaction compared with the baseline method. The bottom row shows how classification accuracy and NRMSE change as additional subject-specific data are used to fine-tune the model.

Conditions. In addition to our force-aware interface ForceSense2, we included a tracking-based interface commonly used in commercial XR platforms as a baseline condition for performance comparison. For ForceSense2, participants interacted with the same proxy box used during the calibration process to exert forces. The decoded finger forces were applied to virtual objects to realize the force-driven interactions shown in Figures 7.15a to 7.15c, namely pulling a control lever, pushing a cabinet drawer, and pressing a door handle. For the baseline condition, we adopted a standard velocity-based approach from prior literature [226], mapping users' hand movement speed (measured as the speed of the palm joint tracked by the Oculus VR SDK) to varying force levels for the same set of interactions. Specifically, the speed range [0, 1] m/s was linearly mapped to the force range [0, 30] N. Participants were instructed to produce a desired force intensity by adjusting their hand movement speed accordingly. Note that the baseline method cannot exploit the proxy box to enhance the interaction experience, as its interaction mechanism requires unobstructed space to generate movement-induced forces.

Tasks. We designed a two-interval forced-choice (2IFC) psychophysical task [227] to measure users' ability to distinguish between virtual interactions requiring different levels of physical workload. Taking the pulling a control lever interaction as an example, during each trial participants interacted sequentially with two control levers with different resistance forces and indicated which one required greater physical effort. The difference in resistance force between the two levers controlled the task difficulty. The pushing a cabinet drawer and turning a door handle tasks followed the same procedure. Both ForceSense2 and the baseline interface were used to complete the same set of 2IFC tasks. Performance was evaluated by the percentage of trials in which participants correctly identified the interaction requiring greater physical effort.

Setup and procedure. During the study, participants wore a Meta Quest Pro headset for visual and auditory feedback. Eight Delsys Trigno sensors were mounted on the right forearm to capture

and stream sEMG signals in real-time. The XR applications implementing the three interaction scenarios (Figures 7.15a to 7.15c) were developed using the Unity game engine [228]. The study was divided into three sessions (one per interaction), with enforced breaks between sessions to mitigate muscle fatigue. A brief warm-up of 12 trials preceded each session to familiarize participants with the task. Each session consisted of (#condition = 2) $\times$ (#difficulty = 3) $\times$ (#repetition = 7) = 42 trials. In each 2IFC trial, each interaction had a time limit of 10 s, with the option to terminate earlier to proceed. Task difficulty was controlled using three force-level pairs centered at 25 N with gaps of 1 N, 3 N, and 7 N (i.e., 24.5 N vs. 25.5 N, 23.5 N vs. 26.5 N, and 21.5 N vs. 28.5 N). To avoid order-induced bias, the condition order was counterbalanced across participants. The order of appearance within each 2IFC pair and the task difficulty levels were randomized. The entire study—including hardware setup, instructions, warm-up trials, and breaks—took approximately 2.5 hours per participant.

Results and discussion. The aggregated 2IFC results across all eight participants are summarized in Figures 7.15d to 7.15f. Across all three interaction scenarios, ForceSense2 consistently outperforms the baseline method. The mean discrimination accuracy with ForceSense2 exceeds 85% when the force difference is greater than 3 N and reaches perfect discrimination for force differences of 7 N. Note that an absolute force difference of 3 N (23.5 N vs. 26.5 N) corresponds to a relative difference of 12.7% in our study. These results demonstrate the effectiveness of ForceSense2 in enhancing XR users' perception of physical workload and highlight the importance of tactile control in immersive XR applications.

7.9 User Study: Multimodal Task Learning

In addition to simple force-driven interaction with virtual objects, we implemented two proof-of-concept XR applications using ForceSense2: piano playing and calligraphy. Figures 7.16a and 7.16b

illustrate the two conditions. These tasks allow us to validate the robustness of the interface in scenarios requiring high-finesse manipulation skills. Through a series of imitation tasks, we evaluate users' task performance and collect subjective feedback, including perceived usability, physical workload, and sense of presence.

Participants. We recruited 12 participants (ages 20–28; 5 female) for the study. All had normal or corrected-to-normal vision, no history of auditory impairment, and no finger, hand, or arm impairments. Among the participants, 3 had no prior experience with XR, 4 had used XR devices five times or fewer, and 5 were familiar with and had regular access to XR devices. None were aware of the research hypothesis. The study protocol was approved by the Institutional Review Board (IRB) at the host institution, and all participants provided written informed consent prior to the study.

Calibration. Similar to the user study described in Section 7.8, we collected four minutes of sEMG-force data from each participant as a pre-study calibration step to adapt the pre-trained ForceSense2 model from Section 7.7 into a user-specific one. During calibration, participants were asked to tap or press the Sensel Morph force trackpad (fixed on a table) using only their index finger, varying the applied force levels, hand and finger postures, and contact durations to ensure diverse data coverage. The resulting personalized models were then used to decode users' finger-level forces in real-time for both case studies described below.

Experimental setup. During the study, participants wore a Meta Quest Pro headset for visual and auditory feedback. Eight Delsys Trigno sensors were mounted on the right forearm to capture and stream sEMG signals in real-time, following the same setup described in Section 7.8. The scenes and applications for virtual piano playing and calligraphy practice (Figure 7.16) were implemented in the Unity game engine [228].



(a) Piano keystroke learning (auditory-tactile) (b) Calligraphy stroke learning (visual-tactile)

Figure 7.16: *Demonstration of multimodal task learning with ForceSense2.* The left panel shows the piano keystroke imitation task, while the right panel shows the calligraphy practice task. These studies use ForceSense2 to couple visual tracking or auditory feedback with decoded finger forces. The volume plot shown with the piano task is included for illustration purposes only and was not available to participants during the study.

7.9.1 Auditory-Tactile Task: Piano Keystroke Imitation

Conditions. For performance comparison, we adopted a hand-tracking interaction method similar to that described in Section 7.8 as the baseline. For ForceSense2, we used a large table as the physical proxy and mapped a specific area of the table to the virtual piano keyboard. Participants were instructed to tap on the table as if performing piano keystrokes. The location of the index finger, detected by the headset cameras, was used to determine the piano key to be struck, while the model-predicted forces were applied to generate the corresponding keystrokes. For the baseline method, the movement speed of the user's index fingertip was mapped to varying force levels, and the converted forces were used to drive the intended piano key. The speed range $[0, 70]$ °/s was linearly mapped to the force range $[0, 17]$ N.

Tasks. We designed an auditory imitation task to evaluate users' ability to reproduce simple piano recordings (considering only single keystrokes) by applying appropriate force levels to generate the keystrokes. At the beginning of each trial, a reference piano recording was played.

Participants were then asked to imitate the reference recording to match its volume profile. Each trial allowed up to five imitation attempts. The absolute volume error, measured in decibels, was used as the evaluation metric. Among all imitation attempts within a trial, only the final attempt was recorded and compared with the reference recording to compute the metric. Both ForceSense2 and the baseline method were used to complete the same set of imitation tasks. Our experimental setup is illustrated in Figure 7.16a. Note that the volume plot shown in the figure is for illustration purposes only and was not visible to the participants.

Procedure. We used three single-keystroke piano recordings with different peak volume levels as reference stimuli. The study consisted of $(\#\text{condition} = 2) \times (\#\text{reference} = 3) \times (\#\text{repetition} = 3) = 18$ trials. Before the main study, participants completed a short warm-up session (approximately 4 minutes) to familiarize themselves with the task. To minimize order-induced performance bias, the condition order was counterbalanced across participants, and the reference recording order was randomized. After completing the trials, participants filled out a post-study questionnaire evaluating the two conditions along four dimensions: subjective workload, perceived usability, technology acceptability, and sense of presence. The questionnaire contained 17 items (Table 7.6) drawn from five standard XR/HCI instruments: NASA-TLX [229], UMUX [230], UTAUT [231], PQ [232], and IPQ [233]. The entire piano study—including hardware setup, instructions, warm-up trials, and the post-study survey—took approximately 40 minutes per participant.

7.9.2 Visual-Tactile Task: Calligraphy Stoke Imitation

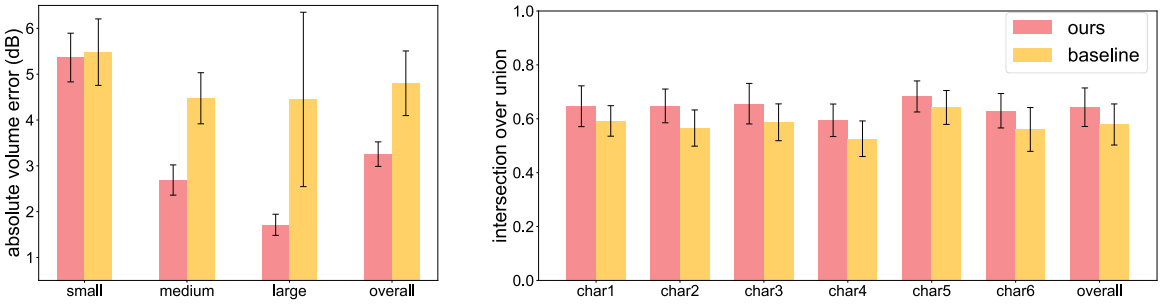
Conditions. In addition to our force-aware interface ForceSense2, we included a hand-tracking interaction method as the baseline for performance comparison. For ForceSense2, we again used a large table as the physical proxy and mapped a specific area of the table to the virtual canvas. Participants were instructed to slide their fingers on the table to generate handwriting strokes. The current position of the index finger determined the stroke location, while the decoded finger

forces were used to control the stroke size. For the baseline condition, we defined a horizontal plane above the virtual canvas and mapped the penetration depth of the participant's index finger relative to that plane to varying force levels. The depth range [0, 9] mm was linearly mapped to the force range [0, 13] N.

Tasks. We designed a visual imitation task to evaluate users' performance in reproducing several calligraphy works. In particular, participants were instructed to control the stroke size by adjusting the applied force intensity. During each trial, a reference calligraphy work rendered in light gray was displayed on the canvas, and participants were asked to reproduce it using continuous handwriting strokes. A bright cursor indicating the projected location of the participant's index fingertip was rendered to provide visual guidance. Each trial had a fixed time limit of 30 s. Performance was evaluated using the intersection-over-union (IoU) metric computed between the reference and the reproduced work. Both ForceSense2 and the baseline interaction method were used to reproduce the same set of calligraphy works. Figure 7.16b illustrates the experimental

Table 7.5: *One-way ANOVA on the interaction method.* In both the piano and calligraphy studies, the interaction method shows a significant main effect on users' multimodal task learning performance.

calligraphy work	$F_{1,11}$	p
1	24.6	$< .0005$
2	25.5	$< .0005$
3	27.3	$< .0005$
4	29.9	$< .0005$
5	13.7	$< .005$
6	16.8	$< .005$
overall	46.4	$< .0001$
peak volume	$F_{1,11}$	p
small	0.0152	ns
medium	10.0	$< .01$
large	2.06	$> .05$
overall	5.43	$< .05$



(a) Piano (auditory-tactile)

(b) Calligraphy (visual-tactile)

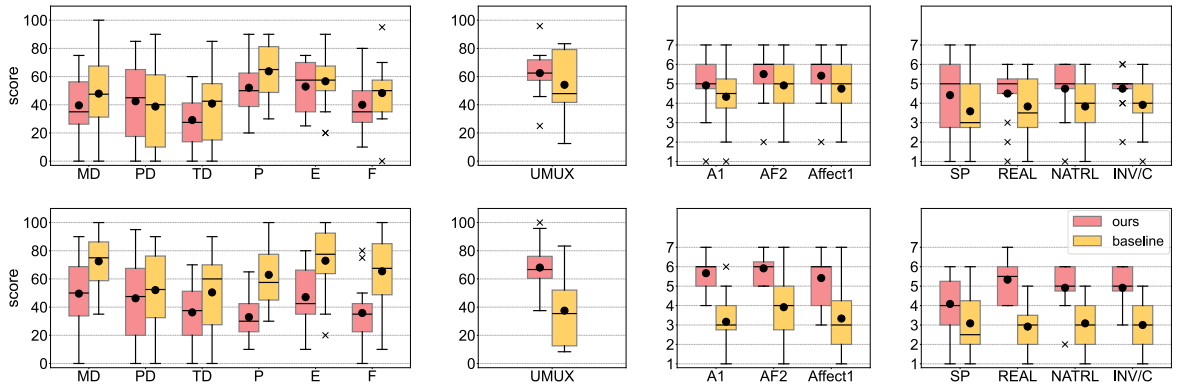
Figure 7.17: *Quantitative evaluation of multimodal task learning with ForceSense2.* The left panel shows users' auditory-tactile task performance during piano keystroke imitation. Lower volume errors indicate more accurate reproduction of the reference keystrokes. The right panel shows users' visual-tactile task performance when reproducing calligraphy works. Higher IoU scores indicate smaller discrepancies between the reference and reproduced works and therefore better imitation performance.

setup.

Procedure. We prepared six calligraphy works containing both Chinese and English characters as reference stimuli. In total, the study consisted of (#condition = 2) × (#reference = 6) × (#repetition = 3) = 36 trials. Before the main study, participants completed a short warm-up session (approximately 4 minutes) to familiarize themselves with the task. To reduce order-induced bias, the condition order was counterbalanced across participants, and the reference order was randomized. The same post-study survey described in Section 7.9.1 was used. The entire study—including hardware setup, instructions, warm-up trials, and the post-study survey—took approximately 70 minutes per participant.

7.9.3 Results

Figure 7.17 summarizes the aggregated imitation errors across all 12 participants for both studies, with Figures 7.17a and 7.17b presenting the auditory-tactile and visual-tactile tasks separately. In the piano-playing study, the hand-tracking baseline performs comparably to ForceSense2 for recordings with small peak volumes but falls behind for recordings with medium to large peak



(a) NASA-TLX (b) UMUX (c) UTAUT (d) PQ/IPQ

Figure 7.18: *Subjective evaluation of multimodal task learning with ForceSense2 through a post-study survey.* The post-study survey contains 17 questions (Table 7.6) drawn from five standard XR/HCI questionnaires: NASA-TLX, UMUX, UTAUT, PQ, and IPQ. The Likert-scale responses are aggregated into 13 subgroups to assess both conditions along four dimensions: subjective workload, perceived usability, technology acceptability, and sense of presence. *Higher* scores indicate better performance for UMUX, UTAUT, PQ, and IPQ, whereas *lower* scores indicate lower workload for NASA-TLX. Box plots summarize the response distributions. The boxes represent the interquartile range (Q1–Q3), dots indicate the mean, lines indicate the median, and whiskers extend to the farthest points within 1.5× IQR. Outliers beyond this range are plotted as black crosses. The top row shows the results for the piano study, while the bottom row shows the results for the calligraphy study.

volumes. Notably, ForceSense2 enables users to reproduce piano keystrokes with a mean volume error of 3.26 dB across the entire volume range, which is 32.1% lower than that of the baseline. For the calligraphy imitation study, ForceSense2 achieves consistently higher IoU scores across all six calligraphy works and outperforms the baseline by 11.1% on average. Figure 7.19 illustrates several randomly sampled imitation results. As reported in Table 7.5, a one-way repeated-measures ANOVA reveals a significant main effect of the interaction method on participants' imitation performance in both studies. The Likert-scale ratings aggregated from the post-study survey (Figure 7.18) further indicate that, compared with the hand-tracking interaction method, our force-aware interface ForceSense2 imposes lower physical and mental workload, provides better usability, and delivers a stronger sense of presence for XR users.

Table 7.6: *The 17 questions used in the post-study survey.* Our survey is designed to assess and compare *ForceSense2* with the baseline method along four dimensions: subjective workload, perceived usability, technology acceptability, and sense of presence.

UX Component	Question
Usability (from UMUX [230])	1) This interface's capabilities meet my requirements. 2) Using this interface is a frustrating experience. 3) This interface is easy to use. 4) I have to spend too much time correcting things with this interface.
Presence (from PQ [232] and IPQ [233])	5) With this interface, I had a sense of acting in the virtual space, rather than operating something from outside. 6) How much did your experience with this interface in the virtual environment seem consistent with your real-world experience? 7) How natural did your interactions with the objects seem? 8) How proficient in interacting with the virtual environment did you feel at the end of the experience?
Technology Adoption (from UTAUT [231])	9) Using the interface is a bad/good idea. 10) Working with the interface is fun. 11) I like working with the interface.
Workload (from NASA-TLX [229])	12) How mentally demanding was using the interface? 13) How physically demanding was using the interface? 14) How hurried or rushed was the pace of using the interface? 15) How successful were you in accomplishing what you were asked to do with this interface? 16) How hard did you have to work to accomplish your level of performance with this interface? 17) How insecure, discouraged, irritated, stressed, and annoyed were you?

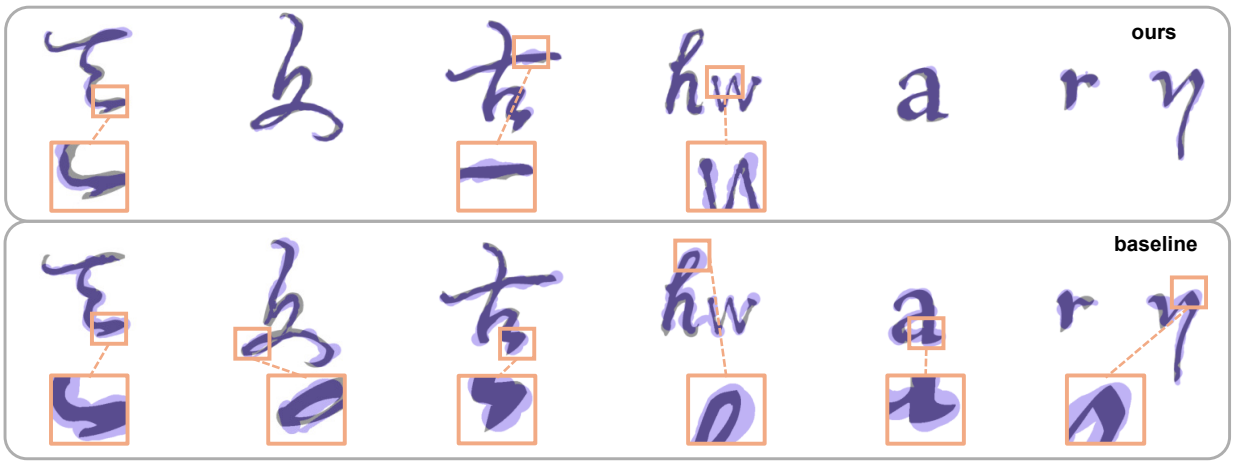


Figure 7.19: *Qualitative results from the calligraphy study.* The reference calligraphy works (gray) are overlaid with users' imitation results (purple). The top and bottom rows show several samples produced using ForceSense2 and the baseline method, respectively.

7.9.4 Discussion

The results above demonstrate that ForceSense2 can effectively enhance XR users' task performance and learning efficiency in virtual environments. While both the hand-tracking baseline and ForceSense2 employ a force-driven paradigm to support hand-object interactions, ForceSense2 provides a more intuitive interface for XR users to interact with virtual objects. Importantly, ForceSense2 allows users to naturally exert physical forces without constraining hand or finger postures. The appropriate force levels are then decoded and applied to the virtual objects with which users intend to interact. Beyond the objective evaluations, the post-study questionnaire assessing users' subjective preferences further suggests that ForceSense2 outperforms the hand-tracking baseline in terms of workload, technology acceptability, perceived usability, and sense of presence.

We attribute these benefits to the fact that ForceSense2 facilitates a stronger synergy among visual, auditory, and haptic modalities. In the physical world, humans naturally explore their environment through coordinated visual, auditory, and tactile sensing. Faithfully recreating this multimodal interaction loop in virtual environments is therefore essential for designing intuitive

and effective XR interfaces.

7.10 Discussion

This chapter shows that finger-level force can serve as a practical interaction signal for immersive systems when it is decoded directly from forearm muscle activity. Across two linked stages, we established synchronized sEMG-force datasets, developed two predictive models with complementary strengths, and demonstrated that decoded force can support both elemental physical interaction and more complex dexterous tasks. The results indicate that force-aware interfaces can narrow the gap between what users physically intend and what immersive systems are able to sense and simulate, especially in tasks where hand posture alone is insufficient.

The first major outcome is methodological. ForceSense1 established the feasibility of low-latency finger-force decoding from multichannel sEMG with an autoencoder model and a joint classification-regression objective. That system demonstrated robust prediction accuracy, reasonable transfer to unseen users with short calibration, and clear benefits in psychophysical studies of material perception and tapping-based control. ForceSense2 then extended the same research direction with a transformer model that better captures temporal dependencies and supports more demanding interactive tasks. Together, the two models illustrate a progression from proof-of-concept force decoding toward a more mature interface component for immersive interaction pipelines.

The second outcome is experiential. Force-aware interaction changes not only what the system measures, but what users can perceive and accomplish. In the first study, decoded force improved users' discrimination of virtual material properties, indicating that physically meaningful input helps close the loop between action and perceptual judgment. In the later studies, the same force channel improved the fidelity of pulling, pushing, pressing, piano imitation, and calligraphy practice. These gains suggest that tactile and motor realism in XR does not come only from

output-side haptics; it also depends on whether the system can faithfully sense the user's own exertion.

Several limitations remain. The interaction repertoire in the collected datasets is still limited relative to the full diversity of everyday dexterous behavior. Although this chapter studies pressing, pinching, pulling, pushing, tapping, and task-specific manipulation, additional actions such as grasping, squeezing, or interaction with more complex 3D objects would broaden coverage and likely improve deployment robustness. Moreover, both stages still rely on short per-user calibration to accommodate individual muscular variation. Reducing or eliminating that requirement remains important for real-world adoption. Online adaptation, unsupervised personalization, or larger cross-user training corpora are promising directions.

Another limitation is that the interface is centered on force decoding rather than full hand-object state estimation. Real interactions depend jointly on force, pose, contact configuration, and sometimes resistance from external objects. Future systems could integrate decoded force with hand tracking, arm pose, or tactile sensing to better separate active force generation from environmental resistance and to support more varied manipulation tasks. Extending evaluation to broader user populations, including users with motor impairments, would also strengthen the case for accessibility-oriented applications.

Despite these limitations, this chapter demonstrates that sEMG-based force decoding is a viable path toward more natural and embodied immersive interation. It adds a motor and haptic dimension to the dissertation's broader argument that interactive graphics systems should be guided by human capabilities rather than computation alone. Here, the relevant capability is not only what users can see or hear, but also what they can physically do with their hands.

Chapter 8

Toward Ergonomically Friendly XR Experiences

Visual fidelity and interaction richness alone are not sufficient for successful immersive systems. If the physical effort required to look around, search for targets, or sustain attention becomes uncomfortable, even a technically impressive experience may be difficult to use for long periods. This issue is especially relevant for head-mounted displays (HMDs). Their wide field of view and head tracking enable natural viewing, but they also place additional mass on the head and alter the mechanical demands of neck motion. As immersive devices move from short demonstrations toward longer sessions of work, training, and entertainment, ergonomic efficiency becomes a fundamental systems concern rather than a secondary usability detail.

This chapter studies neck muscle workload as a measurable and predictable component of immersive interaction. The central quantity is muscle contraction level (MCL), which reflects the muscular effort required to hold and move the head while wearing an HMD. Unlike optical motion, muscle activity is not directly visible to a camera, and unlike subjective discomfort ratings, it can be measured continuously during interaction through electromyography. That makes MCL a useful bridge between low-level biomechanics and high-level interface design: if we can model how

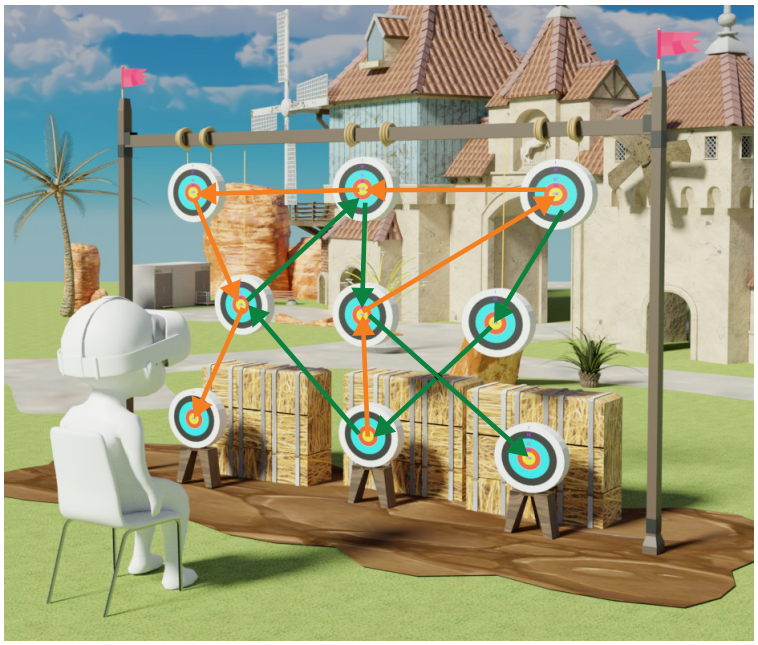
target placement and head motion affect muscle contraction, we can begin to forecast ergonomic cost before an application is deployed.

Prior work has already established that HMDs can increase neck and shoulder discomfort, especially during prolonged or wide-range viewing [65, 234–237]. At the same time, surface electromyography (sEMG) has been widely used to analyze muscle activity in biomechanics, ergonomics, and human-computer interaction [51, 238, 239]. What remains less developed is a predictive model that connects those measurements to immersive content design. Existing studies often quantify discomfort after the fact, but many extended reality (XR) design problems require pre-hoc reasoning: where should targets be placed, how demanding is a scan path likely to be, and can an interface be optimized before users experience it?

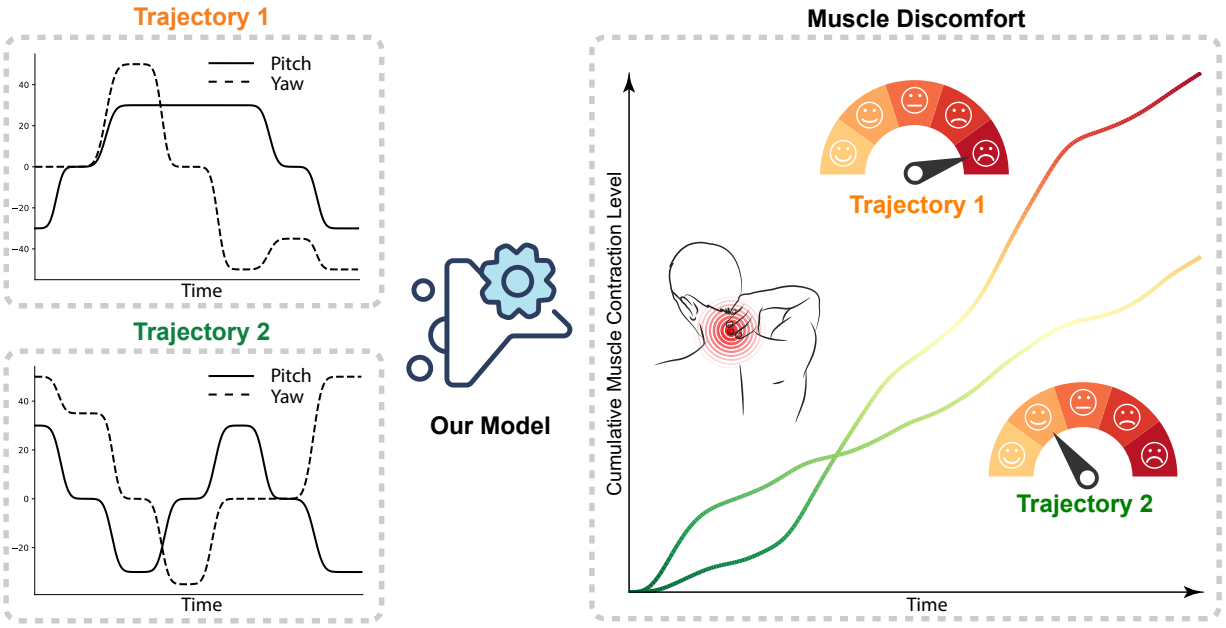
We address those questions by combining sEMG measurements of neck muscle activity with a bio-physically inspired model of head movement. Figure 8.1 illustrates a motivating example. Two head-motion trajectories may appear visually comparable, yet they can differ substantially in the muscular effort they require. By modeling that effort, we can estimate workload after a movement and predict it before the movement occurs. This distinction is important. Post-hoc estimation provides a quantitative ergonomic analysis tool, while pre-hoc prediction enables design guidance for layout optimization, task authoring, and potentially ergonomics-aware adaptive interfaces.

This chapter begins with a physiological study in XR that characterizes how neck muscle contraction varies across static viewing poses and dynamic head movements. This pilot analysis reveals several important trends: head pose strongly affects baseline workload, pitch and yaw contribute differently, and dynamic motion further increases muscular demand in ways that depend on movement direction and acceleration. These observations motivate a computational model that maps head pose and motion to predicted MCL.

We then introduce a two-part modeling framework. The first part, MCLNet, estimates MCL from complete head trajectories using a bio-physically inspired formulation that relates passive torque, active torque, and angular acceleration. The second part, TrajectoryNet, predicts represen-



(a) Two head motion trajectories during immersive viewing



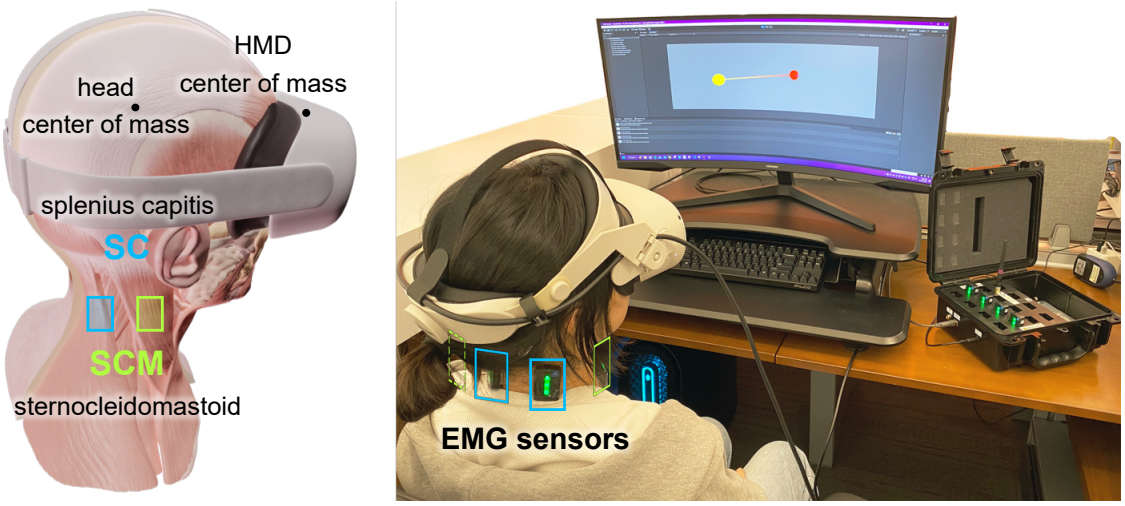
(b) Model-predicted neck muscle discomfort level over time

Figure 8.1: *Predicting neck muscle discomfort level during immersive viewing.* The top panel shows an XR user choosing between two candidate head motion trajectories that require seemingly similar workload for a visual task. The bottom panel shows our computational model predicting the potential neck muscle contraction level and the corresponding neck muscle discomfort trend before the movements occur. 3D asset credits: Mixall, Bizulka, and RootMotion (Unity); shockwavegamez01 and joseVG (Sketchfab).

tative motion trajectories from starting and ending head poses so that cumulative workload can be forecast even before a user moves. Together, these models support both retrospective analysis and prospective ergonomic prediction. Their performance is evaluated on held-out subjects and unseen conditions, and the chapter concludes with a user study showing that model-guided scan paths correspond to participants' subjective discomfort judgments.

This work connects to the dissertation's broader theme by extending human-centered graphics beyond perception and direct interaction toward sustained physical comfort. Earlier chapters examined how representation, rendering, generation, and force-aware interfaces can align systems with what users perceive and do. Here we consider another constraint that becomes increasingly important as immersion deepens: what it costs the body to participate. If multimodal immersive experiences are to be effective over time, they must account not only for perceptual saliency and interaction fidelity, but also for ergonomic burden.

This chapter is organized as follows. We first characterize neck muscle contraction during head movements in XR. We then develop a bio-physically inspired prediction model, incorporating both the main method and the sEMG processing details required to derive MCL from raw signals. Next, we evaluate estimation and prediction accuracy for complete trajectories and target-pose-based forecasts. Finally, we present a user study demonstrating that our model can guide the design of target layouts that reduce perceived neck discomfort. Together, these sections move ergonomic reasoning from descriptive measurement toward predictive support for immersive system design.



(a) Neck muscle anatomy (b) Pilot study experimental setup

Figure 8.2: *Human neck muscle anatomy and pilot study experimental setup.* **Left:** the major muscles controlling head movements, with highlighted sEMG sensor attachment regions. **Right:** our experimental setup with sEMG sensors annotated and stimuli displayed. 3D asset credits: joseVG and danielmclogan (Sketchfab).

8.1 Quantifying Neck Muscle Contraction during Immersive Viewing

8.1.1 Neck Muscles Controlling Head Movements

We aim to model and predict neck muscle contraction level (MCL). Human neck is a highly flexible skeletal structure that allows the head to change its pitch, yaw, and roll angles. As shown in Figure 8.2a, the major neck muscles include sternocleidomastoid (SCM) and splenius capitis (SC) [240]. In particular, SCM/SC laterally rotate the head to the opposite/same side when acting unilaterally and flex/extend the head when acting bilaterally.

8.1.2 Pilot Study

Participants and setup. We recruited 8 participants (ages 23 – 31, 3 female). 4 of them have prior experience with XR headsets before the study. All participants reported normal neck muscle

conditions. For each participant, we attached 4 Delsys Trigno wireless sEMG sensors on the left/right SCM and SC muscles (see Figures 8.2a and 8.2b). We originally tested with 6 sEMG sensors by also including the upper trapezius (UT) muscles. However, sEMG sensors on the UT exhibited significantly weaker signals compared to the others and were thus excluded from the experiments. The sEMG sensors detect the electric potential (in millivolts, mV) generated by users' muscle fibers at 2000 Hz and stream the data to a PC with $<$ 1ms latency. During the study, every participant wore an Oculus Quest 2 HMD, remained seated, and performed a target reaching task with visual stimuli. Their head poses were tracked by the HMD and streamed to the same PC for time synchronization with sEMG. The HMD provides 1872×1856 resolution per eye at 90 FPS, and $98^\circ/104^\circ$ vertical/horizontal field of view (FoV).

Stimuli and tasks. As illustrated in Figure 8.2b, the stimuli were sequentially displayed pairs of spheres (3° in the FoV), one colored yellow and the other red. The yellow sphere indicates an "anchor" head pose $\mathbf{r} \triangleq (p, y)$, where p/y represent pitch and yaw angles. The red sphere indicates a "target" head pose that is $\mathbf{\Delta r} \triangleq (\Delta p, \Delta y)$ away from $\mathbf{r}$. While keeping their torso stationary, participants were instructed to rotate their heads from the yellow anchor to the red target. Once the user successfully fixated on the target for 2 seconds consecutively (simulating stationary viewing), both spheres were shifted to continue with the next trial. A line connecting the two spheres was rendered to guide the user's head rotation to the next target, designed to eliminate potential errors for target searching.

Conditions. Across participants, the stimuli pairs appeared randomly with a pre-sampled anchor set $\mathcal{R} \triangleq \{(\mathbf{r}_i, \mathbf{\Delta r}_i), i = 1, \ldots N_{\mathcal{R}}\}$. Given the comfortable range of human head rotations [241], we sampled stimulus directions within a $60^\circ \times 100^\circ$ visual field. We tested and chose not to cover additional vertical range to ensure participants' neck comfort and avoid sprains. We started with a preliminary test to determine the effective range of $\mathbf{r}$. Our test with 3 participants

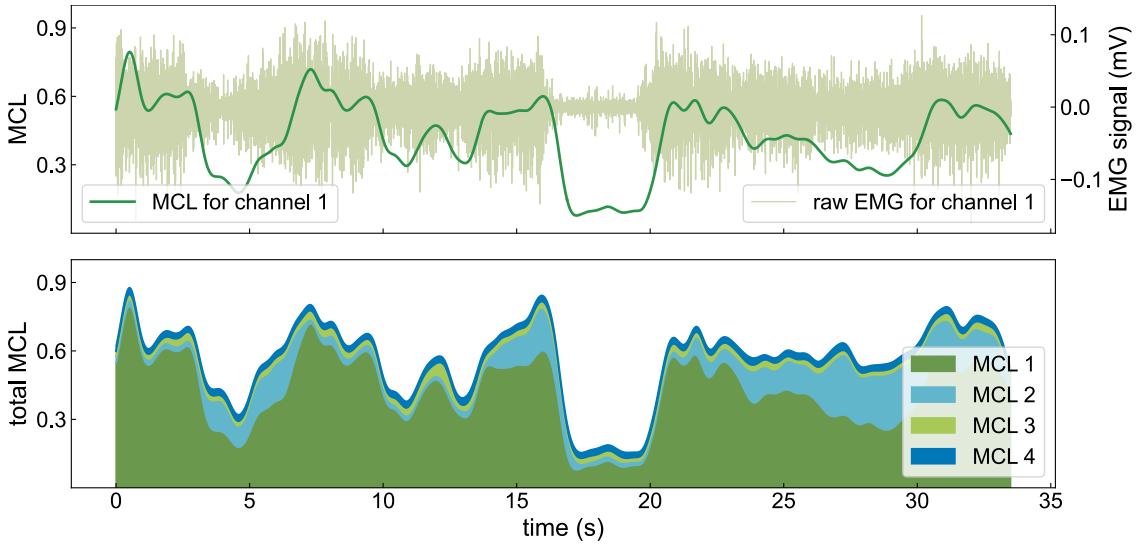


Figure 8.3: *sEMG data processing to obtain normalized MCL.* The top row shows an example raw sEMG sequence (light green curve and right Y-axis) and its corresponding normalized MCL (dark green curve and left Y-axis). The bottom row shows the total MCL integrated across all 4 channels.

showed non-trivial sEMG change for $|y| > 15°$ and $|p| > 5°$. Therefore, using a $10°$ step size, we sampled 63 anchor head poses for $p \in \{\pm 30°, \pm 20°, \pm 10°, 0°\}$ and $y \in \{\pm 50°, \pm 40°, \pm 30°, \pm 20°, 0°\}$. Here, $-/+$ indicates left/right or down/up from the head's forward-facing direction. For each of the 63 anchor poses $\mathbf{r}$, 8 surrounding targets giving varying movement patterns $\mathbf{\Delta r}$ were studied. They were selected with $\Delta p \in \{\pm 35°, 0\}, \Delta y \in \{\pm 25°, 0\}$. We discarded the conditions with target stimuli outside of the $60° \times 100°$ range. That is, we ensured $\mathbf{r} + \mathbf{\Delta r} \in [-30°, +30°] \times [-50°, +50°]$.

Duration. We split the study into 7 sessions (about 5 minutes each) with enforced breaks in between to avoid posture drifting. Every session was monitored to ensure the subject remained stationary below the neck. The study, including hardware setup, pre-study instructions, warm-up session (30 discarded trials), and breaks, took about 2.5 hours per participant. In total, we collected about 5 hours of time-synchronized motion-sEMG paired data.

Data processing and analysis. We aim to model and optimize neck muscle contraction driving head movements. However, raw sEMG signals cannot be directly used because they exhibit:

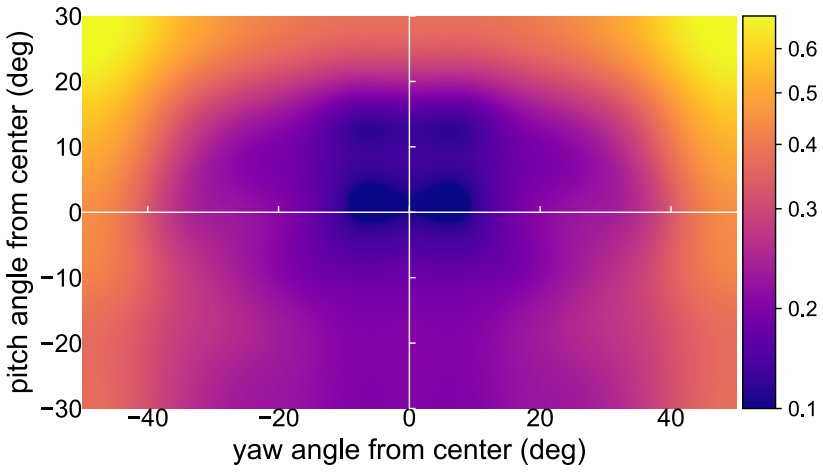


Figure 8.4: *MCL during stationary viewing.* Visualization of the user-aggregated MCL during stationary viewing.

1) frequency-dependent sensory noise; 2) oscillations between negative and positive values; 3) left-right asymmetry due to sensor positioning error [65, 242]; 4) cross-user difference in scale for the same head movement. Therefore, similar to prior literature [243, 244], we performed a series of sEMG signal processing, including detrending, bandpass filtering, and rectification, as well as inter-channel balancing, normalization, and integration. The processing details are integrated into Section 8.2.4. At each time frame, our processing pipeline outputs a single normalized muscle contraction value, integrated across processed sEMG signals from all four channels. Figure 8.3 illustrates the sEMG-to-MCL transformation with an example sequence.

8.1.3 Results

Stationary viewing ($|\Delta\mathbf{r}| = 0$). Figure 8.4 shows the MCL when the head remains static. The average normalized MCL was .32 ± .12. The head pose demanding the least MCL (.17 ± .02) was $\mathbf{r} = (0, 0)$, significantly lower than far-reaching poses. For instance, when the target was located at $\mathbf{r} = (30°, 50°)$, the MCL was higher at .68 ± .09. A repeated measures ANOVA showed that both p and y had a significant main effect on MCL ($F_{6,42} = 1.17, p < .001$ for p, $F_{8,56} = 1.5, p < .001$ for y). A significant $p \times y$ interaction effect was also observed ($F_{48,336} = 7.4, p < .001$). In particular,

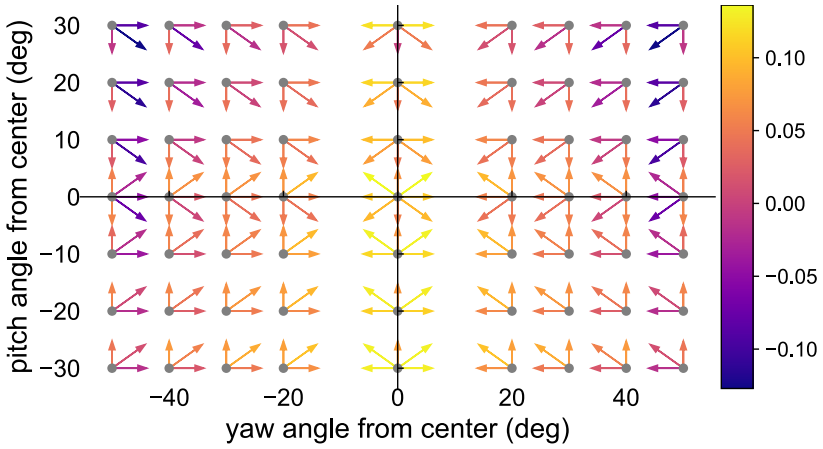


Figure 8.5: *ΔMCL during dynamic viewing.* Visualization of the movement-induced ΔMCL for dynamic viewing. Each ΔMCL, i.e., arrow, was computed by subtracting the MCL during stationary viewing at an anchor head pose from the average MCL during head movement from that anchor head pose (arrow tail) to a target head pose (arrow head). The arrow lengths were scaled to 13% for clearer visualization.

higher absolute values of yaw elevate the corresponding average MCL, from .21 ± .09 with $y = 0°$ to .47 ± .15 with $|y| = 50°$. A Mann-Kendall (M.K.) trend test showed a significant monotonic trend ($\tau = 1.0, p < .05$). On the other hand, the effect from pitch was asymmetric and non-monotonic. The highest values of p, $p = +30°/-30\circ$ induces MCL at .49 ± .16/.27 ± .09. An M.K. trend test did not show a significant monotonic trend of pitch angle's effect on MCL ($\tau = -.6, p = .07$).

Dynamic viewing ($|\Delta\mathbf{r}| > 0$). Figure 8.5 visualizes the ΔMCL during head movements, which was computed by subtracting the stationary MCL at an anchor from the average MCL during the movement. Introducing movements (i.e., non-zero $\Delta\mathbf{r}$) significantly elevated MCL up to 31.21% across all studied $\mathbf{r}$. In addition to $\mathbf{r}$, movement pattern $\Delta\mathbf{r}$ jointly influences the observed MCL. A repeated measures ANOVA showed that $\Delta p \times \Delta y$ has a significant main effect on MCL ($F_{3,21} = 23.44, p < .001$). For each time frame, we further extracted the angular acceleration in both directions $\boldsymbol{\alpha} \triangleq (\boldsymbol{\alpha}_p, \boldsymbol{\alpha}_y)$. Pearson correlation coefficients were computed to assess the relationship; There were positive correlations between MCL and $|\boldsymbol{\alpha}_p|$ ($r(12431) = .14, p < .001$), as well as MCL and $|\boldsymbol{\alpha}_y|$ ($r(12431) = .038, p < .001$). The elevation rate depends on individual $\mathbf{r}$. For example, the rate in yaw direction with movement starting at $\mathbf{r} = (30°, 50°)$ was 88.9% higher than $\mathbf{r} = (0°, 0°)$.

8.1.4 Discussion

The analysis above leads us to several observations and motivations for learning a computational model. First, despite individual participants' variances in muscular strengths and sizes, the measured MCL shares consistent trends for each condition, both during static viewing and dynamic movements. Second, the head pose (yaw y and pitch p) significantly influences MCL. In particular, despite the left-right symmetry with yaw, the pitch angle exhibits significantly asymmetric and non-monotonic effects on MCL. Lower pitch angles (i.e., heads facing downward) tend to reduce MCL. Third, in dynamic scenarios, increasing acceleration significantly elevates MCL. The elevation effect size depends on both the corresponding starting head pose and movement direction.

Similar to prior art [245], we hypothesize that the spatial-temporal muscle contraction levels alter both head holding and moving torques, as reflected in the sEMG-obtained MCL. These observations motivate us to develop a bio-physically inspired computational model by learning from the collected data.

8.2 Predicting Neck Muscle Contraction from Head Movements

The analysis of sEMG-motion paired data from the pilot study motivates us to establish a computational model relating head movements to neck muscle contraction level (MCL). Note that while MCL can be measured using sEMG sensors, they are 1) tedious and costly to deploy; 2) insufficient to *predict* MCL before a movement happens. Therefore, we first propose a bio-physically inspired MCL estimation model with open functions to characterize muscle-driven head motions in Section 8.2.1. Using the collected data, we then fit these functions with machine learning models to:

- Estimate the MCL associated with a completed head movement trajectory, i.e., *after* a movement happens (Section 8.2.2);
- Predict the MCL for a potential head movement using target head poses only, i.e., *before* a movement happens (Section 8.2.3).

8.2.1 A Bio-Physically Inspired MCL Model

Muscle-generated torque is proportional to MCL [246–248]. Our neck muscles actively generate the required amount of torque to enable head rotation at varying speeds. Denoting this **a**ctive torque as $\mathcal{T}_\mathbf{a} \in \mathbb{R}^2$, we establish a mapping $\mathcal{E}(\cdot)$ such that MCL $= \mathcal{E}(\mathcal{T}_\mathbf{a})$.

From our pilot study (Figure 8.4), maintaining the head stationary at various poses requires significantly different levels of MCL, and thus $\mathcal{T}_\mathbf{a}$. To maintain stationary viewing, however, there must be another pose-dependent torque that counterbalances $\mathcal{T}_\mathbf{a}$. We term this underlying torque as **p**assive torque $\mathcal{T}_\mathbf{p} \in \mathbb{R}^2$. We hypothesize that $\mathcal{T}_\mathbf{p}$ is induced by factors such as gravity and muscle relaxation. Notably, $\mathcal{T}_\mathbf{p}$ exists during both stationary and dynamic viewing conditions, but only depends on head poses, i.e., $\mathcal{T}_\mathbf{p} \triangleq \mathcal{T}_\mathbf{p}(\mathbf{r})$. By contrast, we proactively generate $\mathcal{T}_\mathbf{a}$ to perform stationary viewing (compensating $\mathcal{T}_\mathbf{p}$) and change our head pose at wish for dynamic viewing. Therefore, through the moment of inertia $I \in \mathbb{R}$, $\mathcal{T}_\mathbf{a}$ correlates both with the head pose $\mathbf{r}$ and angular acceleration $\boldsymbol{\alpha}$:

$$\mathcal{T}_\mathbf{p}(\mathbf{r}) + \mathcal{T}_\mathbf{a}(\mathbf{r}, \boldsymbol{\alpha}) = I \times \boldsymbol{\alpha}. \tag{8.1}$$

This further transforms the mapping $\mathcal{E}(\cdot)$ between $\mathcal{T}_\mathbf{a}$ and MCL:

$$\begin{aligned} \text{MCL} &= (\mathcal{E} \circ \mathcal{T}_\mathbf{a})(\mathbf{r}, \boldsymbol{\alpha}) = \mathcal{E}\left(I \times \boldsymbol{\alpha} - \mathcal{T}_\mathbf{p}(\mathbf{r})\right) \\ &\triangleq \mathcal{H}_m\left(\mathcal{E}, \mathcal{T}_\mathbf{p}, I, \mathbf{r}, \boldsymbol{\alpha}\right) \mapsto \mathbb{R}^+, \end{aligned} \tag{8.2}$$

where I, $\mathcal{T}_{\mathbf{p}}(\cdot)$, and $\mathcal{E}(\cdot)$ are the unknowns that map $\mathbf{r}/\boldsymbol{\alpha}$ to MCL.

8.2.2 Estimating MCL from Completed Motion Trajectories

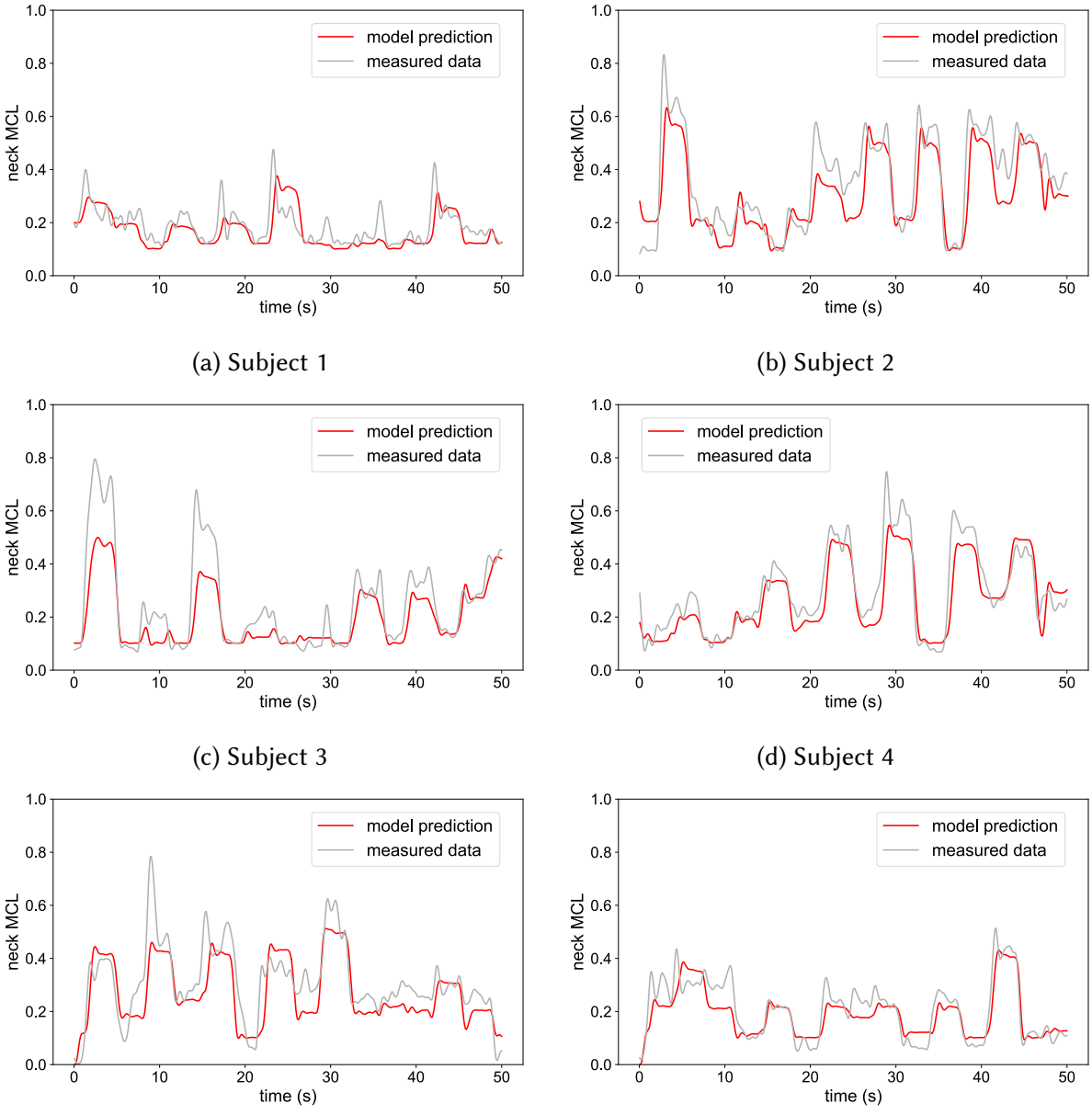
The data from our pilot study provide a large set of time-synchronized head movement trajectories and MCL sequences:

$$\mathbf{r}_t, \boldsymbol{\alpha}_t \mapsto \mathcal{H}_m\left(\mathcal{E}, \mathcal{T}_{\mathbf{p}}, I, \mathbf{r}_t, \boldsymbol{\alpha}_t\right). \tag{8.3}$$

Using these paired sequential data, we formulate an MCL regression problem and optimize 1D CNN models with L_2 loss to jointly approximate the unknowns I, $\mathcal{T}_{\mathbf{p}}(\cdot)$, and $\mathcal{E}(\cdot)$. The complete model, named *MCLNet*, is illustrated in Figure 8.7a.

Notably, a phenomenon called electro-mechanical delay exists between sEMG signals and muscular motions. Depending on individuals and muscle areas, the delay can incur a temporal offset between the two modalities up to 100ms [249]. To accommodate this temporal inconsistency for robust prediction, our model takes in motion sequences with T=400ms window and predicts MCL for the central 200ms interval, i.e., inputs cover additional 100ms outputs from the beginning and end. Given a sequence of uniformly sampled head poses $\{\mathbf{r}^t\}_{t=1}^T$, we first calculate the corresponding angular accelerations $\{\boldsymbol{\alpha}^t\}_{t=1}^T$ through finite difference, then execute our model at each t to obtain the overlapping sequences of predicted MCL. Figure 8.7b visualizes the prediction-measurement comparison for an example trajectory taken from the test set. Visualized results for each subject are shown in Figure 8.6.

So far, a core requirement of MCLNet is the prior knowledge of completed head movement trajectories. However, to benefit real-life applications such as UX design and cinematography, we shall reduce the potential discomfort before deploying to users. To this end, we further extend our model to *forecast* MCL before a movement.



(a) Subject 1

(b) Subject 2

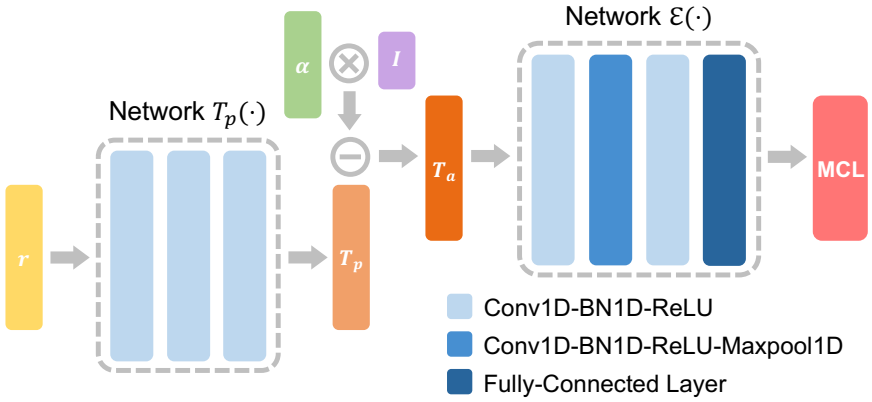
(c) Subject 3

(d) Subject 4

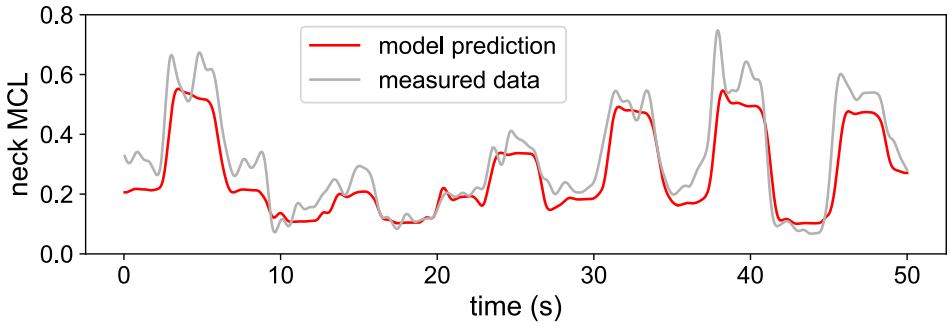
(e) Subject 5

(f) Subject 6

Figure 8.6: *Qualitative MCL estimation results.* Each panel compares model-predicted and hardware-measured neck MCL for one of the six subjects who contributed evaluation data. The sequences shown here are randomly sampled from each subject's held-out evaluation set.



(a) Illustration of the MCLNet architecture design



(b) Model-predicted vs. hardware-measured neck MCL

Figure 8.7: *MCLNet architecture and example MCL estimation results.* The top panel illustrates the architecture design of MCLNet for jointly learning I, $\mathcal{T}_p(\cdot)$, and $\mathcal{E}(\cdot)$ from paired head-motion and MCL data. The bottom panel compares the model-predicted and hardware-measured MCL for a representative test sequence.

8.2.3 Predicting MCL from Target Head Poses

Given the starting and ending head poses $\{\mathbf{r_s}, \mathbf{r_e}\}$ of a uni-directional head movement, the required MCL to travel between them is determined by the actual movement trajectory. However, as evidenced by our analysis in Section 8.1.4, $\{\mathbf{r_s}, \mathbf{r_e}\}$ alone carry significant influence on the overall MCL. Therefore, using our collected data, we regress a representative motion trajectory for each pair of $\{\mathbf{r_s}, \mathbf{r_e}\}$, to approximate the temporal patterns of angular velocity $\boldsymbol{\omega}_t$:

$$\boldsymbol{\omega}_t(\mathbf{r_s}, \mathbf{r_e}) \in \mathbb{R}^2, \;\; \text{s.t.} \;\; \mathbf{r_s} + \int_{t=t_s}^{t_e} \boldsymbol{\omega}_t(\mathbf{r_s}, \mathbf{r_e}) \mathrm{d}t = \mathbf{r_e}, \tag{8.4}$$

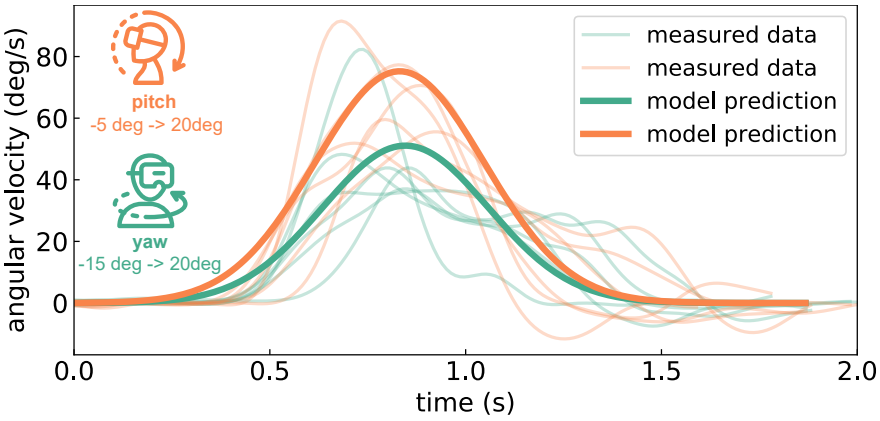


Figure 8.8: *Predicting head motion trajectories from starting and ending head poses.* Light and dark curves show tracked and predicted angular velocities, respectively. Orange and green curves correspond to pitch and yaw directions, respectively.

Motivated by prior literature studying the main sequence effect of head movements [250] and our observations of a single main peak in each velocity profile (Figure 8.8), we perform a unimodal Gaussian approximation for the angular velocity:

$$\boldsymbol{\omega}_t^i(\mathbf{r_s}, \mathbf{r_e}) \triangleq A^i(\mathbf{r_s}, \mathbf{r_e}) e^{-\frac{\left(t-\mu^i(\mathbf{r_s},\mathbf{r_e})\right)^2}{2\left(\sigma^i(\mathbf{r_s},\mathbf{r_e})\right)^2}}, i \in \{p, y\}, \tag{8.5}$$

Using our collected data, we formulate a trajectory regression problem and optimize a Multi-Layer Perceptron (MLP) model, annotated as *TrajectoryNet*, to predict $\{A^i, \mu^i, \sigma^i\}_{i\in\{p,y\}}$ given an arbitrary pair of head poses $\{\mathbf{r_s}, \mathbf{r_e}\}$. Using predicted angular velocity curves, we can approximate the overall MCL:

$$\mathrm{H}_c(\mathbf{r_s}, \mathbf{r_e}) = \int_{t=t_s}^{t_e} \mathcal{H}_m \left(\mathcal{E}, \mathcal{T}_\mathrm{p}, I, \mathbf{r_s} + \int_{\bar{t}=t_s}^{t} \boldsymbol{\omega}_{\bar{t}}(\mathbf{r_s}, \mathbf{r_e}) \mathrm{d}\bar{t}, \dot{\boldsymbol{\omega}}_t \right) \mathrm{d}t, \tag{8.6}$$

Figure 8.8 compares the velocity curves collected from our users with TrajectoryNet's predictions over an example pair of $\{\mathbf{r_s}, \mathbf{r_e}\}$ taken from the test set. The data synchronization, sEMG processing, network architectures, and training details used for both MCLNet and TrajectoryNet are summarized in Section 8.2.4.

8.2.4 sEMG Signal Processing and Model Implementation Details

Following standard sEMG signal processing approaches [243, 244], we transformed the collected four-channel raw sEMG signals into an overall neck muscle contraction level, or MCL. As summarized in Figure 8.9, the pipeline detrends the raw signals, applies band-pass and low-pass filtering, computes RMS envelopes, and performs left-right normalization before integrating the channel-wise responses into a single normalized contraction measure. Each user's total MCL was linearly normalized to $[0, 1]$ to mitigate muscular-strength and morphology differences across participants, allowing the data from multiple users to be analyzed and learned jointly.

sEMG and head pose data synchronization. To collect sEMG signals from both sides of the sternocleidomastoid (SCM) and splenius capitis (SC) muscles, we used four Delsys Trigno wireless sEMG sensors synchronized at 2000 Hz. The captured sEMG signals were streamed through a dedicated sEMG base station to a Python program running on a desktop PC. Head-pose data were collected from the Oculus Quest 2 at 90 Hz and transmitted from Unity running on the same machine. After sEMG processing, the two modalities were synchronized using system timestamps and re-sampled to 20 Hz.

Trajectory regression. TrajectoryNet takes in a pair of starting and ending head poses $\{\mathbf{r_s} \in \mathbb{R}^2, \mathbf{r_e} \in \mathbb{R}^2\}$ and outputs the parameters of predicted Gaussian-shaped angular velocity curves for both pitch and yaw directions $\{A^i, \mu^i, \sigma^i\}_{i\in\{p,y\}}$. Specifically, $\{\mathbf{r_s}, \mathbf{r_e}\}$ are first transformed into $\{\mathbf{r_s}, \mathbf{r_e} - \mathbf{r_s}\}$ and then concatenated into a vector of length 4 before feeding into TrajectoryNet for accelerated convergence during training and improved generalization ability of the model. To obtain the head motion trajectory $\mathbf{r}_t(\mathbf{r_s}, \mathbf{r_e})$ associated with the predicted angular velocity $\boldsymbol{\omega}_t(\mathbf{r_s}, \mathbf{r_e})$, we integrate it over $[t_s = 0, t_e]$, where t_e is determined such that:

$$\mathbf{r_s} + \int_{t=0}^{t_e} \boldsymbol{\omega}_t(\mathbf{r_s}, \mathbf{r_e})\mathrm{d}t = \mathbf{r_e}. \tag{8.7}$$

sEMG to MCL Data Processing Pipeline

```
function sEMG2MCL(sEMG)
    ▷ 4-channel sEMG signals
    t = GetTimestamps(sEMG)                          ▷ sEMG timestamps
    axis = 0
    bandpass_freq_low = 20 Hz
    bandpass_freq_high = 150 Hz
    lowpass_freq = 1 Hz
    butterworth_order = 4
    ▷ Constant detrending
    sEMG = sEMG − Mean(sEMG, axis)
    ▷ 4th order Butterworth bandpass filter
    sampling_rate = Len(sEMG)/(t[−1] − t[0])
    low = (2 * bandpass_freq_low)/sampling_rate
    high = (2 * bandpass_freq_high)/sampling_rate
    sEMG = sos_bp(sEMG, order, [low, high], axis)
    ▷ RMS envelope filter
    window = [1/1000, ..., 1/1000]                   ▷ 1000-length
    for i ← [0, 4) do:
        sEMG[i] = √(conv(sEMG[i]^2, window))
    ▷ 4th order Butterworth lowpass filter
    low = (2 * lowpass_freq)/sampling_rate
    sEMG = sos_lp(sEMG, order, low, axis)
    ▷ Left/Right normalization
    sEMG[0] = max(sEMG[1])/max(sEMG[0])
    sEMG[2] = max(sEMG[3])/max(sEMG[2])
    ▷ Compute overall MCL
    axis = 1
    return sum(sEMG, axis)                           ▷ Channel-wise sum
end function
```

Figure 8.9: *Our sEMG-to-MCL data processing pipeline.* The illustrated function filters and transforms multi-channel raw sEMG signals into overall neck MCL.

MCL estimation. The inputs to our MCLNet are composed of a head motion sequence of T=400ms, which is equivalent to 8 samples of pitch/yaw angles under the re-sampling rate of 20 Hz, and the corresponding angular acceleration values computed via finite difference (8 samples of pitch/yaw angular acceleration). Given the pair of input head motion and angular acceleration sequences (T=400ms, 8 samples for each), MCLNet outputs the instantaneous MCL for the central 200ms interval (4 samples under the sampling rate of 20 Hz), i.e., compared to the outputs, the inputs cover additional 100ms in both directions of the time axis.

TrajectoryNet and MCLNet architectures. TrajectoryNet consists of 3 repeating FullyConnected-BatchNorm1D-ReLU blocks (the number of channels was set to 20, and the last block does not contain BatchNorm1D layer). The passive torque module $\mathcal{T}_{\mathrm{p}}(\cdot)$ and torque-to-MCL module $\mathcal{E}(\cdot)$ of MCLNet were approximated by two 1D convolutional neural network (CNN) models. Specifically, $\mathcal{T}_{\mathrm{p}}(\cdot)$ consists of 3 repeating Convolution1D-BatchNorm1D-ReLU blocks (number of channels was set to 20, and the last block does not contain ReLU activation); $\mathcal{E}(\cdot)$ consists of 3 repeating Convolution1D-BatchNorm1D-ReLU blocks (number of channels was set to 20), with the second one followed by a 1D MaxPooling layer (both kernel size and stride were set to 2) and the last one followed by a 20×1 FullyConnected layer.

TrajectoryNet and MCLNet training. TrajectoryNet was optimized using L_2 loss and Adam optimizer ($\beta_1 = 0.9, \beta 2 = 0.999$) for 25 epochs. The learning rate started at $1e^{-3}$ and dropped by 10× at epoch 15. The batch size was set to 64. A weight decay factor of $1e^{-5}$ was enforced to reduce over-fitting. MCLNet was optimized using L_2 loss and Adam optimizer ($\beta_1 = 0.9, \beta 2 = 0.999$) for 20 epochs. The learning rate started at $1e^{-3}$ and dropped by 10× at epoch 10. The batch size was set to 64. A weight decay factor of $5e^{-4}$ was enforced to reduce over-fitting.

8.3 Evaluating Muscle Contraction Level Prediction Performance

We present a series of objective measurements on the model performance in predicting neck MCL. We first evaluate MCLNet's estimation accuracy with completed head motion trajectories in Section 8.3.1, then extend to predict MCL *before* head movements occur by incorporating TrajectoryNet in Section 8.3.2.

8.3.1 MCL Estimation After Head Movement Completion

Experimental setup. We leveraged our data from the conditions detailed in Section 8.1.2 to train our model. Additionally, during the pilot study, we also collected two groups of conditions to establish an evaluation dataset with unseen conditions. It contains 4 pitch and 4 yaw angles, $p \in \{\pm 25°, \pm 5°\}, y \in \{\pm 45°, \pm 15°\}$, resulting in a total of 16 anchor head poses. The same 8 surrounding targets with travel angle $\Delta p \in \{\pm 35°, 0°\}, \Delta y \in \{\pm 25°, 0°\}$ were introduced to each of them. Note that the evaluation conditions were designed to contain no overlap with the training set. Due to the extra long collection process and thus scheduling conflicts, 6 (3 female) of the 8 participants completed the evaluation condition session. We adopted their data for this experiment. Two quantitative metrics were considered: Normalized Root-Mean-Square Error (NRMSE) and Normalized Mean Absolute Error (NMAE). These metrics measure the error ratio between the model-predicted and hardware-measured MCL values, where the hardware measurements were obtained through the sEMG processing pipeline summarized in Section 8.2.4. Lower values therefore indicate better model performance.

Results and discussion. MCLNet achieves an overall performance of 12.39 ± 4.74% NRMSE and 9.54 ± 4.14% NMAE across all 6 subjects and 16 anchor head poses. Figure 8.10 summarizes its subject-wise performance, while Figure 8.6 provides representative qualitative comparisons

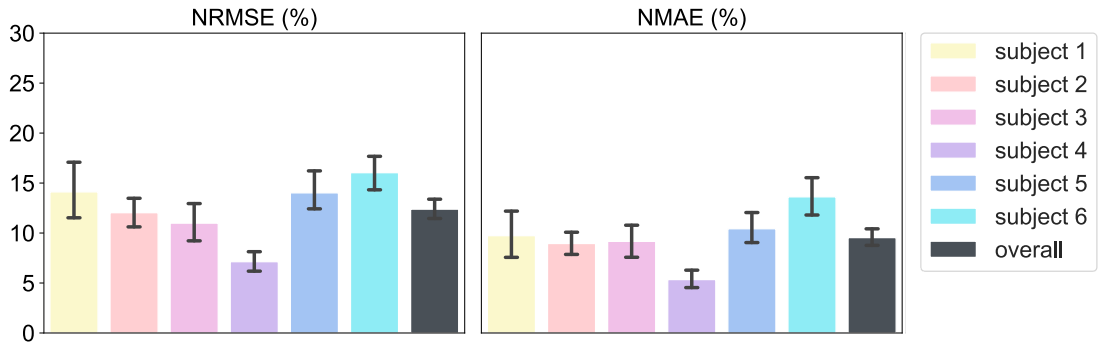
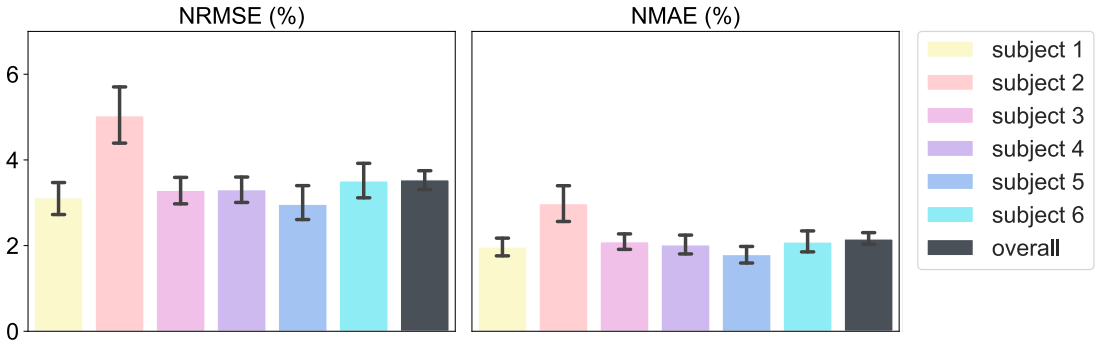


Figure 8.10: *Performance of MCLNet in estimating neck MCL from completed head motion trajectories.* The box plots summarize NRMSE and NMAE across subjects for unseen evaluation conditions when completed head motion trajectories are available.

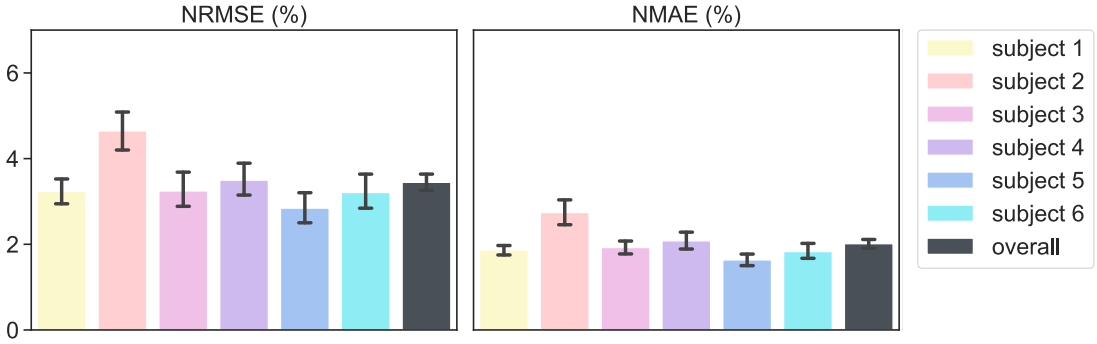
for all six evaluated subjects. Beyond the average accuracy, we further measure whether the model predicts the elevation and reduction of MCL consistently with the hardware measurement. We therefore analyze Pearson's and Spearman's coefficients between the two conditions. The results indicate a significant correlation between model predictions and hardware measurements ($r(70066) = .62, p < .001$ and $r(70066) = .60, p < .001$). Together, these analyses validate our method's effectiveness in estimating neck MCL when complete head motion trajectories are known beforehand.

8.3.2 MCL Prediction Before Head Movements Occur

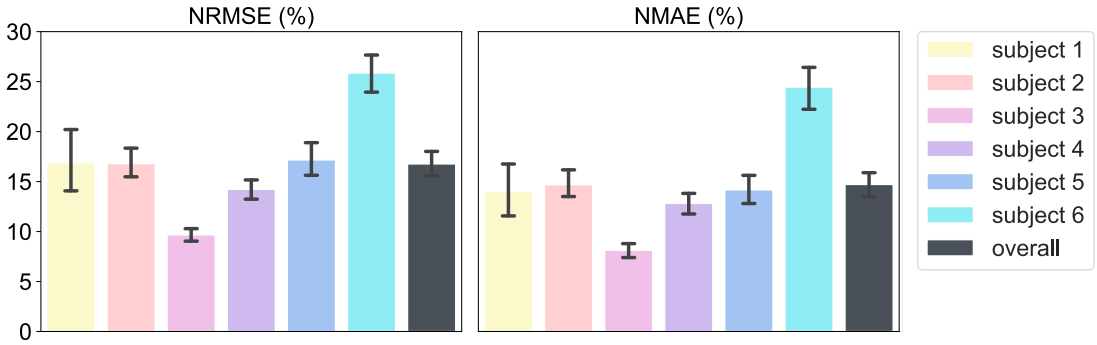
Experimental setup. For this experiment, we first used the head motion data from Section 8.3.1 to optimize and validate TrajectoryNet (Section 8.2.3), then combined it with MCLNet to predict MCL using target head poses only. Since our dataset consists of sequences of stationary head pose followed by pose-changing movements, we extracted the dynamic part to construct a dataset of starting/ending head poses $\{\mathbf{r_s}, \mathbf{r_e}\}$ paired with HMD-tracked trajectories. The train-test data split from Section 8.3.1 was used for evaluation.



(a) Performance of TrajectoryNet in predicting pitch angular velocity



(b) Performance of TrajectoryNet in predicting yaw angular velocity



(c) Performance of MCLNet +TrajectoryNet in predicting neck MCL

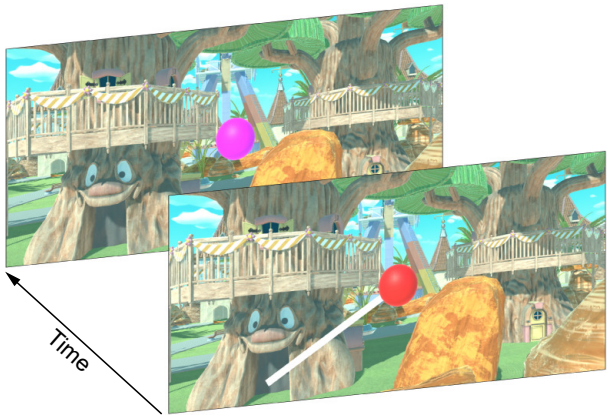
Figure 8.11: *Performance of MCLNet +TrajectoryNet in predicting neck MCL using target head poses only.* The first two panels report TrajectoryNet's pitch- and yaw-velocity prediction accuracy, while the third panel shows the resulting MCL prediction performance after combining MCLNet with TrajectoryNet.

Results and discussion. The performance of our MCL prediction framework, composed of trajectory regression and MCL estimation, is shown in Figure 8.11. TrajectoryNet achieves an overall NRMSE/NMAE of $3.54 \pm 1.11\%/2.16 \pm 0.65\%$ in pitch velocity and $3.45 \pm 0.98\%/2.01 \pm 0.51\%$ in yaw. The overall MCL prediction performance is $16.76 \pm 6.05\%$ NRMSE and $14.71 \pm 5.96\%$ NMAE. Pearson's and Spearman's coefficients between the two conditions are $r(70066) = .59, p < .001$ and $r(70066) = .57, p < .001$, indicating a significant correlation. The results above demonstrate that our method can reliably predict the potential MCL with only the target head poses, before the actual movement occurs.

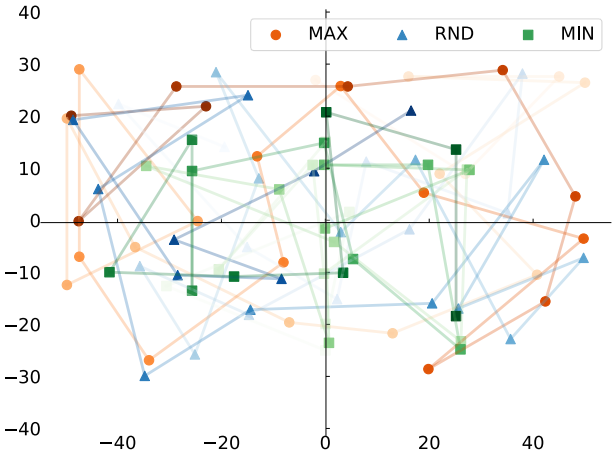
8.4 Reducing Neck Muscle Discomfort via Content Layout Optimization

Participants and setup. We recruited 13 participants (ages $20 - 35$, 6 female). None of them were aware of the hypothesis, the research, or the number of conditions. One participant came in with a prior condition of neck injury and was excluded during warm-up sessions. There was no overlap between these participants and those from the pilot study in Section 8.1. We conducted the study using an Oculus Quest 2 HMD (*without EMG*). During the study, participants observed stimuli through the HMD and were instructed to remain seated while keeping their torso stationary. The study took around 60 minutes for each participant, including breaks between sessions.

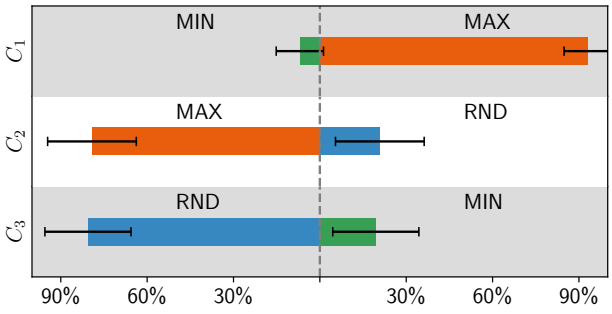
Stimuli. As shown in Figure 8.12a, we developed and experimented with a 3D balloon-popping game with target acquisition. The stimuli were displayed as a sequence of red balloon targets rendered in an amusement park scene. As an indicator, the target color was changed to magenta upon a participant's fixation. A line pointing toward the next target was displayed to guide the user.



(a) Task setting and stimuli



(b) Visualization of study conditions



(c) Aggregated voting results

Figure 8.12: *User study stimuli and results.* The top panel illustrates the balloon-popping task setting and stimuli. The middle panel visualizes the angular distributions of the visual targets for three example scan paths. Color gradients indicate temporal order. All three conditions share the same total head rotation budget while maintaining broad visual-field coverage. The bottom panel summarizes the voting results for the three pairwise comparisons. Individual voting results are reported in Table 8.1. 3D asset credit: Mixall (Unity).

Tasks. The task was designed as a two-alternative forced-choice procedure to avoid the bias often associated with scaled ratings, similar to prior works measuring muscular discomfort [251, 252]. During each session, the balloon targets were displayed one at a time, following a pre-defined scan path with 31 targets in different directions $\mathcal{S} = \{\mathbf{r}_t, t = 0, \ldots, 30, \mathbf{r}_i \in \mathcal{R}\}$, similar to the definition in Section 8.1.2. Participants were instructed to rotate their heads to fixate on each balloon until it disappeared. To trigger both dynamic and stationary head status, a 1-second fixation on each target was enforced before the next one appeared at $\mathbf{r}_{t+1}$. Each session contained a pair of two sequentially tested $\mathcal{S}$ instances generated from 2 out of the 3 conditions described below. After each session, the participants used the keyboard to indicate which of the two scan paths felt more uncomfortable, tiring, or difficult for the neck.

Conditions. Guided by our model, we designed 3 conditions of progressively generated scan paths. They were created to ensure an identical total head rotations and similar spatial coverage but varied cumulative MCL. To ensure fair movements across conditions, we randomly pre-partitioned a fixed amount of total head rotations 900° into 30 steps. Then, at each step t, the next target pose $\mathbf{r}_{t+1}$ was chosen from a set of candidate poses by maximizing the corresponding score function of the selected condition:

- **MAX**: $\mathrm{C}(\mathbf{r}_{i\in[1,t+1]}) + \mathrm{H}_c(\mathbf{r}_t, \mathbf{r}_{t+1})$
- **RND**: $\mathrm{C}(\mathbf{r}_{i\in[1,t+1]})$
- **MIN**: $\mathrm{C}(\mathbf{r}_{i\in[1,t+1]}) - \mathrm{H}_c(\mathbf{r}_t, \mathbf{r}_{t+1})$

Here, C is a term to ensure full visual field coverage for condition-wise fairness. On average, the ratio between the cumulative MCL of the three conditions **MAX**/**RND**/**MIN** was 3.48 vs. 1.95 vs. 1.00. The algorithm used to generate those conditions is summarized in Figure 8.13, and Figure 8.12b shows one example of each condition. The three conditions generate 3 different pairs for two-interval forced-choice (2IFC) comparisons, namely $\mathbf{C_1}$: **MAX** vs. **MIN**; $\mathbf{C_2}$: **RND** vs. **MAX**;

User Study Condition Generation

```
function GetNextPose(r_s, t, θ°, Condition)
    ▷ Unif. sample 180 head poses that are θ° from r_s
    poses = UniformSample(r_s, θ°, 180)
    all_traj = []
    for r_e ← poses do:
        ▷ Predict the head trajectory from r_s to r_e
        trajectory = TrajectoryNet(r_s, r_e)
        ▷ Include 1-second fixation on r_e
        trajectory = AddFixation(trajectory, r_e)
        all_traj.Append(trajectory)
    ▷ Pick candidate with highest score as next pose
    maxScore = −Inf
    Init r                                  ▷ Current candidate for next pose
    for i ← [0, ..., len(all_traj)-1] do:
        r_e = GetEndPose(all_traj[i])
        if Condition == MAX:
            score = C(r_{j∈[1,t]}, r_e) + H_c(all_traj[i])
        if Condition == RND:
            score = C(r_{j∈[1,t]}, r_e)
        if Condition == MIN:
            score = C(r_{j∈[1,t]}, r_e) − H_c(all_traj[i])
        ▷ Update r if score greater than maxScore
        if score > maxScore:
            maxScore = Score
            r = r_e
    return r
end function
```

Figure 8.13: *User study condition generation algorithm.* Given the current head pose $\mathbf{r_e}$ on a scan path, this function extends the scan path by generating the optimal next head pose $\mathbf{r}$ for one of the 3 conditions **MAX**/**MIN**/**RND**.

C_3: **MIN** vs. **RND**. We repeated the random pre-partition of total head rotations and condition generation process to get 6 sets of **MAX**/**RND**/**MIN** conditions for a total of 18 sessions. The appearance order of the 18 sessions was randomized and counter-balanced across participants, same for the 2 conditions within each session, to avoid bias.

Results. Table 8.1 and Figure 8.12c show the individual votes and the summary for each comparison, respectively. By aggregating all sessions, **MAX**/**RND**/**MIN** were 86.1%/50.7%/13.1% voted as being more uncomfortable in the related comparisons. Among all comparisons, the difference was significantly higher than a random guess (50%). By analyzing individual votes, a repeated measures ANOVA indicated that the condition had a statistically significant effect on the votes ($F_{2,22} = 89.46, p < .001$). Post-hoc pairwise t-tests with Holm adjustments showed that the difference was significant among all 3 comparisons ($p < .001$ for all conditions).

Discussion. The analysis above shows the significant difference in participants' subjectively perceived discomfort levels among the three conditions. The participant-rated discomfort levels, **MAX** > **RND** > **MIN**, also matched our model's prediction (H_c). Note that the significant difference was not induced by head rotation angles which were ensured to be identical via our progressive trajectory generation. These results demonstrated our model's capability of predicting a user's neck discomfort with target head poses only, i.e., before the head movement takes place.

Table 8.1: *User study 2IFC results.* The numbers indicate, for each participant and condition, how many times that condition was judged more uncomfortable across the 18 two-alternative forced-choice (2IFC) trials.

cond \ user	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12
MAX	11	12	11	12	10	10	9	12	9	10	9	9
RND	5	5	6	4	4	7	8	6	7	6	6	9
MIN	2	1	1	2	4	1	1	0	2	2	3	0

8.5 Discussion

This chapter demonstrates that neck muscle workload in immersive viewing can be measured, modeled, and predicted in a way that is directly useful for interface design. Starting from sEMG-recorded muscle activity during head movement in XR, we derived a biophysically inspired model for estimating muscle contraction level after motion and forecasting it before motion using target head poses alone. The combination of objective evaluations and subjective user study results shows that our model captures meaningful ergonomic differences among viewing conditions and can support the design of more comfortable immersive experiences.

One important outcome is methodological. This work moves beyond descriptive reports that HMDs increase neck discomfort by offering a computational route from head pose and motion to predicted muscular effort. That route matters because many design decisions in XR are spatial: where should targets be placed, how demanding is a scan path, and how should content be arranged across a wide field of view? By estimating MCL from completed trajectories and forecasting it from candidate target layouts, this chapter provides a quantitative basis for answering those questions rather than relying on intuition alone.

This work also reveals several substantive observations about ergonomic behavior in XR. Static head pose already carries a nontrivial cost, and that cost varies systematically with pitch and yaw. Dynamic motion further increases workload, with the magnitude of the increase depending on both acceleration and starting pose. Those findings help explain why two layouts with similar geometric coverage can still feel very different to users. The user study confirms this point: scan paths generated to maximize predicted workload were consistently perceived as more uncomfortable than those generated to minimize it, even when overall head-rotation budgets were controlled.

At the same time, this chapter remains a first step. The present model focuses on yaw and pitch, leaving roll largely unexplored. It also concentrates on the neck, even though discomfort in immersive systems may involve shoulders, upper back, and broader postural adaptations.

Extending our approach to additional muscle groups and whole-body tasks would make the model more complete. The trajectory-regression stage also relies on representative Gaussian velocity profiles, which capture the dominant behavior in the collected data but may not fully represent individual variability or more complex task dynamics.

Another limitation is ecological scope. The studies were designed around controlled target-reaching and scan-path tasks so that MCL could be measured and modeled cleanly. Real applications may involve cognitive load, scene semantics, posture changes, and interaction styles that alter movement behavior. Future work could therefore combine ergonomic modeling with richer task models, probabilistic motion prediction, or adaptive systems that personalize layouts based on a user's physical condition and historical behavior.

Even with these limitations, this chapter advances the dissertation's broader argument that immersive graphics systems should be designed around human capabilities and constraints. Here the relevant constraint is physical sustainability over time. If an immersive experience is visually compelling but mechanically fatiguing, it is not yet fully human-centered. By modeling neck muscle contraction as a design signal, we move toward immersive graphics systems that are not only perceptually effective, but also ergonomically responsible.

Chapter 9

Conclusion

This dissertation argues that multimodal interactive graphics systems should be designed not only around computational efficiency or representational power, but also around the perceptual, behavioral, and physical characteristics of their users. Across the six technical chapters, we explored this argument from multiple directions: how visual content can be represented more adaptively, how rendering resources can be allocated according to perceptual importance, how generative models can be guided by human attention, how visual-auditory cues shape reaction performance, how finger force can become an interaction signal for immersive systems, and how muscular workload can inform more comfortable immersive experiences. These topics are connected by a shared perspective: human-centered signals should function as design principles throughout the graphics pipeline.

One contribution of this dissertation is to show that this perspective applies across different stages of interactive graphics. In the visual domain, the work on content-adaptive image representation and perceptually guided 3DGS demonstrates that representation and rendering need not be optimized solely for machine-oriented fidelity or static resource budgets. Instead, they can be structured around perceptual sensitivity, spatial structure, and viewing conditions. The work on saliency-guided image generation extends this idea further by moving from efficient representation

and rendering to controllable content creation, showing that generative models can be directed not only toward image realism, but also toward intended patterns of visual attention. Taken together, these chapters suggest that perception-aware graphics is not a narrow optimization problem, but a broader way of deciding what content matters, where it matters, and how computational effort should be allocated accordingly.

A second contribution is to broaden the notion of perception-aware graphics beyond vision alone. The chapter on visual-auditory reaction latency shows that multimodal cueing can be modeled in a probabilistic way that supports content analysis and interface design. This moves the discussion from what users see to how quickly they detect and respond to events across modalities. The chapter on force-aware interfaces further extends the dissertation into embodied interaction, demonstrating that immersive systems can benefit from sensing what users physically do, not merely where their hands appear to be. Finally, the XR ergonomics chapter highlights that a system is not fully human-centered if it remains perceptually effective but physically fatiguing. Graphics systems increasingly operate inside closed loops that include looking, listening, reacting, exerting force, and sustaining physical effort over time. Designing for such systems therefore requires reasoning about perception, action, and comfort together rather than in isolation.

Viewed together, these results point toward a broader research agenda for multimodal interactive graphics. Future systems should become increasingly adaptive, integrating perceptual models, behavioral prediction, interaction sensing, and ergonomic reasoning within shared computational frameworks. This creates several promising directions. Perceptual rendering and learned representations can be made more personalized by conditioning on individual differences in attention, sensitivity, and task context. Generative models can move beyond static saliency control toward closed-loop content adaptation that responds to gaze, reaction behavior, and user intent in real-time. Force-aware and ergonomics-aware interaction methods can be integrated with richer models of hand pose, whole-body movement, and long-term usage patterns, enabling immersive systems that support both dexterity and physical sustainability. More broadly, the

boundary between graphics algorithms and user models should continue to weaken, allowing future systems to treat human response as part of the computation rather than as an external criterion applied only after the fact.

At a high level, this dissertation presents multimodal interactive graphics as a human-in-the-loop discipline. The central challenge is no longer simply to render more detail, generate more content, or build more capable interaction techniques in isolation. It is to build systems that decide what to represent, what to prioritize, how to guide attention, how to support action, and how to remain comfortable in ways that are coherent with the people who use them. We see this as an important direction for graphics, XR, and AI-assisted media alike. By grounding computation in human perception and behaviors, we can build interactive graphics systems that are not only more efficient and more powerful, but also more natural, more responsive, and ultimately more meaningful to the users they are designed to serve.

Bibliography

[1] Michael F Deering. The limits of human vision. In *2nd International Immersive Projection Technology Workshop*, volume 2, 1998.

[2] LN Thibos, FE Cheney, and DJ Walsh. Retinal limits to the detection and resolution of gratings. *Journal of the Optical Society of America A*, 4(8):1524–1529, 1987.

[3] Cynthia Owsley. Contrast sensitivity. *Ophthalmology Clinics of North America*, 16(2): 171–177, 2003.

[4] Rafał K Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. Fovvideovdp: A visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics (TOG)*, 40(4):1–19, 2021.

[5] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. Foveated 3d graphics. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[6] David R Walton, Rafael Kuffner Dos Anjos, Sebastian Friston, David Swapp, Kaan Akşit, Anthony Steed, and Tobias Ritschel. Beyond blur: Real-time ventral metamers for foveated rendering. *ACM Transactions on Graphics*, 40(4):1–14, 2021.

[7] Sabine Kastner and Leslie G Ungerleider. Mechanisms of visual attention in the human cortex. *Annual Review of Neuroscience*, 23(1):315–341, 2000.

[8] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112 (52):16054–16059, 2015.

[9] Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. Saliency in vr: How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1633–1642, 2018.

[10] Hans Colonius and Adele Diederich. The optimal time window of visual-auditory integration: a reaction time analysis. *Frontiers in Integrative Neuroscience*, page 11, 2010.

[11] David Burr and David Alais. Combining visual and auditory information. *Progress in Brain Research*, 155:243–258, 2006.

[12] Marc O Ernst and Martin S Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

[13] Joanna Bergström and Kasper Hornbæk. Human–computer interaction on the skin. *ACM Computing Surveys (CSUR)*, 52(4):1–14, 2019.

[14] Terry A Welch. High speed data compression and decompression apparatus and method, December 10 1985. US Patent 4,558,302.

[15] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[16] Texture block compression in direct3d 11, 2024. `https://learn.microsoft.com/en-us/windows/win32/direct3d11/texture-block-compression-in-direct3d-11`.

[17] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-Time Rendering*. AK Peters/CRC Press, 2019.

[18] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[19] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020.

[20] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[21] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41 (4):1–15, 2022.

[23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4): 1–14, 2023.

[24] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. MIT Press, 2023.

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[26] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[27] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022.

[28] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021.

[29] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision*, pages 333–350. Springer, 2022.

[30] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[33] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[35] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[36] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[37] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[38] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *arXiv preprint arXiv:2305.16322*, 2023.

[39] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (11):1254–1259, 1998.

[40] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. *Advances in Neural Information Processing Systems*, 19, 2006.

[41] Tilke Judd, Krista Ehinger, Frédo Durand, and Antonio Torralba. Learning to predict where humans look. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2106–2113. IEEE, 2009.

[42] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1072–1080, 2015.

[43] Wenguan Wang, Jianbing Shen, Jianwen Xie, Ming-Ming Cheng, Haibin Ling, and Ali Borji. Revisiting video saliency prediction in the deep learning era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):220–237, 2019.

[44] Sen Jia and Neil DB Bruce. Eml-net: An expandable multi-layer network for saliency prediction. *Image and Vision Computing*, 95:103887, 2020.

[45] Benjamin Long, Sue Ann Seah, Tom Carter, and Sriram Subramanian. Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Transactions on Graphics (TOG)*, 33(6):1–10, 2014.

[46] Xinge Yu, Zhaoqian Xie, Yang Yu, Jungyup Lee, Abraham Vazquez-Guardado, Haiwen Luan, Jasper Ruban, Xin Ning, Aadeel Akhtar, Dengfeng Li, et al. Skin-integrated wireless haptic interfaces for virtual and augmented reality. *Nature*, 575(7783):473–479, 2019.

[47] Minglu Zhu, Zhongda Sun, Zixuan Zhang, Qiongfeng Shi, Tianyiyi He, Huicong Liu, Tao Chen, and Chengkuo Lee. Haptic-feedback smart glove as a creative human-machine interface (hmi) for virtual/augmented reality applications. *Science Advances*, 6(19):eaaz8693, 2020.

[48] Yudai Tanaka, Alan Shen, Andy Kong, and Pedro Lopes. Full-hand electro-tactile feedback without obstructing palmar side of hand. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2023.

[49] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, 2019.

[50] Eric J Gonzalez, Eyal Ofek, Mar Gonzalez-Franco, and Mike Sinclair. X-rings: A hand-mounted 360° shape display for grasping in virtual reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 732–742, 2021.

[51] Angkoon Phinyomark and Erik Scheme. Emg pattern recognition in the era of big data and deep learning. *Big Data and Cognitive Computing*, 2(3):21, 2018.

[52] Yilin Liu, Shijia Zhang, and Mahanth Gowda. Neuropose: 3d hand pose tracking using emg wearables. In *Proceedings of the Web Conference 2021*, pages 1471–1482, 2021.

[53] Yunxiang Zhang, Benjamin Liang, Boyuan Chen, Paul M Torrens, S Farokh Atashzar, Dahua Lin, and Qi Sun. Force-aware interface via electromyography for natural vr/ar interaction. *ACM Transactions on Graphics (TOG)*, 41(6):1–18, 2022.

[54] Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. Towards foveated rendering for gaze-tracked virtual reality. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016.

[55] Anton S Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[56] Okan Tarhan Tursun, Elena Arabadzhiyska-Koleva, Marek Wernikowski, Radosław Mantiuk, Hans-Peter Seidel, Karol Myszkowski, and Piotr Didyk. Luminance-contrast-aware foveated rendering. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019.

[57] Maliha Ashraf, Rafał K Mantiuk, Alexandre Chapiro, and Sophie Wuerger. castlecsf—a contrast sensitivity function of color, area, spatiotemporal frequency, luminance and eccentricity. *Journal of Vision*, 24(4):5–5, 2024.

[58] Nianchen Deng, Zhenyi He, Jiannan Ye, Budmonde Duinkharjav, Praneeth Chakravarthula, Xubo Yang, and Qi Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3854–3864, 2022.

[59] Weikai Lin, Yu Feng, and Yuhao Zhu. Metasapiens: Real-time neural rendering with efficiency-aware pruning and accelerated foveated rendering. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, pages 669–682, 2025.

[60] Rachel McDonnell, Michéal Larkin, Benjamín Hernández, Isaac Rudomin, and Carol O'Sullivan. Eye-catching crowds: saliency based selective variation. *ACM Transactions on Graphics (TOG)*, 28(3):1–10, 2009.

[61] Youssef A Mejjati, Celso F Gomez, Kwang In Kim, Eli Shechtman, and Zoya Bylinskii. Look here! a parametric learning based approach to redirect visual attention. In *European Conference on Computer Vision*, pages 343–361, 2020.

[62] Kfir Aberman, Junfeng He, Yossi Gandelsman, Inbar Mosseri, David E Jacobs, Kai Kohlhoff, Yael Pritch, and Michael Rubinstein. Deep saliency prior for reducing visual distraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19851–19860, 2022.

[63] S Mahdi H Miangoleh, Zoya Bylinskii, Eric Kee, Eli Shechtman, and Yağiz Aksoy. Realistic saliency guided image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 186–194, 2023.

[64] Lai Jiang, Mai Xu, Xiaofei Wang, and Leonid Sigal. Saliency-guided image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16509–16518, 2021.

[65] Takanori Chihara and Akihiko Seo. Evaluation of physical workload affected by mass and center of mass of head-mounted display. *Applied Ergonomics*, 68:204–212, 2018.

[66] Marcus Thuresson, Jan Linder, Karin Harms-Ringdahl, et al. Neck muscle activity in helicopter pilots: effect of position and helmet-mounted equipment. *Aviation, Space, and Environmental Medicine*, 74(5):527–532, 2003.

[67] Peter Le, Charles A Weisenbach, Emily HL Mills, Lanie Monforton, and Micah J Kinney. Exploring the interaction between head-supported mass, posture, and visual stress on neck muscle activation. *Human Factors*, page 00187208211019154, 2021.

[68] Sheila A Dugan and Walter R Frontera. Muscle fatigue and muscle injury. *Physical Medicine and Rehabilitation Clinics of North America*, 11(2):385–403, 2000.

[69] Wanwan Li, Biao Xie, Yongqi Zhang, Walter Meiss, Haikun Huang, and Lap-Fai Yu. Exertion-aware path generation. *ACM Transactions on Graphics*, 39(4):115, 2020.

[70] Ziv Epstein, Aaron Hertzmann, Investigators of Human Creativity, Memo Akten, Hany Farid, Jessica Fjeld, Morgan R Frank, Matthew Groh, Laura Herman, Neil Leach, et al. Art and the science of generative ai. *Science*, 380(6650):1110–1111, 2023.

[71] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit Bermano, Eric Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. In *Computer Graphics Forum*, volume 43, page e15063. Wiley Online Library, 2024.

[72] Karthik Vaidyanathan, Marco Salvi, Bartlomiej Wronski, Tomas Akenine-Moller, Pontus Ebelin, and Aaron Lefohn. Random-access neural compression of material textures. *ACM Transactions on Graphics (TOG)*, 42(4):1–25, 2023.

[73] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8628–8638, 2021.

[74] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. Relu fields: The little non-linearity that could. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–9, 2022.

[75] Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. Neurbf: A neural fields representation with adaptive radial basis functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4182–4194, 2023.

[76] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023.

[77] Wang Yifan, Felice Serena, Shihao Wu, Cengiz Öztireli, and Olga Sorkine-Hornung. Differentiable surface splatting for point-based geometry processing. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019.

[78] Georgios Kopanas, Julien Philip, Thomas Leimkühler, and George Drettakis. Point-based neural rendering with per-view optimization. In *Computer Graphics Forum*, volume 40, pages 29–43. Wiley Online Library, 2021.

[79] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. IEEE, 2003.

[80] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.

[81] Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D Fairchild. Flip: A difference evaluator for alternating images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2):15:1–15:23, 2020.

[82] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, et al. gsplat: An open-source library for gaussian splatting. *arXiv preprint arXiv:2409.06765*, 2024.

[83] Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, pages 327–345. Springer, 2024.

[84] George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020.

[85] Yueyu Hu, Wenhan Yang, Zhan Ma, and Jiaying Liu. Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4194–4211, 2021.

[86] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[87] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR, 2023.

[88] Yanwei Pang, Jiale Cao, Yazhao Li, Jin Xie, Hanqing Sun, and Jinfeng Gong. Tju-dhd: A diverse high-resolution dataset for object detection. *IEEE Transactions on Image Processing*, 30:207–219, 2020.

[89] Rick A Vander Kam, Ping Wah Wong, and Robert M Gray. Jpeg-compliant perceptual coding for a grayscale image printing pipeline. *IEEE Transactions on Image Processing*, 8(1):1–14, 1999.

[90] Jean J Lorre and Alan R Gillespie. Artifacts in digital images. In *Applications of Digital Image Processing to Astronomy*, volume 264, pages 123–135. SPIE, 1980.

[91] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2758–2767, 2020.

[92] Yuanhao Cai, Yixun Liang, Jiahao Wang, Angtian Wang, Yulun Zhang, Xiaokang Yang, Zongwei Zhou, and Alan Yuille. Radiative gaussian splatting for efficient x-ray novel view synthesis. In *European Conference on Computer Vision*, pages 283–299. Springer, 2024.

[93] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Lau, Feng Gao, Yin Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–1, 2024.

[94] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21634–21643, 2024.

[95] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19595–19604, 2024.

[96] Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J Black, Bernhard Thomaszewski, Christina Tsalicoglou, and Otmar Hilliges. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. In *International Conference on 3D Vision 2025*, 2025.

[97] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19447–19456, 2024.

[98] Zhiwen Yan, Weng Fei Low, Yu Chen, and Gim Hee Lee. Multi-scale 3d gaussian splatting for anti-aliased rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20923–20931, 2024.

[99] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024.

[100] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024.

[101] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in Neural Information Processing Systems*, 37:140138–140158, 2024.

[102] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024.

[103] Michael Niemeyer, Fabian Manhardt, Marie-Julie Rakotosaona, Michael Oechsle, Daniel Duckworth, Rama Gosula, Keisuke Tateno, John Bates, Dominik Kaeser, and Federico Tombari. Radsplat: Radiance field-informed gaussian splatting for robust real-time rendering with 900+ fps. In *International Conference on 3D Vision 2025*, 2025.

[104] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, pages 165–181. Springer, 2024.

[105] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[106] Nicholas Milef, Dario Seyb, Todd Keeler, Thu Nguyen-Phuoc, A Božič, Sushant Kondguli, and Carl Marshall. Learning fast 3d gaussian splatting rendering using continuous level of detail. In *Computer Graphics Forum*, page e70069. Wiley Online Library, 2025.

[107] Jialin Wang, Rongkai Shi, Wenxuan Zheng, Weijie Xie, Dominic Kao, and Hai-Ning Liang. Effect of frame rate on user experience, performance, and simulator sickness in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2478–2488, 2023.

[108] Jialin Wang, Rongkai Shi, Zehui Xiao, Xueying Qin, and Hai-Ning Liang. Effect of render resolution on gameplay experience, performance, and simulator sickness in virtual reality games. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–15, 2022.

[109] Yang Zhao, Dave Lindberg, Bruce Cleary, Olivier Mercier, Ryan Mcclelland, Eric Penner, Yu-Jen Lin, Julia Majors, and Douglas Lanman. Retinal-resolution varifocal vr. In *ACM SIGGRAPH 2023 Emerging Technologies*, pages 1–3. 2023.

[110] Kenneth Chen, Thomas Wan, Nathan Matsuda, Ajit Ninan, Alexandre Chapiro, and Qi Sun. Pea-pods: Perceptual evaluation of algorithms for power optimization in xr displays. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024.

[111] Maria Perez-Ortiz, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafał K Mantiuk. From pairwise comparisons and rating to a unified quality scale. *IEEE Transactions on Image Processing*, 29:1139–1151, 2019.

[112] H. Liang, T. Wu, P. Hanji, F. Banterle, H. Gao, R. Mantiuk, and C. Öztireli. Perceptual quality assessment of nerf and neural view synthesis methods for front-facing views. *Computer Graphics Forum*, 43(2):e15036, 2024. doi: https://doi.org/10.1111/cgf.15036. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.15036`.

[113] Rafał K. Mantiuk, Anna Tomaszewska, and Radosław Mantiuk. Comparison of four subjective methods for image quality assessment. *Computer Graphics Forum*, 31(8):2478–2491, 2012. doi: https://doi.org/10.1111/j.1467-8659.2012.03188.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.03188.x`.

[114] Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafał Mantiuk. Active sampling for pairwise comparisons via approximate message passing and information gain maximization. In *2020 IEEE International Conference on Pattern Recognition (ICPR)*, January 2021.

[115] Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023.

[116] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.

[117] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[118] Xufang Pang, Ying Cao, Rynson WH Lau, and Antoni B Chan. Directing user attention via visual flow on web designs. *ACM Transactions on Graphics*, 35(6):1–11, 2016.

[119] Muhammad Helmi Abu Bakar, Mohd Asyiek Mat Desa, and Muhizam Mustafa. Attributes for image content that attract consumers' attention to advertisements. *Procedia-Social and Behavioral Sciences*, 195:309–314, 2015.

[120] Arthur P Shimamura, Brendan I Cohn-Sheehy, Brianna L Pogue, and Thomas A Shimamura. How attention is driven by film edits: A multimodal experience. *Psychology of Aesthetics, Creativity, and the Arts*, 9(4):417, 2015.

[121] Matthias Kümmerer, Lucas Theis, and Matthias Bethge. Deep gaze i: Boosting saliency prediction with feature maps trained on imagenet. In *International Conference on Learning Representations*, 2015.

[122] Richard Droste, Jianbo Jiao, and J Alison Noble. Unified image and video saliency modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 419–435. Springer, 2020.

[123] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4903, 2018.

[124] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[125] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015.

[126] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023.

[127] Kyle Min and Jason J Corso. Tased-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2394–2403, 2019.

[128] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[129] Matthias Kummerer, Thomas SA Wallis, and Matthias Bethge. Saliency benchmarking made easy: Separating models, maps and metrics. In *Proceedings of the European Conference on Computer Vision*, pages 770–787, 2018.

[130] Robert J Peters, Asha Iyer, Laurent Itti, and Christof Koch. Components of bottom-up gaze allocation in natural images. *Vision Research*, 45(18):2397–2416, 2005.

[131] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[132] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015.

[133] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, pages 8748–8763, 2021.

[134] Charles W Eriksen and Yei-yu Yeh. Allocation of attention in the visual field. *Journal of Experimental Psychology: Human Perception and Performance*, 11(5):583, 1985.

[135] Byron Reeves, Annie Lang, Eun Young Kim, and Deborah Tatar. The effects of screen size and message content on attention and arousal. *Media Psychology*, 1(1):49–67, 1999.

[136] Kaitlin M Gallagher, Laura Cameron, Diana De Carvalho, and Madison Boule. Does using multiple computer monitors for office tasks affect user experience? a systematic review. *Human Factors*, 63(3):433–449, 2021.

[137] Yunxiang Zhang, Kenneth Chen, and Qi Sun. Toward optimized vr/ar ergonomics: Modeling and predicting user neck muscle contraction. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.

[138] Lee W Gregg and WJ Brogden. The relation between reaction time and the duration of the auditory stimulus. *Journal of Comparative and Physiological Psychology*, 43(5):389, 1950.

[139] Drew Fudenberg, Whitney Newey, Philipp Strack, and Tomasz Strzalecki. Testing the drift-diffusion model. *Proceedings of the National Academy of Sciences*, 117(52):33141–33148, 2020.

[140] Maxwell Shinn, Norman H Lam, and John D Murray. A flexible framework for simulating and fitting generalized drift-diffusion models. *ELife*, 9:e56938, 2020.

[141] Budmonde Duinkharjav, Praneeth Chakravarthula, Rachel Brown, Anjul Patney, and Qi Sun. Image features influence reaction time: a learned probabilistic perceptual model for saccade latency. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022.

[142] Lore Thaler, Alexander C Schütz, Melvyn A Goodale, and Karl R Gegenfurtner. What is the best fixation target? the effect of target shape on stability of fixational eye movements. *Vision research*, 76:31–42, 2013.

[143] Edward J Engelken, Kenneth W Stevens, and John D Enderle. Relationships between manual reaction time and saccade latency in response to visual and auditory stimuli. Technical report, SCHOOL OF AEROSPACE MEDICINE BROOKS AFB TX, 1991.

[144] Robert J Kosinski. A literature review on reaction time. *Clemson University*, 10(1):337–344, 2008.

[145] Ralf Engbert and Konstantin Mergenthaler. Microsaccades are triggered by low retinal image slip. *Proceedings of the National Academy of Sciences*, 103(18):7192–7197, 2006.

[146] Franklin M Henry. Stimulus complexity, movement complexity, age, and sex in relation to reaction latency and speed in limb movements. *Research Quarterly. American Association for Health, Physical Education and Recreation*, 32(3):353–366, 1961.

[147] Florian Schmiedek, Klaus Oberauer, Oliver Wilhelm, Heinz-Martin Süß, and Werner W Wittmann. Individual differences in components of reaction time distributions and their relations to working memory and intelligence. *Journal of experimental psychology: General*, 136(3):414, 2007.

[148] Robert J Van Beers. The sources of variability in saccadic eye movements. *Journal of Neuroscience*, 27(33):8757–8770, 2007.

[149] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[150] Roger Ratcliff and Gail McKoon. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20(4):873–922, 2008.

[151] Michael N Shadlen and Roozbeh Kiani. Decision making as a window on cognition. *Neuron*, 80(3):791–806, 2013.

[152] Ian Krajbich and Antonio Rangel. Multialternative drift-diffusion model predicts the relationship between visual fixations and choice in value-based decisions. *Proceedings of the National Academy of Sciences*, 108(33):13852–13857, 2011.

[153] Ian Krajbich, Dingchao Lu, Colin Camerer, and Antonio Rangel. The attentional drift-diffusion model extends to simple purchasing decisions. *Frontiers in psychology*, 3:193, 2012.

[154] Sebastian Bitzer, Hame Park, Felix Blankenburg, and Stefan J Kiebel. Perceptual decision making: drift-diffusion model is equivalent to a bayesian model. *Frontiers in human neuroscience*, 8:102, 2014.

[155] Milica Milosavljevic, Jonathan Malmaud, Alexander Huth, Christof Koch, and Antonio Rangel. The drift diffusion model can account for the accuracy and reaction time of value-based choices under high and low time pressure. *Judgment and Decision making*, 5(6): 437–449, 2010.

[156] Nitzan Shahar, Tobias U Hauser, Michael Moutoussis, Rani Moran, Mehdi Keramati, Nspn Consortium, and Raymond J Dolan. Improving the reliability of model-based decision-making estimates in the two-stage decision task with reaction-times and drift-diffusion modeling. *PLoS computational biology*, 15(2):e1006803, 2019.

[157] J Leroy Folks and Raj S Chhikara. The inverse gaussian distribution and its statistical application—a review. *Journal of the Royal Statistical Society: Series B (Methodological)*, 40 (3):263–275, 1978.

[158] Evan M Palmer, Todd S Horowitz, Antonio Torralba, and Jeremy M Wolfe. What are the shapes of response time distributions in visual search? *Journal of experimental psychology: human perception and performance*, 37(1):58, 2011.

[159] Baj AJ Reddi, Kaleab N Asrress, and Roger HS Carpenter. Accuracy, information, and response time in a saccadic decision task. *Journal of neurophysiology*, 90(5):3538–3546, 2003.

[160] Hans Colonius and Adele Diederich. Multisensory interaction in saccadic reaction time: a time-window-of-integration model. *Journal of cognitive neuroscience*, 16(6):1000–1009, 2004.

[161] Brandon M Turner, Juan Gao, Scott Koenig, Dylan Palfy, and James L. McClelland. The dynamics of multimodal integration: The averaging diffusion model. *Psychonomic bulletin & review*, 24:1819–1843, 2017.

[162] Wei Ji Ma and Alexandre Pouget. Linking neurons to behavior in multisensory perception: A computational review. *Brain research*, 1242:4–12, 2008.

[163] Derek H Arnold, Kirstie Petrie, Cailem Murray, and Alan Johnston. Suboptimal human multisensory cue combination. *Scientific Reports*, 9(1):1–11, 2019.

[164] Peter W Battaglia, Robert A Jacobs, and Richard N Aslin. Bayesian integration of visual and auditory signals for spatial localization. *Josa a*, 20(7):1391–1397, 2003.

[165] Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

[166] Pablo Gomez, Roger Ratcliff, and Manuel Perea. A model of the go/no-go task. *Journal of Experimental Psychology: General*, 136(3):389, 2007.

[167] Brian A Nosek and Mahzarin R Banaji. The go/no-go association task. *Social cognition*, 19 (6):625–666, 2001.

[168] Fang-Yi Chao, Cagri Ozcinar, Chen Wang, Emin Zerman, Lu Zhang, Wassim Hamidouche, Olivier Deforges, and Aljosa Smolic. Audio-visual perception of omnidirectional video for virtual reality applications. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6, 2020. doi: 10.1109/ICMEW46912.2020.9105956.

[169] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[170] Raul Altosaar, Adam Tindale, and Judith Doyle. Physically colliding with music: Full-body interactions with an audio-only virtual reality interface. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 553–557, 2019.

[171] Luke Hespanhol, Oliver Bown, Jingwen Cao, and Martin Tomitsch. Evaluating the effectiveness of audio-visual cues in immersive user interfaces. In *Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration*, pages 569–572, 2013.

[172] Kristine Song, Amit Chakraborty, Matthew Dawson, Adam Dugan, Brian Adkins, and Christopher Doty. Does the podcast video playback speed affect comprehension for novel curriculum delivery? a randomized trial. *Western Journal of Emergency Medicine*, 19(1):101, 2018.

[173] J Wierzbicki, P Małecki, and J Wiciak. Localization of the sound source with the use of the first-order ambisconic microphone. *Acta Physica Polonica A*, 123(6):1114–1117, 2013.

[174] Vanessa Tabry, Robert J Zatorre, and Patrice Voss. The influence of vision on sound localization abilities in both the horizontal and vertical planes. *Frontiers in psychology*, 4: 932, 2013.

[175] James A Roberts, Guy Wallis, and Michael Breakspear. Fixational eye movements during viewing of dynamic natural scenes. *Frontiers in psychology*, 4:797, 2013.

[176] Matteo Lisi, Joshua A Solomon, and Michael J Morgan. Gain control of saccadic eye movements is probabilistic. *Proceedings of the National Academy of Sciences*, 116(32):16137–16142, 2019.

[177] Jane Farrell. The effect of increasing music volume on reaction time. *The Journal of Science and Medicine*, 2021.

[178] Edwardo Arata Y Murakami. Reaction time and emg measurement applied to human control modeling. *Measurement*, 43(5):675–683, 2010.

[179] Eyal Ophir, Clifford Nass, and Anthony D Wagner. Cognitive control in media multitaskers. *Proceedings of the National Academy of Sciences*, 106(37):15583–15587, 2009.

[180] Ming C Lin and Miguel Otaduy. *Haptic rendering: foundations, algorithms, and applications*. CRC Press, 2008.

[181] Yiyue Luo, Chao Liu, Young Joong Lee, Joseph DelPreto, Kui Wu, Michael Foshey, Daniela Rus, Tomás Palacios, Yunzhu Li, Antonio Torralba, et al. Adaptive tactile interaction transfer via digitally embroidered smart gloves. *Nature Communications*, 15(1):868, 2024.

[182] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[183] Yu Du, Yongkang Wong, Wenguang Jin, Wentao Wei, Yu Hu, Mohan S Kankanhalli, and Weidong Geng. Semi-supervised learning for surface emg-based gesture recognition. In *IJCAI*, pages 1624–1630, 2017.

[184] Vincent Becker, Pietro Oldrati, Liliana Barrios, and Gábor Sörös. Touchsense: classifying finger touches and measuring their force with an electromyography armband. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 1–8, 2018.

[185] He Mao, Peng Fang, and Guanglin Li. Simultaneous estimation of multi-finger forces by surface electromyography and accelerometry signals. *Biomedical Signal Processing and Control*, 70:103005, 2021.

[186] Jiaqi Xue and King Wai Chiu Lai. Dynamic gripping force estimation and reconstruction in emg-based human-machine interaction. *Biomedical Signal Processing and Control*, 80: 104216, 2023.

[187] Christian Von Hardenberg and François Bérard. Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on Perceptive user interfaces*, pages 1–8, 2001.

[188] W Melzer, E Rios, and MF Schneider. Time course of calcium release and removal in skeletal muscle fibers. *Biophysical journal*, 45(3):637–641, 1984.

[189] Christopher Spiewak, M Islam, A Zaman, and Mohammad Habibur Rahman. A comprehensive study on emg feature extraction and classifiers. *Open Access Journal of Biomedical Engineering and Biosciences*, 1(1):1–10, 2018.

[190] Reza Boostani and Mohammad Hassan Moradi. Evaluation of the forearm emg signal features for the control of a prosthetic hand. *Physiological measurement*, 24(2):309, 2003.

[191] James N Ingram, Konrad P Körding, Ian S Howard, and Daniel M Wolpert. The statistics of natural hand movements. *Experimental brain research*, 188(2):223–236, 2008.

[192] Feng Xu, Yang Zheng, and Xiaogang Hu. Real-time finger force prediction via parallel convolutional neural networks: a preliminary study. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 3126–3129. IEEE, 2020.

[193] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[194] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[195] Tu-Hoa Pham, Abderrahmane Kheddar, Ammar Qammaz, and Antonis A Argyros. Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2810–2819, 2015.

[196] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. Inferring forces and learning human utilities from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3823–3833, 2016.

[197] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A Argyros, and Abderrahmane Kheddar. Hand-object contact force estimation from markerless visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2883–2896, 2017.

[198] Wonjun Hwang and Soo-Chul Lim. Inferring interaction force from visual information without using physical force sensors. *Sensors*, 17(11):2455, 2017.

[199] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 224–233, 2020.

[200] Yu Sun, John M Hollerbach, and Stephen A Mascaro. Predicting fingertip forces by imaging coloration changes in the fingernail and surrounding skin. *IEEE Transactions on Biomedical Engineering*, 55(10):2363–2371, 2008.

[201] Thomas Grieve, Lucas Lincoln, Yu Sun, John M Hollerbach, and Stephen A Mascaro. 3d force prediction using fingernail imaging with automated calibration. In *2010 IEEE Haptics Symposium*, pages 113–120. IEEE, 2010.

[202] Thomas R Grieve, John M Hollerbach, and Stephen A Mascaro. 3-d fingertip touch force prediction using fingernail imaging with automated calibration. *IEEE Transactions on Robotics*, 31(5):1116–1129, 2015.

[203] Thomas R Grieve, John M Hollerbach, and Stephen A Mascaro. Optimizing fingernail imaging calibration for 3d force magnitude prediction. *IEEE transactions on haptics*, 9(1): 69–79, 2015.

[204] Navid Fallahinia and Stephen A Mascaro. Comparison of constrained and unconstrained human grasp forces using fingernail imaging and visual servoing. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2668–2674. IEEE, 2020.

[205] Navid Fallahinia and Stephen A Mascaro. Feasibility study of force measurement for multi-digit unconstrained grasping via fingernail imaging and visual servoing. *ASME Letters in Dynamic Systems and Control*, 1(2), 2021.

[206] Navid Fallahinia and Stephen Mascaro. Real-time tactile grasp force sensing using fingernail imaging via deep neural networks. *arXiv preprint arXiv:2109.15231*, 2021.

[207] Wang Dangxiao, Guo Yuan, Liu Shiyi, Yuru Zhang, Xu Weiliang, and Xiao Jing. Haptic display for virtual reality: progress and challenges. *Virtual Reality & Intelligent Hardware*, 1 (2):136–162, 2019.

[208] Elisabeth Baumgartner, Christiane B. Wiebel, and Karl R. Gegenfurtner. Visual and haptic representations of material properties. *Multisensory Research*, 26(5):429–455, 2013. ISSN 2213-4808(Electronic),2213-4794(Print). doi: 10.1163/22134808-00002429.

[209] Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2):109–118, 2007.

[210] Miles Macklin, Matthias Müller, and Nuttapong Chentanez. Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th International Conference on Motion in Games*, pages 49–54, 2016.

[211] Jose María Méndez and Lidia Martínez. Virtual method studio, 2021. URL `http://obi.virtualmethodstudio.com/index.html`.

[212] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (TOG)*, 39(4):87–1, 2020.

[213] David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 167–176, 2012.

[214] Lee Stearns, Uran Oh, Leah Findlater, and Jon E Froehlich. Touchcam: Realtime recognition of location-specific on-body gestures to support users with visual impairments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–23, 2018.

[215] Chris Harrison and Scott E Hudson. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 205–208, 2008.

[216] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, et al. Fingerping: Recognizing fine-grained hand poses using active acoustic on-body sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2018.

[217] Xuhai Xu, Haitian Shi, Xin Yi, WenJia Liu, Yukang Yan, Yuanchun Shi, Alex Mariakakis, Jennifer Mankoff, and Anind K Dey. Earbuddy: Enabling on-face interaction via wireless earbuds. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2020.

[218] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, 35(4):1–19, 2016.

[219] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 851–860, 2016.

[220] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 1472–1480. IEEE, 2015.

[221] Heba Abdelnasser, Khaled Harras, and Moustafa Youssef. A ubiquitous wifi-based fine-grained gesture recognition system. *IEEE Transactions on Mobile Computing*, 18(11):2474–2487, 2018.

[222] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[223] Thomas Waltemate, Irene Senna, Felix Hülsmann, Marieke Rohde, Stefan Kopp, Marc Ernst, and Mario Botsch. The impact of latency on perceptual judgments and motor performance in closed-loop interaction in virtual reality. In *Proceedings of the 22nd ACM conference on virtual reality software and technology*, pages 27–35, 2016.

[224] Rachel Albert, Anjul Patney, David Luebke, and Joohwan Kim. Latency requirements for foveated rendering in virtual reality. *ACM Transactions on Applied Perception (TAP)*, 14(4): 1–13, 2017.

[225] Ernst Kruijff, J Edward Swan, and Steven Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pages 3–12. IEEE, 2010.

[226] Tim Zindulka, Myroslav Bachynskyi, and Jörg Müller. Performance and experience of throwing in virtual reality. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–8, 2020.

[227] Yaffa Yeshurun, Marisa Carrasco, and Laurence T Maloney. Bias and sensitivity in two-interval forced choice procedures: Tests of the difference model. *Vision research*, 48(17): 1837–1851, 2008.

[228] Vinh T Nguyen and Tommy Dang. Setting up virtual reality and augmented reality learning environment in unity. In *2017 IEEE International symposium on mixed and augmented reality (ISMAR-Adjunct)*, pages 315–320. IEEE, 2017.

[229] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[230] James R Lewis. Measuring perceived usability: The csuq, sus, and umux. *International Journal of Human–Computer Interaction*, 34(12):1148–1156, 2018.

[231] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. User acceptance of information technology: Toward a unified view. *MIS quarterly*, pages 425–478, 2003.

[232] Bob G Witmer and Michael J Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.

[233] Thomas Schubert, Frank Friedmann, and Holger Regenbrecht. The experience of presence: Factor analytic insights. *Presence: Teleoperators & Virtual Environments*, 10(3):266–281, 2001.

[234] Kelsey A Forde, Wayne J Albert, Michael F Harrison, J Patrick Neary, James Croll, and Jack P Callaghan. Neck loads and posture exposure of helicopter pilots during simulated day and night flights. *International Journal of Industrial Ergonomics*, 41(2):128–135, 2011.

[235] Sai Akhil Penumudi, Veera Aneesh Kuppam, Jeong Ho Kim, and Jaejin Hwang. The effects of target location on musculoskeletal load, task performance, and subjective discomfort during virtual reality interactions. *Applied ergonomics*, 84:103010, 2020.

[236] Richard W Marklin Jr, Ashley M Toll, Eric H Bauman, John J Simmins, John F LaDisa Jr, and Robert Cooper. Do head-mounted augmented reality devices affect muscle activity and eye strain of utility workers who do procedural work? studies of operators and manhole workers. *Human factors*, 64(2):305–323, 2022.

[237] Alexis D Souchet, Domitile Lourdeaux, Alain Pagani, and Lisa Rebenitsch. A narrative review of immersive virtual reality's ergonomics and risks at the workplace: cybersickness, visual fatigue, muscular fatigue, acute stress, and mental overload. *Virtual Reality*, pages 1–32, 2022.

[238] Roberto Merletti and Philip J Parker. *Electromyography: physiology, engineering, and non-invasive applications*, volume 11. John Wiley & Sons, 2004.

[239] Eleanor Criswell. *Cram's introduction to surface electromyography*. Jones & Bartlett Publishers, 2010.

[240] Anita N Vasavada, Siping Li, and Scott L Delp. Influence of muscle morphometry and moment arms on the moment-generating capacity of human neck muscles. *Spine*, 23(4): 412–422, 1998.

[241] Nataniel Ruiz, Eunji Chong, and James M Rehg. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2074–2083, 2018.

[242] Gregory J Lehman and Stuart M McGill. The importance of normalization in the interpretation of surface electromyography: a proof of principle. *Journal of manipulative and physiological therapeutics*, 22(7):444–446, 1999.

[243] Carolyn M Sommerich, Sharon MB Joines, Veerle Hermans, and Samuel D Moon. Use of surface electromyography to estimate neck muscle activity. *Journal of Electromyography and kinesiology*, 10(6):377–398, 2000.

[244] Mamun Bin Ibne Reaz, M Sazzad Hussain, and Faisal Mohd-Yasin. Techniques of emg signal analysis: detection, processing, classification and applications. *Biological procedures online*, 8(1):11–35, 2006.

[245] Brian R Umberger, Karin GM Gerritsen, and Philip E Martin. A model of human muscle energy expenditure. *Computer methods in biomechanics and biomedical engineering*, 6(2): 99–111, 2003.

[246] Kohei Watanabe and Hiroshi Akima. Normalized emg to normalized torque relationship of vastus intermedius muscle during isometric knee extension. *European journal of applied physiology*, 106(5):665–673, 2009.

[247] Edward A Clancy, Lukai Liu, Pu Liu, and Daniel V Zandt Moyer. Identification of constant-posture emg–torque relationship about the elbow using nonlinear dynamic models. *IEEE Transactions on Biomedical Engineering*, 59(1):205–212, 2011.

[248] James Paquin and Geoffrey A Power. History dependence of the emg-torque relationship. *Journal of Electromyography and Kinesiology*, 41:109–115, 2018.

[249] Peter R Cavanagh and Paavo V Komi. Electromechanical delay in human skeletal muscle under concentric and eccentric contractions. *European journal of applied physiology and occupational physiology*, 42(3):159–163, 1979.

[250] Wolfgang H Zangemeister, Ashby Jones, and Lawrence Stark. Dynamics of head movement trajectories: main sequence relationship. *Experimental neurology*, 71(1):76–91, 1981.

[251] Eleana A Pinto, Dimitri ML van Ryckeghem, Ann Meulders, Diana M Torta, Amber Claus, and Stefaan Van Damme. Motor action changes pain perception: a sensory attenuation paradigm in the context of pain. *Pain*, 162(7):2060–2069, 2021.

[252] Bassem Farid, Paul Yielder, Michael Holmes, Heidi Haavik, and Bernadette A Murphy. Association of subclinical neck pain with altered multisensory integration at baseline and 4-week follow-up relative to asymptomatic controls. *Journal of manipulative and physiological therapeutics*, 41(2):81–91, 2018.