Human-in-the-Loop Design Optimization

Yi-Chi Liao
Human-in-the-Loop Design Optimization

Yi-Chi Liao

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall T2 of the school on 15 December 2023 at 14.

Aalto University
School of Electrical Engineering
Information and Communications Engineering
Computational Behavior Lab
Supervising professor
Professor Antti Oulasvirta, Aalto University, Finland

Thesis advisor
Professor Antti Oulasvirta, Aalto University, Finland

Preliminary examiners
Dr. Yuki Koyama, National Institute of Advanced Industrial Science and Technology, Japan.
Assistant Professor Seongkook Heo, University of Virginia, U.S.A.

Opponent
Associate Professor Pedro Lopes, University of Chicago, U.S.A.
Abstract

This dissertation presents novel computational methods and investigations to enable human-in-the-loop optimization (HILO) for a wider range of realistic applications, allowing designers to efficiently explore the design space of practical problems. Designing effective interaction techniques requires careful consideration of various parameters, that significantly impact user experience and performance. However, optimizing these parameters can be challenging due to the large, multi-dimensional design space, the unclear relationship between parameter settings and user performance, and the complexity of balancing multiple design objectives.

Traditionally, designers perform manual optimization via iterative design processes, which can be time-consuming and effortful, and do not guarantee the best outcome. HILO emerged as a more principled solution for design optimization, using a computational optimizer to intelligently select the next design instance for user testing. Despite some examples of HILO in the human-computer interaction (HCI) field, its application scope is limited to a single objective, for a single user, and for graphical user interfaces. How to extend HILO for multi-objective problems, optimizing for a population, and supporting physical interfaces has remained unclear. Furthermore, conducting HILO does not eliminate the costs arising from human involvement, and practitioners have been reluctant to embrace a technique whose positive and negative qualities are not fully understood.

This dissertation presents a set of computational methods and investigations that address these challenges. Pareto-frontier learning is utilized to handle multi-objective design tasks, and I introduce novel extensions for practical solutions of group-level Bayesian optimization. To reduce the effort and time in prototyping, I propose using physical emulation to render physical design instances, enabling HILO to be applied to the design of physical interactions. The dissertation presents user experiments and a design workshop conducted to enrich the understanding of Bayesian optimization-supported design processes' strengths and limitations. Finally, in light of the resource-intensive nature of user studies, a simulation-based optimization framework is proposed whereby artificial users evaluate design instances.

With the ultimate goal of expanding HILO's utility in realistic and general design tasks, this dissertation opens new directions for future HILO research. One important path for exploration involves more advanced optimization techniques, such as methods that enable greater efficiency and support a high-dimensional design space. The project also spotlights the value of investigating better human-machine collaboration mechanisms in design optimization such that the designers can steer the optimization as required or fine-tune the suggestions proposed by the optimizer. Lastly, simulation-based optimization methods require further validation, and developing human-like models will be a crucial next step.

Keywords Human-in-the-loop optimization, Bayesian optimization, human-computer interaction, computational interaction, machine learning

Author
Yi-Chi Liao

Name of the doctoral thesis
Human-in-the-Loop Design Optimization

Publisher School of Electrical Engineering

Unit Information and Communications Engineering

Series Aalto University publication series DOCTORAL THESES 219/2023

Field of research Interactive Systems

Permission submitted 4 May 2023 Date of the defence 15 December 2023

Monograph Article thesis Essay thesis

Abstract

Permission for public defence granted (date) 29 June 2023 Language English


ISSN (printed) 1799-4934 ISSN (pdf) 1799-4942

Location of publisher Helsinki Location of printing Helsinki Year 2023

Right before embarking on my Ph.D. journey in Finland, I attended ACM CHI 2018 in Montreal, followed by a brief stay in Toronto before catching the connecting flight to Finland. What I have never shared with my friends is that, during this layover, I was completely overwhelmed by the fear of the uncertainty that lay ahead of me. I had no idea what to expect to live in this distant country, Finland, or what the Ph.D. years would hold. As I sat at the gate in Toronto Airport, I even had the sudden idea of returning to Taiwan and abandoning my pursuit of a Ph.D.

Looking back, I’m so grateful that I summoned just enough courage to board that flight and come to Finland. It turned out to be an incredible journey, one that I would not trade for anything. What made it truly amazing were the exceptional people who supported me, guided me, and shared their lives with me. I want to take this opportunity to express my deepest gratitude to these extraordinary friends and colleagues.

First and foremost, I would like to extend my special thanks to my supervisor, Prof. Antti Oulasvirta, for his guidance and support over the entire course of my doctoral studies. Antti provided me with the space and encouragement to explore my academic passions. He taught me the art of thinking and the fun of tackling challenging and complex research problems. It is no exaggeration to say that without his supervision and guidance, I would not have become the person I am today. Beyond research projects, Antti is an excellent friend, an exceptional group leader, a great thinker, and an ambitious mind. He will always be a role model for me in my pursuit as a researcher or, perhaps someday, as a professor.

I would like to thank my pre-examiners, Dr. Yuki Koyama and Prof. Seongkook Heo. I feel truly privileged to have received their insightful and constructive feedback on my dissertation, which significantly helped me in enhancing the overall quality of the work.

I am also greatly honored to have Prof. Pedro Lopes as my opponent. Prof. Lopes might not be aware of this: when I initially started my Ph.D. journey, I had the privilege of attending his Ph.D. defense at UIST 2018.
That experience was truly inspiring and motivating. Now, 5 years later, having Prof. Lopes as my opponent in this final stage of my studies brings a beautiful closure to my Ph.D. journey.

I genuinely thank every member of the User Interfaces Group / Computational Behavior Lab for their friendships and support throughout the years. I want to thank my dear friend, Kashyap, who has supported me in every aspect — academically, emotionally, and far beyond. Without him, I would not have had the chance to do an internship at Meta, become involved in the CHI organization, or enjoy countless other wonderful moments. I must thank Sunjun, who not only endured my constant pestering but also guided me step-by-step on the path to becoming a researcher. I can not thank him more for his patience, kindness, and his unlimited knowledge of buttons and HCI. I would like to thank Aleks and Markku, who welcomed me with a warm embrace when I first arrived in Finland, making me feel truly at home. My wholehearted gratitude goes to Aurélien, who I am proud to consider almost like an alternative family to me. I will never forget the game nights that we shared, and I am grateful for having you during my challenging time. I want to thank Luis for not only always giving me great advice but also for the joyful moments we played ping pong together.

Further, I would like to express my deep gratitude to Aini, Lena, Joongi, Danqing, Suyog, and Yue with whom I shared the office for an extended period. You hold an irreplaceable place in my memories of my time in the group. I couldn’t have asked for better companions. I also need to thank the former Ph.D. colleagues, Anna, Janin, and Morteza, for being outstanding examples for me to follow. I am thankful for having the opportunity to be a colleague of many talented minds: Aida, Jussi, Niraj, Carlos, Michael, Camille, Thomas (van Gemert), Thomas (Grabot), Ai, Yunfei, Christoph, Kristian, etc. I will always cherish the time shared with you.

I could not accomplish any of the publications without my fantastic co-authors. I am fortunate to have had the opportunity to collaborate with Dr. John Dudley on many papers. John is a warm and incredibly dependable person, and his tireless guidance and proofreading of my text have always served as a beacon of light that illuminated my academic journey. I am privileged to have collaborated with Prof. Per Ola Kristensson, Prof. Andrew Howes, and Prof. Liwei Chan. I am glad to receive their wisdom not just in research projects but also in making career decisions. I need to thank Prof. Byungjoo Lee, who guided me in the button project and stands as a role model for me to follow.

I heartily thank Dr. Aditya Acharya and Antti Keurulainen for their persistent support in the affordance project (especially, during the intense deadline week), and George and Chun-Lien, who stood by me through the peaks and valleys in the Bayesian optimization papers. I sincerely thank
Hee-Seung for involving me in the target inference project, being a great friend, and introducing me to the world of Nintendo Switch. I would like to thank Aleksi Ikkala for generously sharing knowledge on biomechanical models with me, and I sincerely look forward to future collaboration.

I had a fantastic internship at Meta Reality Labs, which boosted my growth as a researcher. I would like to thank Dr. Aakar Gupta for giving me this opportunity and for providing support in every detail. I want to thank Dr. Ruta Desai for her strong guidance in the technical aspects of the project. Ruta always asked the right (i.e., most difficult) questions, which I learned so much from. Big thanks to Pierce for aiding with implementations, and Krista for conducting the studies. My days in Seattle and Redmond were filled with memorable moments, largely thanks to the wonderful people, e.g., Sebastian, Naveen, Joao, David, Rishi, Matthias, Tanya, Ting, and many others. These memories are treasured forever.

I must thank all the incredible staff at Aalto University for their assistance and support throughout my Ph.D. studies. Their warmth and efficiency never ceased to amaze me. Special appreciation goes to the E-support team, the HR team, Essi, Sanna, and the Doctoral Studies team.

I want to thank my parents, who encouraged me to be curious and unique when I was a kid, which ultimately led me here. I am grateful for their trust and support in me in making every decision. My deepest thanks go to my grandma, whose kindness and patience have influenced and benefited my whole life. Being the first in my family to venture abroad, I extend my utmost thanks to every member of my family. Knowing that I have a loving family in Taiwan, a place I can always return to, provides me with a sense of security that allows me to continue exploring the world.

Last but certainly not least, I want to thank my amazing partner, Chieh-Ling. Our journey from Taiwan to Finland and beyond has been filled with both challenges and happy memories. Thank you for always being there for me, listening to me, uplifting me when I am down, helping me to find myself when I am lost, and taking care of me in every aspect of my life. Your unconditional love is the strongest driving force behind my exploration of the universe. I can not wait to create more memories with you in the next chapter of our life together.

Helsinki, November 21, 2023,

Yi-Chi Liao
1. Introduction
   1.1 The Design-Optimization Problem
   1.1.1 Design Parameters and Design Space
   1.1.2 Objective Functions
   1.1.3 Interactions as Black-Box Functions
   1.1.4 Challenges of Design Optimization
   1.2 Human-in-the-Loop Optimization
   1.2.1 Bayesian Optimization
   1.2.2 Research Questions
   1.3 Research Objectives and Methods
   1.3.1 Method #1: Optimization
   1.3.2 Method #2: Emulation, Prototyping, and Modeling
   1.3.3 Method #3: Simulation, User Modeling, and Theory
   1.3.4 Method #4: User Research
   1.4 Contributions
   1.5 The structure of the Dissertation

2. Background
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Design Processes and Empirical Research</td>
<td>41</td>
</tr>
<tr>
<td>2.1.1 Design Thinking’s Double-Diamond Model</td>
<td>42</td>
</tr>
<tr>
<td>2.1.2 User-Centered Design</td>
<td>42</td>
</tr>
<tr>
<td>2.1.3 Empirical Methods for Design Evaluation</td>
<td>43</td>
</tr>
<tr>
<td>2.2 Engineering Design Optimization</td>
<td>45</td>
</tr>
<tr>
<td>2.3 Human-in-the-Loop Optimization</td>
<td>46</td>
</tr>
<tr>
<td>2.4 Bayesian Optimization</td>
<td>47</td>
</tr>
<tr>
<td>2.4.1 Mathematical Properties of Bayesian Optimization</td>
<td>48</td>
</tr>
<tr>
<td>2.4.2 The Gaussian Process</td>
<td>48</td>
</tr>
<tr>
<td>2.4.3 The Acquisition Function</td>
<td>49</td>
</tr>
<tr>
<td>2.5 The State of the Art in Summary</td>
<td>50</td>
</tr>
<tr>
<td>3. Related Work: HILO in HCI</td>
<td>53</td>
</tr>
<tr>
<td>3.1 Human-in-the-Loop Bayesian Optimization in HCI</td>
<td>53</td>
</tr>
<tr>
<td>3.2 Online Optimization Systems in HCI</td>
<td>54</td>
</tr>
<tr>
<td>3.3 Preexisting Optimization Tools</td>
<td>56</td>
</tr>
<tr>
<td>4. From a Single Objective to Multiple Objectives</td>
<td>59</td>
</tr>
<tr>
<td>4.1 Multi-Objective Optimization</td>
<td>60</td>
</tr>
<tr>
<td>4.1.1 Pareto-Front Learning</td>
<td>62</td>
</tr>
<tr>
<td>4.2 Designing with Multi-Objective HILO</td>
<td>62</td>
</tr>
<tr>
<td>4.2.1 Interaction: Tactile Icons</td>
<td>63</td>
</tr>
<tr>
<td>4.2.2 The Workshop</td>
<td>65</td>
</tr>
<tr>
<td>4.2.3 Results</td>
<td>67</td>
</tr>
<tr>
<td>4.3 Investigating Performance in Multi-Objective HILO Conditions</td>
<td>70</td>
</tr>
<tr>
<td>4.3.1 3D Touch Interaction</td>
<td>71</td>
</tr>
<tr>
<td>4.3.2 The User Study</td>
<td>73</td>
</tr>
<tr>
<td>4.3.3 Results</td>
<td>74</td>
</tr>
<tr>
<td>4.4 The Work in Summary</td>
<td>77</td>
</tr>
<tr>
<td>5. From Individual to Population</td>
<td>79</td>
</tr>
<tr>
<td>5.1 Group-Level Optimization</td>
<td>80</td>
</tr>
<tr>
<td>5.1.1 Procedure of Group-Level Optimization</td>
<td>80</td>
</tr>
<tr>
<td>5.1.2 The Global GP: Deriving Group-Level Optimal Designs</td>
<td>81</td>
</tr>
<tr>
<td>5.1.3 Warm-Start GP: A Rapidly Adapting Surrogate Model</td>
<td>83</td>
</tr>
<tr>
<td>5.2 Evaluating Group-Level Optimization via Simulations</td>
<td>84</td>
</tr>
<tr>
<td>5.2.1 Test Functions</td>
<td>84</td>
</tr>
<tr>
<td>5.2.2 Setting up Simulations</td>
<td>85</td>
</tr>
<tr>
<td>5.2.3 Simulation Results</td>
<td>86</td>
</tr>
<tr>
<td>5.3 Designing with Global GP</td>
<td>86</td>
</tr>
<tr>
<td>5.3.1 Group-Level Optimized 3D Touch</td>
<td>87</td>
</tr>
<tr>
<td>5.3.2 The User Study</td>
<td>87</td>
</tr>
<tr>
<td>5.3.3 Results</td>
<td>89</td>
</tr>
</tbody>
</table>
### 5.4 Designing with the Warm-Start GP

- **5.4.1 The Fast-Adaptation Touch-Button**
- **5.4.2 The User Study**
- **5.4.3 Results**

### 5.5 The Work in Summary

### 6. From Physical Prototyping to Emulation

- **6.1 Background**
- **6.2 Emulation: The Case of a Push-Button Emulation Pipeline**
  - **6.2.1 FDVV Modeling**
  - **6.2.2 The Emulation Pipeline**
  - **6.2.3 Evaluation of the Emulation**
- **6.3 An Example of Optimization: HILO for Button Design**
  - **6.3.1 The Design Parameters and Objective Function**
  - **6.3.2 The Study Setting**
  - **6.3.3 Results**
- **6.4 Summary and Discussion**

### 7. From the Real World toward Simulation

- **7.1 A Simulation-Based HILO Framework**
  - **7.1.1 Background and Related Work**
  - **7.1.2 The Simulation-Based Optimization Framework**
  - **7.1.3 Physics Simulation**
  - **7.1.4 Reinforcement-Learning-Based User Models**
  - **7.1.5 Reinforcement Learning for Understanding of Affordance**
- **7.2 A Preliminary Example Case: 3D Touch Optimization**
  - **7.2.1 Configuration for the 3D Touch Interaction and Optimization**
  - **7.2.2 Agent Settings**
  - **7.2.3 Results**
- **7.3 The Work in Summary**

### 8. Discussion and Conclusion

- **8.1 The Findings Overall and Their Implications**
- **8.2 Limitations and Future Work**
  - **8.2.1 Advanced Optimization Methods**
  - **8.2.2 Making HILO More Usable for Designers**
  - **8.2.3 Work toward More Realistic Simulation and Emulation**
- **8.3 Conclusion**

### References

### Errata
This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.


Author’s Contribution

Publication I: “Interaction Design With Multi-objective Bayesian Optimization”

This paper investigated supporting design exploration with multi-objective Bayesian optimization. I planned and conducted a design workshop and analyzed the results, which formed the basis of this paper. I also developed the optimization with George B. Mo, and I implemented a smartwatch prototype for the workshop. The research ideation process was a team effort involving all co-authors. The writing process was also a collaborative effort, with Dr. John J. Dudley leading the final submission process. Prof. Per Ola Kristensson and Prof. Antti Oulasvirta provided support in the ideation and writing throughout.

Publication II: “Investigating Positive and Negative Qualities of Human-in-the-Loop Optimization for Designing Interaction Techniques”

This research project aimed to investigate the effectiveness of human-in-the-loop Bayesian optimization in comparison to the designer-led approach, under the leadership of Prof. Liwei Chan. I participated in the research ideation phase alongside other co-authors and designed the procedure of the user study. The user study was primarily conducted by Chun-Lien Cheng under Prof. Chan’s guidance. Additionally, I implemented Bayesian optimization with George B. Mo and the Unity application with Chun-Lien Cheng. My role in implementation was primarily bridging the optimization algorithm and target application. My contribution to the writing is mainly in Section 4, and Figures 6 and 7. I was also involved in writing and revising other sections throughout the submission process.
Publication III: “Practical Approaches to Group-Level Multi-Objective Bayesian Optimization in Interaction Technique Design”

In this paper, we proposed two practical approaches for achieving group-level multi-objective Bayesian optimization. I led the project ideation and application development, working closely with my co-authors. George B. Mo implemented the Pareto-frontier learning algorithm, which I adapted for various applications. Additionally, Chun-Lien Cheng developed the Unity program for a specific target application. I was involved in planning and conducting two user studies with the support of Prof. Liwei Chan and Chun-Lien Cheng. Throughout the project, all co-authors contributed to the ideation and writing process. Dr. John J. Dudley provided particularly valuable support during the writing and submission of the paper.

Publication IV: “Button Simulation and Design via FDVV Models”

The idea of simulating the tactile characteristics of push-buttons was initially proposed by Prof. Antti Oulasvirta and Prof. Sunjun Kim. In my first attempt to achieve button simulation (published in UIST ’18 Adjunct), I followed the traditional Force-Display model approach, but it turned out to be unrealistic for certain types of push-buttons. Therefore, I proposed a more sophisticated FDVV model for capturing button-pressing together with Prof. Sunjun Kim. I further developed an end-to-end button simulation pipeline that starts from profiling a button to rendering its characteristics on a physical simulator. Throughout the development process, Prof. Sunjun Kim provided valuable support in prototyping and implementation. Prof. Byungjoo Lee provided close guidance on the ideation, model development, and formulation of the temporal-pointing application, contributing to the project’s overall success. Prof. Antti Oulasvirta was also closely involved in the overall development of the project. The paper was written collaboratively with all co-authors.

Publication V: “Rediscovering Affordance: A Reinforcement Learning Perspective”

I proposed a new theory and a novel model for affordance formulation and perception. I designed and conducted two user studies, and ran a simulation experiment that employed a reinforcement-learning agent. All the authors jointly consolidated the research idea, and I played a key role in proposing the novel theory and model implementation. Prof. Antti Oulasvirta and Prof. Andrew Howes provided valuable perspectives from Cognitive Science and Psychology backgrounds, which is valuable in
theory formulation. Dr. Kashyap Todi and Dr. Aditya Acharya provided their views from design and machine-learning perspectives, which also contributed to the theory. Antti Keurulainen and Dr. Aditya Acharya helped me with the model implementation. Dr. Kashyap Todi provided valuable suggestions for the user study plan. The writing was mainly done in collaboration with Dr. Kashyap Todi and Dr. Aditya Acharya, and all the co-authors were involved in the writing process.
List of Figures

1.1 Design thinking’s double-diamond model for the design process. .................................................. 27
1.2 The design-optimization problem: within the design space (X), there are many design candidates (x), each constituting one “parameter setting.” Particular designs/settings lead to particular user performances, or objective-function values (y), which lie in the objective-function space (Y). .................. 28
1.3 At left, a diagram of the traditional UCD workflow for design optimization. The right-hand pane illustrates a human-in-the-loop workflow. Note the dashed line in the HILO workflow, which indicates that adjusting design parameters and objective functions is an optional step. In contrast, UCD is an iterative process, and redesigning is necessary. .................................................. 31
1.4 The state of the art of HILO using Bayesian optimization. It handles design problems with a single objective, and it optimizes for a single user at a time. .................. 32
1.5 The contributions of this dissertation. .................. 38

2.1 Adapted from the figure in Martins and Ning [145]: Illustration of the manual design workflow in engineering fields. 45
2.2 Adapted from the figure in Martins and Ning [145]: Illustration of the optimization-driven workflow in engineering fields. .......................... 46
4.1 Two example objective functions, illustrating the Pareto-optimal design set and the Pareto hypervolume. At left, points \((f_1(x_i), f_2(x_i))\) are shown as dots, with red ones representing the Pareto-optimal design set and black ones representing dominated points. The gray region is the current Pareto hypervolume with respect to the reference point \(v_{ref}\). With pane b, a new observation is made at \((y_1', y_2') = (f_1(x'), f_2(x'))\), which dominates one point that was previously Pareto-optimal. The cyan region is the Pareto hypervolume increase after the observation (the green point). If the new observation is dominated by some previously observed point, there would be zero change in Pareto hypervolume.

4.2 Multi-objective HILO conducted by a designer.

4.3 The prototype (a), experiment setup (b), and interface (c). Publication I provides more details of the setting.

4.4 Boxplots showing the ratings given by the designers in the SUS questionnaire. Publication I provides details.

4.5 Boxplots presenting the eight designers’ ratings from the NASA-TLX instrument. Publication I provides further details.

4.6 A diagram of the 3D touch interaction. (a) illustrates the cursor position when the length of the real arm vector is less than the threshold distance D. (b) showcases the cursor position when the real arm vector is beyond the threshold distance D. (c) shows the vibration feedback. More details are given in Publication II.

4.7 In the study, the designer was also the user who observed the final results.

4.8 Average completion time and spatial error (a) and the ratings of the general experience for its sense of satisfaction, confidence, agency, and ownership. The result indicates that the optimizer-driven procedure derives better user performance, but sacrifices the designer’s sense of agency and ownership. More details are presented in Publication II.

5.1 Illustration of the mechanism of the Global GP.

5.2 Illustration of the mechanism of the Warm-Start GP.

5.3 The Global GP aggregates all observations from the user-specific optimization processes. The consolidated model can thereby estimate the group’s average performance at any given design parameter value. After constructing the Global GP, I perform a fine-grid sampling of the design space to identify the global Pareto-optimal design instances.
5.4 The predicted Pareto-optimal objective values from Global GP and the global Pareto-optimal designs. 88
5.5 Results of the comparative study on three designs. The result indicates that the designs generated by the Global GP outperform the Go-Go Technique in completion time. Further, the accuracy-oriented design also outperforms the other designs. Please refer to Publication 3 for further details. 89
5.6 The smartphone prototype with a touch-sensitive button. (a) The smartphone prototype and the touch point. (b) The vibration motor is placed inside the prototype for emitting vibration feedback. Please refer to Publication III for more details. 91
5.7 Illustrative design example of the target interaction. Please refer to Publication III for more details. 91
5.8 A simplified sketch of the study interface during button-pressing (a): the participant is asked to activate the button when the red bullet reaches the yellow target area, at which point the bullet turns blue. After 24 presses, the user is asked to rate the vibration cue (b). Pane c shows the study’s interaction in action. 93
5.9 The results from the evaluation of Warm-Start GP. More details are provided in Publication III. 94
6.1 The workflow of HILO with physical emulation. 100
6.2 A depiction of what the force–displacement–vibration–velocity model represents, detailed more fully in Publication IV. 102
6.3 Capturing the profile of a button. 102
6.4 The button emulator. 103
6.5 Iterative compensation finds a way to render an FDVV model via a given simulator plant. 104
6.6 The setup for the study. 105
6.7 The setting for the push-button optimization. 107
6.8 The results of the push-button optimization. The optimal button is the personalized optimal design. The Clear, Brown, Black, and Red switches are 4 mm mechanical Cherry MX switches rendered by our emulator. Lastly, the random button is a design generated randomly within the design space. 107
7.1 The simulation-based HILO framework. 112
7.2 The team developed a computational model of the affordance theory, implemented in the MuJoCo physics engine. It enables a virtual robot to interact with widgets. 114
7.3 The virtual robot interacted with widgets of several types: (a) button widgets; (b) slider widgets; and (c) a deceptive widget that, while resembling a slider, allows only push actions, similar to a button. ........................................ 114
7.4 The results of the model training and testing, detailed more fully in Publication V. ........................................ 115
7.5 The simulation of 3D touch. ................................. 116
7.6 The progress of agent training. .............................. 119
7.7 The achieved performances during the optimization step. 119
7.8 The results of the evaluation. The left plot shows the achieved hit rates, and the right plot shows the achieved efficiency. The error bar shows one standard deviation. .... 120
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Design parameterization for the haptic display</td>
<td>65</td>
</tr>
<tr>
<td>5.1</td>
<td>The differences between standard Bayesian optimization and Group-in-the-loop optimization</td>
<td>82</td>
</tr>
<tr>
<td>5.2</td>
<td>The parameters of the simulation tasks: the corresponding objective functions of each parameter and the parameters’ ranges</td>
<td>84</td>
</tr>
<tr>
<td>5.3</td>
<td>The simulation task’s objective functions and their ranges</td>
<td>85</td>
</tr>
<tr>
<td>5.4</td>
<td>The three design conditions evaluated in Phase 2</td>
<td>89</td>
</tr>
<tr>
<td>5.5</td>
<td>Design parameterization of the touch-button</td>
<td>92</td>
</tr>
<tr>
<td>5.6</td>
<td>The design objectives of the touch-button</td>
<td>92</td>
</tr>
</tbody>
</table>
Abbreviations

AF  Acquisition function
AR  Augmented reality
AutoML  Automated machine learning
BO  Bayesian optimization
CQ  Chapter-level research question
EHVI  Expected Hypervolume Improvement
EI  Expected Improvement
FD  Force-displacement model
FDVV  Force-displacement-velocity-vibration model
GP  Gaussian process
HCI  Human-computer interaction
HILO  Human-in-the-loop optimization
IT  Information transfer rate
MOBO  Multi-objective Bayesian optimization
PI  Predicted Improvement
RL  Reinforcement learning
RQ  Dissertation-level research question
SM  Surrogate model
UCB  Upper Confidence Bound
UCD  User-centered design
UI  User interface
VR  Virtual reality
Symbols

\( f \) a target function to be optimized; in the context of human-in-the-loop optimization, an interaction can be seen as a function that maps a design candidate \( x \) to objective function value(s) \( y \), such that \( f(x) = y \).

\( \mathbb{R} \) real number.

\( X \) a design space, which is composed of \( n \) design parameters \((X \in \mathbb{R}^n)\).

\( x \) a specific design candidate or parameter setting; such a design candidate is within the design space \( X \) \((x \in X)\).

\( Y \) an objective-function space, which is composed of \( m \) objective functions \((Y \in \mathbb{R}^m)\).

\( y \) one value or a set of values of the objective function(s); such objective-function value(s) is within the objective-function space \( Y \) \((y \in Y)\).

\( \emptyset \) empty set.
1. Introduction

“Don’t ever make the mistake [of thinking] that you can design something better than what you get from ruthless massively parallel trial-and-error with a feedback cycle.” — Linus Torvalds

“True optimization is the revolutionary contribution of modern research to decision processes.” — George Dantzig

An interaction technique is a system comprising hardware and software elements that enable users to complete tasks [66]. Design parameters define the characteristics of an interaction technique. For instance, among a touchscreen button’s design parameters might be its length and width, its color, and the force required to trigger it. Since the values assigned to those parameters have a significant impact on the usability and user experience [2, 86], appropriate exploration of their various possible values is critical in the design process for any given interaction technique [186]. However, exploring the vast design space containing all possible parameter settings is a challenging task. Traditionally, designers perform iterative manual optimization, exemplified by the user-centered design (UCD) workflow [1], which involves the designer proposing a design candidate and developing a prototype. The design that emerges undergoes user testing [142], with the results guiding the designer’s iterative tuning of the idea until an acceptable design is reached. While manual optimization in iterative user-centered design is a viable approach for exploring the design space, it does not guarantee an optimal result. The central challenge stems from the large design space that most design tasks bring. For example, even a single physical button, or “push-button,” requires a design parameter specifying the travel range, a separate one for the point at which a press gets registered, and at least two defining haptic feedback [56]. Because all these parameters are continuous, the number of possible design combinations is infinite. Even restricting each parameter to 10 discrete levels still entails 10,000 possible combinations. Hence, exhaustively exploring
all potentially good design candidates proves nearly impossible.

Wrestling with such a large design space is not the only challenge of the manual optimization process. It also takes time and consumes other resources. In our example case of designing a push-button, the interaction designer must develop a proposed design, making assumptions about the expected result along the way, after which a developer or engineer has to create a corresponding prototype [228]. These efforts are followed by those of a user researcher, who conducts a user study and analyzes the data. Finally, the designers have to interpret the results together and propose a new design. The significant time and effort demanded by this multi-phase process renders exploration even more effortful [1]. Moreover, manual design is susceptible to bias and fixation issues, since it relies heavily on the designer’s experience and prior knowledge. With multiple, possibly conflicting design objectives, carefully balancing the objectives and predicting the effects of each alteration to the design while exploring the vast design space demands immense mental effort. In consequence, designers often settle for designs that “feel right” instead of pursuing a genuinely optimal solution.

**Human-in-the-loop optimization (HILO)** offers an alternative: rather than relying on designers to choose the next design for evaluation, the process uses the computational optimizers to intelligently select the design instances while *human subjects* evaluate the design instances by interacting with them. Please refer to Figure 1.4 for an illustration of the concept of HILO. While HILO has demonstrated potential to enter the picture as a more principled optimization procedure [54], it has been employed for a few specific applications only [111]. One significant limit in scope stems from today’s confinement of attention to single-objective problems, whereas realistic design challenges often involve tradeoffs among several objectives. For instance, designing any input devices often needs to consider both the efficiency and accuracy of the interaction. Multi-objective optimization is required to fully address this challenge. Another issue is that current HILO methods optimize designs for a single user, while practical design work often must optimize at the population level, for various user groups [165]. Without a population-level optimization, optimization needs to be deployed on every single user, leading to low efficiency. Additionally, HILO thus far has been applied only for user interfaces whose design process does not include physical prototyping, with the main reason being the time-consuming nature of such prototyping [15]. A more general factor holding back progress is that the benefits and drawbacks of applying optimization in the design process have gone unexplored. Accordingly, designers are less strongly motivated to make wider use of HILO procedures. Lastly, the costs remain considerable: HILO still requires user interaction in each iteration. Yet, recruiting user participants and conducting user studies is a costly step, which limits the use of HILO in design practice. It is also worth
expanding the possibility of utilizing simulated human users to boost the efficiency of HILO. It is clear that the aforementioned challenges hamper applying HILO to realistic design problems today.

The primary goal behind this dissertation is to expand the applicability of HILO for design interactions. I developed a set of novel computational methods to this end. Furthermore, I conducted investigations to evaluate these extensions’ performance and effectiveness, thus contributing to a comprehensive understanding of the strengths and limitations of HILO. Ultimately, these advances should enable HILO to address a wider range of realistic design problems. With this chapter, I begin my presentation of them by outlining the challenges inherent to design-optimization problems and introducing human-in-the-loop optimization. My research questions then can be situated against this backdrop. Finally, I describe the research objectives, methods, and contributions.

1.1 The Design-Optimization Problem

Practitioners employ various design processes and models; among them, the double-diamond model [162, 13] (see Figure 1.1) applied in “design thinking” is an especially well-established framework describing the general design process. It extends all the way from the initial broad design goal to a final design. My focus is on the latter part of the process, the model’s last two steps: developing and exploring possible design solutions and, then, narrowing the set to deliver a final design. Specifically, the dissertation addresses only the parametric optimization problem [82], where the design challenge is to determine the optimal set of parameter values for a given interaction. Figure 1.2 illustrates this problem: the designer considers a specific design space, within which each design
candidate is a unique parameter configuration. Varying the parameter settings produces different objective-function values, which represent user performance; together, all the possible user performances constitute the objective-function space [173]. The designer’s goal is to identify the best design – i.e., a combination of parameter values that leads to optimal performance. My discussion below refers to said parametric optimization problem as “design optimization” and formulates the research problem on this basis.

1.1.1 Design Parameters and Design Space

The design optimization process revolves around the design space, which contains all the valid design candidates. In this dissertation, the design space is denoted as $X$. Each design parameter within the space represents a variable to which the designer can assign a value. For example, one of the design parameters in Web-page design could be the font size of the header, with the designer assigning it a certain value. Design tasks usually involve multiple design parameters, and the design space is multi-dimensional. We can mathematically formulate the design space as $X \in \mathbb{R}^n$, where $X$ has $n$ real-number design parameters. For instance, the length, width, and height parameters in our push-button design example might range from 1 cm to 3 cm. Therefore, we can formally describe the design space as $X = [1\text{cm}, 3\text{cm}]^3$.

A design instance, denoted as $x$, is a possible set of values within the design space $X$. For instance, a design instance of the push-button could have a width of 1 cm, a length of 2 cm, and a height of 1.5 cm. This dissertation uses the terms “design instance” and “parameter setting” interchangeably.
to refer to a possible design $x$ within the design space $X$. We can also denote the latter relationship as $x \in X$.

### 1.1.2 Objective Functions

In design optimization, the objective function represents the performance metric for which the designer aims to optimize. Its value may be minimized or maximized. This metric can reflect either some measurable performance (completion time, success rate, etc.) or a subjective rating (e.g., Likert-scale comfort level). For any interaction, the designer might aim to optimize for a single objective function or several. For instance, in the case of an input device, there might be objective functions for accuracy and efficiency both. Mathematically, we can represent one or several objective function(s) as $y \in \mathbb{R}^m$, where $m$ is the number of objective functions, and all the measured objective value(s) are within a continuous range. Each $y$ contains a set of objective values of an interaction.

All the possible objective function values jointly form a multidimensional objective function space $Y$, which one may formally represent as $y \in Y$.

### 1.1.3 Interactions as Black-Box Functions

In design optimization, each design instance $x$ leads to an objective-function value $y$, which could be measurable or more subjective. To represent the relationship between an interaction’s input ($x$) and its output ($y$), we can use a function ($f$) that maps the design instance to the objective function’s value. Since we seldom possess prior information about the function, we consider it a black-box function [3]. As depicted in Figure 1.2, we can mathematically express the interaction as $y = f(x)$, where $f$ (the interaction) takes $x$ (a design instance) as input and generates a response $y$ (a value of the objective function(s)).

In the example of a button-pressing interaction, changing the size of the button keycap ($x$) can affect such aspects of user performance as typing accuracy ($y$). Although we may have some prior assumptions about the relationship, we generally lack detailed knowledge that would allow us to predict the resulting objective function. Moreover, recall that an interaction can have multiple objective functions. For example, good Web design can improve not only search efficiency ($y_1$) but also visual comfort ($y_2$), among other elements. Therefore, solving the design-optimization problem presents several challenges, expanded upon below.

### 1.1.4 Challenges of Design Optimization

Optimizing design is the process of identifying the optimal design instance(s) $x$ within design space $X$ whereby the optimal objective function(s)
y is/are produced. Several factors make this a difficult task. I present a few of the more prominent ones below.

**The Large Design Space**
Design optimization often is complicated by the need to deal with large design spaces [104]. With every additional design parameter, the design space grows exponentially and it becomes more difficult to identify the best design instance. For example, when each parameter is discretized to 10 levels, a single-parameter design task involves 10 design candidates in total; in contrast, a task with four design parameters presents more than 10,000 possible combinations to contend with. The limited time and monetary budget for manual optimization make an exhaustive search impractical; therefore, effective navigation of the large design space necessitates optimization algorithms.

**Complex Interaction**
The association between design parameters and the objective function is a crucial facet of design optimization. Understanding their interaction can be challenging, however, since it often manifests itself as a black box: one knows only the output value for a given input. This lack of information makes it difficult to optimize the design effectively. To obtain the pairs of output and input values needed for the accurate evaluation of the design, one must carry out user testing, and the manual search for optimal design instances can be mentally and physically demanding.

**Tradeoffs and Multiplicity of Objectives**
The next issue arises from the frequent need for design optimization to consider multiple objectives [173]. Because optimizing for one objective sometimes comes at another objective's cost, it can be difficult for human designers to make sense of and predict the best tradeoff. For example, improving efficiency may require sacrificing accuracy. Especially when there are more than two objectives, it may be very challenging to identify the optimal design, balancing all of them effectively.

**Biases and Design Fixation**
Lastly, alongside the challenges rooted in the problem itself, human designers face their own biases [53]. To select the design candidates to be evaluated, designers often rely on their experience and domain knowledge. That experience comes with potential blind spots such as overlooking some options and getting fixated on other, suboptimal designs [98].

In summary, the complexity of the problem, the large design space, and the difficulty of balancing several objective functions render it mentally demanding to seek optimal design instances manually. Since an exhaustive search is not practical in these conditions, designers have to pick the best
design from the small subset of design instances tested. In this process, the optimal one might well end up omitted.

1.2 Human-in-the-Loop Optimization

Human-in-the-loop optimization offers an alternative approach to tackling the challenge of design optimization. In contrast against reliance on human designers to pick each design for evaluation, HILO employs a computational optimizer to generate the next design for user testing [199, 28] (as illustrated in Figure 1.4). By ruling out human biases and fixation, HILO allows for a more principled and systematic search. Moreover, by eliminating the analysis step featured in the traditional UCD process, HILO yields greater efficiency and reduces the number of empirical studies and analyses needed (as illustrated in Figure 1.3).

In the HILO framework, the designers’ main task is to set up the optimization, including specification of the design space and objective space, and conduct the HILO. Notwithstanding its various potential advantages, HILO is not widely applied by human–computer interaction (HCI) or design practitioners, mainly because of the many questions that still need to be addressed. Therefore, the main goal I set for this dissertation is to address those open questions, elaborated upon in subsection 1.2.2 (“Research Questions”), thus promoting the use of HILO in realistic design optimization by expanding the application scope. This project for addressing the challenges of design optimization enabled me to propose a series of novel methods based on Bayesian optimization. Next, I introduce Bayesian optimization, which has gained popularity in the HILO domain because of its efficiency and its support for extensive customization.

1.2.1 Bayesian Optimization

Bayesian optimization (BO) is characterized as a powerful optimization method for solving expensive black-box functions [68, 29] (again, problems
or systems of which we have no knowledge, where the only way to obtain information about what lies within the “box” is by evaluating them and obtaining one pair of input $x$ and output $y$ at a time). Because evaluating black-box functions can be expensive in more than one respect, a grid search over all possible inputs is impractical. As introduced earlier, the interaction between users and interactive systems can be seen as precisely such a function, and evaluating each design usually demands time and effort.

BO is not the only black-box optimization algorithm. While there are evolutionary algorithms [9], genetic algorithms [105], particle-swarm models [107], and others [3], BO exhibits consistently promising performance [24]. Additionally, its customizability represents a great strength that other algorithms do not offer. There are two elements at the core of the BO framework: Gaussian Process regression (GP) and the acquisition function (AF). The former has several hyperparameters, which users can set to achieve the best performance for their particular problems, and various “kernels” (discussed further on). Similarly, there is a wide range of possible AFs. Users can fine-tune each of these by setting hyperparameters. To adjust the BO to case-specific needs, users can even implement their own GP or AF. Thanks to these advantages, BO has become the most mainstream method of human-in-the-loop optimization and formed the core of the dissertation project. I briefly introduce key elements of BO below, with a more detailed description provided further along, in section 2.4 (“Bayesian Optimization”).

**Surrogate Model (SM)**
The surrogate model (SM), a key component both of BO and of other model-based optimization methods, is constructed in line with the observations made as optimization progresses. It serves as a lower-cost alternative to the actual function [92]. In BO, the SM is a Gaussian Process regression (GP) that is fitted with all the observations thus far [222]. By means of
GP, we can predict the output ($y$) and the variance of the prediction, which represents the level of confidence in the predictions. Exploiting the evaluations of the SM, BO efficiently chooses the next input ($x$) to evaluate with the function proper. The ability to customize hyperparameters and select from among various kernels in a GP-based SM further enhances flexibility and performance. This has contributed to the technique’s popularity as a mainstream method for conducting HILO. The surrogate model in BO plays a vital role in making the optimization process more efficient and effective.

**Acquisition function (AF)**

On the basis of the information provided by the SM, the acquisition function provides the acquisition value ($AF(x)$) of a particular input $x$ [29] – i.e., the worth of evaluating the actual function with this input $x$. At the beginning of each iteration, BO samples many design candidates $x$ and expresses queries for the acquisition values $AF(x)$. Then, the design instance $x$ with the highest value is selected to be evaluated in this iteration. Various AFs have been proposed for this core element of BO: the popular upper confidence bound (UCB) function, expected improvement (EI), etc. These seek balance in the exploration–exploitation tradeoff and guide the search toward promising regions of the input space. In addition to these now-standard AFs, researchers have proposed many novel AFs to address specific challenges posed by diverse domains. Some AFs are designed to handle multi-objective optimization, while others are intended specifically to address noisy or dynamic objective functions. These novel AFs often incorporate domain-specific knowledge and heuristics to improve search efficiency and effectiveness.

**General Procedure of BO**

The general procedure of BO is described below via Algorithm 1, an adaptation from Frazier’s introduction paper [68]. Given an expensive target function $f$ and a budget (maximum iteration count) $N$, BO first initializes a GP instance as the SM. Then, BO randomly samples $n_0$ data points across the entire prospective design space. Within each optimization iteration, GP updates are performed on the basis of all the observations, the algorithm computes the acquisition values for many $x$, and it picks the $x$ with the highest acquisition value for function evaluation. This continues until the iteration number reaches $N$. The $x_i$ that has the optimal $y_i$ function value is returned as the final output.

**BO in Human-in-the-Loop Bayesian Optimization**

The flexibility and effectiveness of BO have helped it gain popularity in the field of human-in-the-loop optimization. One of the earliest projects to utilize BO in HILO was by Brochu et al. [28], who used BO to identify the
Algorithm 1 General Procedure of Bayesian Optimization

1: **Inputs:**
   A target function $f$.
   The optimization budget $N$.
2: **Outputs:**
   The optimal parameter setting $x$
3: **Initialize:**
   Place a Gaussian Process as the surrogate model.
   Observe $f$ at $n_0$ points. Set $n = n_0$.
4: **while** $n \leq N$ **do**
5:   Update GP using all observation data.
6:   Let $x_n$ be a maximizer of the acquisition function over sampled $x$.
7:   Observe $y_n = f(x_n)$.
8:   Increment $n$.
9: **end while**
10: **return** $x_i$ that has the optimal $y_i$ function value.

optimal hyperparameter settings for computer rendering of realistic animation. They showed that, with the support of BO, users could efficiently identify a good design. Later, Khajah and colleagues [108] employed BO to maximize user engagement in gaming, while Yamamoto et al. implemented it for efficiently facilitating a photographer’s setup process for lighting devices [227]. These projects attest to the effectiveness of BO in HILO and how it can be customized for various optimization problems. Further examples are considered in chapter 3 (“Related Work: HILO in HCI”).

1.2.2 Research Questions

While BO-based HILO has demonstrated success in specific cases, certain fundamental limitations continue to restrict its application range. In aims of expanding the usability and application scope of HILO through doctoral research addressing these limitations, I articulated several research questions. The set of questions dealt with in this dissertation (RQs) is structured thus:

**RQ 1. How to enable multi-objective human-in-the-loop optimization?** Most HILO research has focused on single-objective optimization, but real-world design tasks often involve multiple objectives, and how to extend HILO for approaching such tasks has remained unclear.

**RQ 2: What are the benefits and limitations of human-in-the-loop optimization?** Although research has shown that Bayesian optimization can assist with the design process in HILO settings, its effectiveness level and designers’ perceptions of it are less evident.
RQ 3. How can group-level human-in-the-loop optimization be achieved? All prior work has focused on optimizing for an individual user from scratch. However, design practice often aims at 1) optimizing for a group of users or 2) efficiently personalizing a user-specific design from the starting point of a default setting. Current BO methods are not able to support such aims.

RQ 4. How can we equip HILO for interactions that require physical prototyping? Current HILO methods are restricted to a set of user interfaces for which varying the design instance does not require different physical prototypes. Many physical interfaces (e.g., with a mouse, keyboard, or controller) remain excluded from HILO because prototypes must be fabricated for each iteration.

RQ 5. How could we further reduce the cost of conducting user experiments? Completing full-fledged HILO requires human participants serving as evaluators, which can entail high costs for both designers and these users. Since designers may not always have the budget for such HILO, it is important to find ways to reduce the cost of conducting user experiments.

1.3 Research Objectives and Methods

By means of five research publications, I offer solutions responding to all of these research questions. To address RQ 1, I propose multi-objective Bayesian optimization (MOBO) with Pareto-frontier learning [198] for HILO. Rather than a final design for a single performance metric, MOBO obtains a series of Pareto-optimal designs. This affords designers the flexibility to choose the final design amid the tradeoffs across all objective functions. The corresponding articles are publications I and II.

To respond to RQ 2, I conducted several lab-based controlled studies and a workshop to investigate the positive and negative qualities of HILO. These investigations provided insight into the benefits of HILO such as improved performance and shortcomings such as the designer's sense of losing agency and ownership of the design process. Publications I and II describe this work too.

For answering RQ 3, I propose the novel concept of group-level HILO. This entails aggregating optimization data across a group of users to arrive at designs optimized for groups or a fast-adaptation model for rapid design customization. The concept is dealt with in Publication III.

My answer to RQ 4 involves a proposal to replace physical prototyping with physical emulation. In contrast to the former, a financially prohibitive and time-consuming process, emulation employs a hardware or software system that resembles the various relevant systems in its behavior. Build-
ing an emulating system that can render multiple designs of a physical interface instantly should save time and eliminate the cost of constructing real-world prototypes, thereby enabling HILO for physical interface design. The corresponding paper is Publication IV.

Finally, to address RQ 5, I propose a novel simulation-based optimization framework. In this framework, which uses models to simulate users’ behaviors and also the human–interface interaction, a reinforcement-learning-based agent learns to interact with a particular interface. The optimizer generates various design instances of the interface, whereby the framework can derive the optimal design in the simulation environment. This contribution is presented in Publication V.

1.3.1 Method #1: Optimization

In pursuit of the overarching goal behind the dissertation – to expand the application scope of HILO – I developed novel optimization methods or adapted existing ones to HILO applications. For the response to RQ 1, I worked with established multi-objective Bayesian optimization with Pareto-front learning, implementing the technique for HCI design tasks. The method’s applicability for HILO had gone unexplored, and my work filled the gap by demonstrating its effectiveness for handling multiple design objectives.

To address RQ 2, in turn, I explored two extensions based on multi-objective BO for group-level optimization in the context of HILO. The first extension proposed, Global GP, is a unified large model that is constructed to operate from all observations across all of the users. This can be used to derive the group-level optimized design instance(s). The second extension, Warm-Start GP, is a variant of the sparse Gaussian Process method that selects the most representative data points to inform a lightweight model, which can serve as the prior for future optimization. Both extensions improve the scalability and efficiency of HILO for group-level optimization tasks.

1.3.2 Method #2: Emulation, Prototyping, and Modeling

Another key facet of my project is the use of emulation, prototyping, and modeling to enable applying HILO for physical interfaces (under RQ 4). Specifically, for the task of button-pressing, I developed a novel interaction model that captures the nuances of this action. This work was followed by examining a series of prototyping methods (including 3D printing, laser cutting, soldering, and circuit assembly) to employ for the construction of the emulator. Finally, to emulate the button-pressing interaction well, I pioneered an emulation workflow that encompasses data-gathering, signal-processing, and control.
1.3.3 Method #3: Simulation, User Modeling, and Theory

My work addressing RQ 5 involved a simulation framework that employs the MuJoCo engine for a set of optimization applications managed by means of physical simulation. To address the need for a user model to inform the simulation of human behavior within the framework, I utilized policy-based reinforcement-learning methods to construct a user model capable of interacting with various widgets and interfaces. To enable simulated users to learn the correct actions for the various objects, I introduce the final component of this method, a novel theory proposed for explaining the process of forming affordance perceptions. This theory helps shed light on how humans perceive and understand objects’ functionality and holds promise for aiding in the creation of more realistic user models in simulation frameworks.

1.3.4 Method #4: User Research

Evaluating the proposed methods’ effectiveness took advantage of several user-research methods. Specifically, a series of user studies investigated whether HILO outperforms traditional design approaches. For instance, an empirical study aligned with RQ 2 compared user performance between outputs from HILO and from traditional design approaches. Likewise, a comparison responding to RQ 4 examined the optimized button design vs. several preexisting button designs. In addition, for RQ 3, group-level optimization was compared with state-of-the-art HILO. Finally, I conducted a design workshop to gather subjective feedback from designers (eight designers were assigned two subjects for the completion of a design task). These studies were critical in assessing the proposed methods’ effectiveness and guiding further improvements.

1.4 Contributions

The work’s central contribution is the development of a set of novel methods and investigations that, by together enabling HILO to explore parameter design spaces across a wider range of realistic applications, enhance its utility for assisting with practical design problems. This dissertation focuses more specifically on developing methods based on Bayesian optimization. New methods and findings connected with design optimization constitute the core of my advances; however, the outcomes also enrich the HCI landscape generally. Several contributions are worthy of specific mention.

Contribution 1 – multi-objective human-in-the-loop design optimization via Pareto-frontier learning (see publications I and II):
While previous work on HILO has focused on a single objective, real-world design challenges usually involve multiple objectives. Therefore, I proposed using Pareto-frontier learning to achieve multi-objective HILO. The evaluation showed that designers using multi-objective HILO engaged in a more variety-rich exploration of the options and ended up with designs that led to better user performance than those produced by traditional approaches.

**Contribution 2 – investigating the qualities of multi-objective human-in-the-loop optimization (see publications I and II):** Via a series of investigations, I showed that multi-objective HILO reduces the design process’s total effort investment and offers a designer the flexibility of trading off between various objectives. However, designers felt less agency and ownership in these conditions relative to the typical, designer-led process.

**Contribution 3 – group-level human-in-the-loop optimization (see Publication III):** Standard Bayesian optimization devotes lengthy iteration cycles to reaching an optimal design for a single user. However, real-world design tasks usually are intended to produce designs for a user group, not single individuals. The dissertation presents methods that I propose, accordingly, for group-level HILO. The project’s research attests that these methods support a) deriving a group-level optimized design and b) implementing a rapidly adapting warm-start setting.

**Contribution 4 – human-in-the-loop optimization with emulation (see Publication IV):** Heretofore limited to interaction involving a static physical/mechanical form, HILO has not been applicable when fabrication...
at each iteration is needed. I have shown that physical emulation mitigates the need for fabrication, thereby enabling HILO’s implementation for such interactions. An example use case is push-button design supported by a button-emulation pipeline.

**Contribution 5 – a simulation-based human-in-the-loop optimization framework (see Publication V):** Finally, I developed a novel framework for moving the entirety of human-in-the-loop design optimization into the simulation domain. The framework enables design optimization without human participants; it utilizes agents as proxy users to interact with the design instances. A preliminary study demonstrated that agent-in-the-loop optimization generates reasonable design results without any users’ involvement.

### 1.5 The structure of the Dissertation

The dissertation’s synthesis portion is organized into eight chapters. In Chapter 2, I provide a review of the fundamental background to the work carried out, especially typical design processes and human-in-the-loop optimization. Chapter 3 covers optimization methods, Bayesian optimization, and its applications in HCI work, and (by way of a brief overview) emulation and simulation as used in that field. After that, I turn my attention to the research related to multi-objective optimization, and I investigate its effectiveness. Then, in Chapter 5, I address the challenge of group-level optimization and detail the two novel extensions developed. Chapter 6 details my proposed technique of emulation as a mechanism for overcoming the difficulties involved in the optimization of physical widgets. Continuing the advances in this direction, the seventh chapter introduces simulation-based optimization as a tool that goes further – for optimizing interaction techniques without imposing any need for physical interactions. Finally, Chapter 8 summarizes the key findings from the dissertation project and discusses the opportunities for building upon this work.
2. Background

The dissertation project’s contributions have their roots in fundamental work in several fields. This chapter presents a comprehensive review of that background, beginning with an overview of design processes, which are crucial in generating ideas and designs. These processes entail designers combing through various design options to identify the optimal solution, typically arrived at via empirical research methods. Conventional techniques’ manual design optimization exhibits limitations, which highlight the need for automatic and principled optimization methods. Another closely connected discipline, alongside design, is engineering design optimization. It employs computational optimizers to aid in decision-making as the course of the optimization procedure progresses. Though the problems tackled in engineering optimization are similar to those in the design field, the latter’s involvement of human users renders the cases examined in this dissertation more complex. That issue prompted me to review recent advancements in human-in-the-loop optimization, defined as a framework in which an optimizer generates design instances while user participants evaluate them. Because Bayesian optimization, a technique in widespread use for HILO, is at the heart of my project, I also provide an in-depth review of key BO concepts and terminology.

2.1 Design Processes and Empirical Research

Designing an interaction is a complex and multifaceted task that requires careful consideration and planning [80, 39]. For a better understanding of the various approaches and methods followed in design, researchers have devoted extensive effort to categorizing and studying the field’s many processes. Below, I review two design processes that are particularly relevant to the dissertation’s discussion: the double-diamond model of design thinking and user-centered design.
2.1.1 Design Thinking's Double-Diamond Model

In the 1960s, psychologists attempted to summarize humans’ procedures for applying creativity to solve complex problems [169, 81]. The 1970s saw design researchers begin documenting how designers solve problems in a more systematic way [48, 5]. That was the era in which the design field recognized the existence of “wicked problems” [190, 30], problems that are ill-defined and have no straightforward solutions. Design thinking emerged in response to the need for solving them nonetheless; the term refers to a set of cognitive, analytical, and practical procedures applied to tackle complex and open-ended design problems [186, 149, 50]. The purpose of design thinking is to obtain a deep understanding of users and problems. Into the 1990s, IDEO continued work on its own version of design-thinking processes and methods [33, 100], which laid the groundwork for the double-diamond model popular today (see Figure 1.1). Among its key concepts are “problem space,” a term originally coined by Newell et al. [162], and the notion of the design process’s “divergent” phase and “convergent” phases [13]. The double-diamond process itself [47], which ultimately became the most well-known framework for design thinking [178], was fruit of further refinements by the British Design Council.

This process comprises four phases. It begins with the “discover” phase, wherein designers gather information and gain in-depth understanding of the problem and user needs. For comprehensive discovery of the problem, the designers spend time interacting with the actual users. Next, in the model’s “define” phase, the designers apply their insight from the previous phase to refine and concretize the problem, moving from exploration of the problem space to focusing on a specific problem. After exploring the problem space in these two phases, the designers enter the “develop” phase, oriented toward seeking diverse solutions and inspiration sources that may answer the question specified. Many possible design candidates get proposed in this stage. Finally, with the “deliver” phase, the designers refine and narrow the set of solutions to arrive at a final design.

Since the last two stages are related to finding the solution, researchers often regard them as exploration of the “solution space.” While all four phases are important for the design process, the dissertation focuses on developing and delivering, specifically for design tasks wherein settings for particular parameters determine individual design variations. In this context, we can regard the two solution phases jointly as “manual design optimization.”

2.1.2 User-Centered Design

While providing a framework for high-level design principles, the double-diamond model is not furnished with the specific activities necessary for
manual design optimization. To address this gap, I consider another widely known design workflow, that of the UCD process [1]. It emphasizes close collaboration with actual users throughout the design process. Vredenburg et al. [218] define UCD, the origins of which lie in a concept from the early work of Norman [167, 166], as “the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and a multi-disciplinary approach.” Detail-level variations in the procedures notwithstanding, all version have several key steps in common, among them gaining understanding of the users and context, designing and prototyping, user testing, and analysis.

The first step in the user-centered design process is to gain understanding of the users and context. The designers investigate why the users need the interaction, how they actually engage in it, and in what context they use the interaction (although this step is essential to any design process, it lay beyond the scope delimited by my research questions, involving well-defined design problems only). Then, with a well-defined design space and set goals, which should be aligned with the findings from the first step, the designers carry out designing and prototyping. They create a working prototype for communicating with others and conduct user testing. The third step is user testing: the designers invite real-world users to interact with the prototype created. This step, aimed at understanding its usability, is also called user testing, a user study, or evaluation. User testing should yield data by means of which the designers can improve the design further. Once the data are gathered, the next step is analysis performed to produce useful information. Designers apply appropriate analysis to judge whether the prototype constitutes the final design. It may need improvement. Designers iterate over these steps until arriving at a final design. Informed by their analysis, they return to previous steps and make improvements until reaching an acceptable design instance. Note that, while these cycles may require revisiting the first step, the aforementioned scope factor precludes delving into it in the dissertation.

2.1.3 Empirical Methods for Design Evaluation

My review of prior work highlighted how crucial user testing is for all design processes as designers conduct experiments and analyze data to uncover the best design instance(s). Empirical research is a well-established quantitative approach to comparisons for particular design qualities [125, 142]. Also, HCI researchers often rely on empirical research and statistical analysis when setting parameters for interaction techniques.

Haptic feedback for button design is among the many examples one could consider. It involves defining parameters such as vibration duration, am-
Background

Background amplitude, and feedback point, all the way to more complex vibration profiles [41]. Even fairly simple interaction such as this demands multiple design choices: Designers search for the optimal design over a high-dimensional design space by means of manual tuning and experiments. Because this can be extremely time-consuming, researchers often select a few parameter sets by referring to experience and experimenting to determine the best ones. They repeat this process until arriving at an acceptable outcome, then iterating at higher level until a single final good design emerges [76]. This can never be the global optimum, however, because of the large number of unexplored designs.

Empirical experiments also serve effectiveness comparisons among various input methods, whether evaluation of desktop input devices such as mouse vs. trackpad [64] or comparison of input methods for mobile devices [168]. Recent studies have even compared particular gaze-input methods’ utility for virtual-reality input [44]. Empirical methods also inform settings for parameter values in pilot studies. In all of these cases, a complicating factor rears its head, though: again, the designers can only conduct comparisons for a small portion of the design space, thereby potentially omitting those design instances that are truly/globally optimal. Therefore, we need a more systematic method of selecting promising candidates.

Limitations of Manual Optimization

While the traditional approach allows designers to reach an acceptable solution, it faces several limitations. The most critical of these is that every step is costly: One must ideate a design instance and may have to create a prototype before user testing. Running a user study requires careful preparation, recruitment of participants, and all the effort of conducting the experiment. Then, the designer must analyze the data and extract useful information before being able to derive the next design instance. Often, prohibitively high costs prevent the designers from conducting a full-fledged UCD process in which they could explore every potentially good design sufficiently; instead, they explore only a few design instances in the vast space and then jump to a conclusion. With numerous design candidates possibly remaining untested/undiscovered, the final design is suboptimal. A further practical constraint is that designers often lack the time, technical, and human resources for completing multiple iterations of user testing. Ultimately, the final design depends purely on what “feels right” to the designers and the developers.

The other most critical limitation is that UCD hinges largely on the designer’s decision-making. Many sources assist in design decisions, such as the designer’s expertise in a specific domain, past experience, and even intuition. Although resources such as expertise and experience can be helpful, they may exert a harmful influence too. For instance, designers may suffer from design fixation and not explore the design space well.
In addition, it is important to recognize that the task of design optimization is quite complicated by nature. As subsection 1.1.4’s review of design optimization’s challenges attests, the optimization is a reasoning process aimed at understanding a complicated black-box function, whose input and output both may have multiple dimensions.

In summary, this process that is necessary in every design endeavor brings with it limitations and challenges that lead to inefficiency and a suboptimal outcome. In light of this vital issue, the next section reviews how engineering fields have explored the use of computational methods to aid in said disciplines’ design-optimization process.

### 2.2 Engineering Design Optimization

Engineering design optimization is methodology whereby algorithms and tools support engineers’ efforts to identify some optimal design solution(s) from among many other alternatives. Martins and Ning’s book *Engineering Design Optimization*, which provides an overview of the methods involved [145]. It points out that, compared to the traditional manual iteration process (depicted in Figure 2.1), the computational optimization workflow (presented in Figure 2.2) is a more principled technique. The computational optimization process offers three major benefits: better performance, lower cost, and less uncertainty. It is important to note that certain basic differences exist in design optimization’s application between the design field and engineering disciplines. To avoid confusion, I use the term “engineering design optimization” for optimization employed in the latter while “design optimization” denotes optimization in the design or HCI domain.

Engineering design optimization involves three fundamental components: design variables, objective functions, and constraints. **Design variables** are the parameters by which the design space is defined, where particular values assigned to the variables result in mutually distinct design options. The **objective function**, in turn, provides a metric for which the engineer aims to optimize, as introduced above. **Constraints** are the conditions that the optimization process must fulfill.

Engineering design optimization has a rich history and has been ap-


Figure 2.2. Adapted from the figure in Martins and Ning [145]: Illustration of the optimization-driven workflow in engineering fields.

plied extensively in a wide range of professions. Mechanical engineering [184, 192, 141], chemical engineering [19, 161], and architecture design [151, 201] are among the many domains that have employed optimization techniques. With the advent of complex optimization problems at the intersection of multiple domains, multidisciplinary design optimization has attracted attention as a promising avenue [144]. For instance, aircraft optimization is a complex problem that involves optimizing aerodynamics, structures, and controls [8, 84].

At this juncture, it is worth considering the aforementioned notable differences between engineering design optimization and design optimization in HCI. While the fields show similarities in their use of computational algorithms and methods such as gradient-based optimization, black-box optimization, and modeling, developers in engineering often have access to detailed models of physical phenomena, which allows for optimization through simulations. In contrast, HCI and design practitioners rarely have a perfect user model for a specific interaction at their disposal. Since this makes optimization through simulation difficult, human evaluators are typically needed, to interact with each design instance. To address this distinctive factor, the dissertation focuses specifically on applying black-box optimization methods without preexisting user models.

2.3 Human-in-the-Loop Optimization

Robotics and machine-learning researchers have carried out extensive work on the type of optimization that formed the core of my project. I begin this section with a review of HILO in these fields, and its application in design and HCI is covered later, in Chapter 3’s discussion of related work.

To build a robotics system that delivers better human–robot interaction, researchers have developed optimization techniques that retain the human aspect of the optimization process [54]. Many of their studies have focused
Background

on wearable robots and exoskeletons. For example, Zhang et al. [233] optimized an assistive exoskeleton system that led to substantial improvements for individuals during walking, and Ding et al. [55] proposed using HILO to optimize hip actions’ assistance through a soft exosuit. Fang and Yuan, meanwhile, employed it for adjusting wearable ankle robots to minimize metabolic cost [61], and other, similar work investigated using Bayesian optimization to regulate the step frequency of walking for the same minimization objective [111]. This line of research [63, 220, 97] takes the human as the source for the objective function (again, the measurement result that the systems aim to minimize or otherwise optimize). The system iteratively updates the relevant system parameter setting on the basis of the user’s response or performance. My project followed similar lines, with human participants in the process providing the evaluation of objective functions.

The goals and methods characterizing HILO’s extensive use in the machine-learning field differ from those in robotics. Machine-learning researchers are more focused on improving how models are trained by exploiting experts’ input and domain knowledge [225, 226]. They have employed HILO for data’s preprocessing and annotation. For example, Self et al. proposed interactive parameter adjustment informed by human models [197]. Gentile et al. introduced an interactive dictionary-expansion tool based on language models [75]. In other contributions, Zhang et al. summarize ways of efficiently extracting training entities with humans in the loop [234], and Liu et al. proposed using humans in combination with reinforcement-learning agents for efficiently labeling data [138]. Finally, HILO is widely applied in natural-language processing; e.g., humans may aid in parsing the data [88, 230]. While human-in-the-loop methods for machine learning form an important part of the overall picture, their use deviates from the goals and methods central for this dissertation.

2.4 Bayesian Optimization

As an optimization method suited to grappling with black-box functions, BO has proven to be one of the most popular methods for HILO in HCI and design tasks. Below, I summarize the basics of BO. This technique updates a surrogate model iteratively for each observation. Typically, this model is a Gaussian Process regression. Then, BO uses an acquisition function that takes the surrogate model as input and generates the next sample. Bayesian optimization is the most suitable for expensive functions (that is, for conditions in which the function evaluation is time-consuming or costly). Also, it exhibits its greatest effectiveness when there are relatively few parameters (e.g., with a dimension value below 20). For a more detailed introduction of BO, the reader is referred to prior literature [68, 29].
2.4.1 Mathematical Properties of Bayesian Optimization

BO is designed to tackle the optimization of black-box problems. We can define this set of problems as

$$\max_{x \in X} f(x),$$

(2.1)

where \( x \) is a design instance and \( X \) is the design space constructed from all the design parameters. The design parameters and objective functions should follow these principles:

- The input (\( x \in \mathbb{R}^d \)) for such a function \( f \) has \( d \) dimensions, where \( d \) should be less than 20 (a smaller \( d \) enables more efficient and effective searching).

- The objective function \( f \) should be continuous, so that it can be modeled by GP.

- \( f \) does not feature any known special structure (such as concavity or linearity) or need to meet certain requirements.

- When observing \( f \), we observe only the output of the function \( f(x) \), not the first or second derivative. Hence, this is a gradient-free task.

- We regard \( f \) as a black box, and BO searches for the global optimum.

The general algorithm for BO is described by Algorithm 1’s pseudocode.

2.4.2 The Gaussian Process

Since BO typically uses GP as the surrogate model, I briefly introduce the mathematical definition and properties of GP, again per a concise version adapted from work by Frazier [68] (for more in-depth description of GP, please refer to Williams [222]). This form of regression is a Bayesian statistical method for fitting black-box functions. Assuming we have a set of design instances \([x_1, x_2, x_3, ..., x_n]\) and the corresponding objective function value \([y_1, y_2, y_3, ..., y_n]\), GP takes all these observations and fits them into a multivariate Gaussian distribution with a particular mean vector and a particular covariant matrix. In Bayesian statistics, this distribution is usually referred to as the prior distribution. The mean vector is derived via a mean function \( \mu_0 \) at each \( x_i \); meanwhile, the covariant matrix is derived through a covariance function (also commonly known as a kernel), \( \Sigma_0 \), for each pair \( x_i \) and \( x_j \), where both \( x_i \) and \( x_j \) come from the design instances we have gathered. We can formally describe the fitted prior distribution as
Background

\[ f(x_{1:k}) \sim \text{Normal}(\mu_0(x_{1:k}), \Sigma_0(x_{1:k}, x_{1:k})). \tag{2.2} \]

With this prior distribution, we can then infer the function value for \( f(x) \) at a particular position \( x \) via Bayes rules.

\[ f(x)/f(x_{1:n}) \sim \text{Normal}(\mu_n(x_{1:n}), \Sigma_n(x_{1:n}, x_{1:n})), \tag{2.3} \]

where the mean function \( \mu_n \) and the covariance function \( \Sigma_n \) are conditioned by the previous \( n \) observations. Such a conditional distribution \( f(x) \) is also known as the posterior distribution.

In summary, this process’s regression is a probability distribution that is derived from a mean function and a covariance function. We can update the distribution by means of the Bayes rule, and we can also infer the value of a certain point via conditional distribution. The technique’s user should select an appropriate kernel and set the hyperparameters before applying BO. Because evaluating the GP is much cheaper (in time and financial cost) than evaluating the actual function (problem), BO uses a special function to search the GP space and identify which point has the greatest value. For this GP evaluation function, we use the term “acquisition function.”

2.4.3 The Acquisition Function

The other essential component of BO is the acquisition function [223, 68]. The AF reads in a design instance \( (x) \) as input, and the outputs \( (AF(x)) \) indicates the potential value of evaluating this specific instance. After refitting the surrogate model, BO evaluates samples across the design space \( X \). Each sample results in an acquisition value \( (AF(x)) \). The design instance \( x \) that leads to the highest acquisition value will be selected for evaluation by the actual function (for the problem proper) in the next iteration. The most representative of the commonly used acquisition functions are introduced below. For a more detailed introduction to additional AF types, the reader is referred to Garnett’s work [73].

The Upper Confidence Bound:
One of the easiest AFs to handle, the UCB functions calculates an upper bound for each design instance \( x \). The UCB is composed of two parts: predicted mean and variance. A design \( x \) with a high predicted mean naturally has greater potential to lead to a higher objective value. Variance too is relevant for this potential: a design \( x \) displaying a higher variance value indicates that the surrogate model is rather uncertain of the prediction at this point; hence, there is potential for a high objective value to result irrespective of the mean. We can construct the UCB thus:

\[ UCB \equiv \mu(x) + \lambda\sigma(x), \tag{2.4} \]
where $\mu(x)$ and the $\sigma(x)$ are the mean and variance values at $x$, while $\lambda$ is the hyperparameter controlling the tradeoff between mean and variance.

**Expected Improvement:**

The EI value represents the expected improvement at a potential design $x$. At a high level, EI is composed of two elements: the first element is the difference between the predicted value of potential design $x$ and the best objective value observed so far, and the second element is related to the standard deviation at $x$. Intuitively, a higher EI value indicates greater potential improvement from sampling this point from the real-world function. We can formulate EI as

$$EI \equiv E[f(x) - f^*_n]^+, \quad (2.5)$$

where $f^*_n$ indicates the best observation within the past $n$ iterations and the + sign indicates considering only cases wherein this value is positive. If $f(x)$ is less than $f^*_n$, leading to a negative value, the EI function will be evaluated as zero.

**Probability of Improvements:**

The predicted improvement (PI) function calculates the probability of sampling at $x$ and retrieves a value that is better than the current optimal observation. This is similar to EI, but, rather than directly compare the mean value, PI integrates the probability by using the GP model. We can formally describe PI as

$$PI \equiv P(f(x) \geq f^*_n), \quad (2.6)$$

with $f^*_n$ indicating the best observation thus far and $P$ calculating the integrated probability.

### 2.5 The State of the Art in Summary

With this chapter, I have provided an overview of the fields that developed the foundation for this dissertation. For some of this background, I outlined the typical design processes, with particular attention to design-thinking models and user-centered design. As the design process nears completion, designers need to select the final design instance(s) addressing the design question specified. Traditionally, designers perform design optimization manually with the support of empirical research methods. However, relying on intuition coupled with empirical methods may not lead to the best outcome, since thorough empirical research requires a significant amount of effort. Since, accordingly, designers often end up choosing from among a considerably limited set of tested designs, a more principled approach is sought in optimization guiding the selection of design instances.
This points to a possible way forward via engineering design optimization, wherein researchers employ well-established computational tools to assist engineers in making decisions. However, fundamental differences between design and engineering settings, arising mainly from a lack of solid user models for specific interactions, typically necessitate retaining interaction with human evaluators when computational methods are deployed for design optimization.

The final element of the review, my examination of human-in-the-loop optimization in other contexts – robotics and machine learning – showcases HILO’s potential as a general framework applicable for a wide range of applications but also reveals that the approach has not yet seen widespread use in interaction design. Having pinpointed various constraints that currently restrict the application of HILO, I set out to address these gaps.
This chapter provides an overview of work specific to the techniques and domains most relevant to my project. I begin by reviewing human-in-the-loop Bayesian optimization, which formed the core of my research. Then, I examine a wide range of other works on HILO and adaptive user interfaces in HCI. I end the introduction by presenting the optimization toolkits available, in brief.

3. Related Work: HILO in HCI

3.1 Human-in-the-Loop Bayesian Optimization in HCI

This chapter narrows the focus from HILO as introduced in the previous chapter (a general framework to solve parameter-optimization problems in which human participants provide the evaluation functions) and from the BO approach to tackling expensive black-box functions, reviewed there as a promising approach for HILO tasks [199, 24]. Whereas (chapter 2 reviewed the use of HILO in robotics work largely concentrating on refining the parameter settings of wearable systems, we now direct our attention to reviewing research conducted in the HCI domain.

Most HCI work employing BO has used it as a design tool. In various ways, studies have addressed its requirement for the designers to provide objective function evaluation. Brochu et al. [28] demonstrated using Bayesian optimization in conjunction with human designers’ subjective ratings for quickly identifying an appropriate hyperparameter setting for realistic animation rendering. Koyama and various colleagues, in turn, proposed applying BO to aid visual designers who optimize for parameters of visualization via sequential linear selections [122] and gallery-based selections [121]. The aforementioned work in which Yamamoto et al. [227] utilized BO for automatically tuning photographic lighting settings is another example of the variety of applications. Other research, by Zhou et al. [237], showcased allowing composers and musicians to generate a melody via BO, and Piovarči et al. [176] used Bayesian optimization to search for design parameters optimized for target friction and vibration.
Related Work: HILO in HCI

objective metrics.

In experiments following a different technique, the end users act as the objective-function evaluators while the designer’s role is to supervise the optimization process. Khajah et al.’s aforementioned work with gamers to fine-tune game-parameter settings for maximal user engagement [108] is one example. Another such application of BO is work in which Kadner et al. [102] customized font designs to optimize the reading speed of individual users. Nielsen et al. showcased fine-tuning hearing devices with BO [164], while Snoek extended the gaze to other assistive technologies [204]. Also worth noting is Dudley et al.’s work [57], which utilized BO to optimize 2D map design directly via a group of end users.

Researchers have dedicated considerable effort to investigating means of making BO more suitable for optimizing interaction or design. An important stumbling block to BO’s higher efficiency is that it typically learns “from scratch.” Accordingly, Brochu et al. sought more efficient optimization by attempting to transfer the kernel learned from previous tasks to the current task [28]. Studies also have explored using crowdsourcing to refine design-parameter values quickly, as opposed to relying on a single user’s evaluation [57, 122]. For addressing high-dimensional design challenges, the Koyama teams’ publications considering the visual realm propose line search wherein the user makes one design judgment at a time. Taking a different task, Koyama and Masataka investigated having BO act as more of an assistant; here, the designer has greater flexibility to take or ignore the BO-produced design suggestions [120]. Finally, several studies have examined the use case for preferential BO in design tasks (i.e., the designer simply comparing two possible designs at a time) [227, 120].

These studies’ success notwithstanding, application of human-in-the-loop BO is still limited to the scope identified in the first chapter: its focus today remains on single-objective optimization cases, customizing for a single user, and interactions entirely confined to graphical user interfaces. It is precisely these limitations that motivated me to extend human-in-the-loop BO for other applications.

3.2 Online Optimization Systems in HCI

Besides BO, prior work in HCI has explored other computational tools for HILO and online optimization. To provide context, this section revisits the most representative of those tools.

**Bandit Systems**

An alternative approach still closely related to Bayesian optimization is the formulation of design-space optimization as a multi-armed bandit problem. This formalization refers to the task of selecting from among
several alternatives offering uncertain outcomes so as to maximize some gain – in the example of a row of slot machines, which machine should be played next if one wishes to maximize one’s winnings. Multi-armed bandits have been proposed and indeed demonstrated as a tool for assisting in interface design where the selection problem becomes one of choosing an interface-design alternative that maximizes some utility [137, 139, 4]. The same exploitation–exploration tradeoff is evident, in that the solution involves balancing learning more about potential alternatives vs. consistently preferring a known good design. It is possible to apply a Bayesian treatment [195, 199] to this selection problem, in which case the approach closely resembles Bayesian optimization.

**Adaptive User Interfaces**

Adaptive user interface are intended to provide means by which the interface or interaction technique adjusts to varying user capability, interest, and behavior. Extensive research into adaptive user interfaces has considered diverse computational techniques, across many devices and domains; I review only the most representative work here. Many adaptive interfaces are based on heuristics and logic. For example, Puerta et al. proposed a system that automatically generates design based on the user model and a set of “design rules” [182], and Gobert et al.’s system adapts menu in accordance with adaptation policies and the user's interactions [78]. These methods are limited in their adaptation scope, however. In more complicated cases, the heuristics or rules cannot fully cover the whole spectrum of possible scenarios. Moreover, these systems require a set of known rules, which is not a workable assumption for all interactions.

Looking beyond heuristics, researchers have also investigated exploiting computational solvers or optimizers to identify the design that optimizes best for given objective functions. For example, Belo et al. [60] proposed a toolkit for adaptive user interfaces that applies solvers to find the optimal design. AutoGain [126] is an optimization method that fine-tunes the transfer functions of input devices on the basis of user aim errors. The gain functions are iteratively updated during the user's interaction. Bailly et al. [11] proposed an online menu-optimizer with an objective focused on predicted user performance. For sketching, meanwhile, Todi et al. introduced an optimization-supported interface [214] that updates the positions, color, and sizes of user-interface elements as interaction with the designer progresses. The objective functions for this optimization were based on design principles.

While my project is aligned with this direction of research, in which a computational optimizer is applied for some given objective function(s), the scope of my work is delimited more narrowly. Firstly, while prior work has explored a wide range of solvers, from grid search [140] to evolutionary algorithms [11], the dissertation focuses principally on applications of BO.
One important reason for this focus on BO is that it offers a general framework that can be extended to a broad range of scenarios. For example, other optimization solvers may not be extended for such aims as my goal of extending HILO from a single objective to multiple objectives. In contrast, BO supports altering the acquisition function, whereby the optimization is guided to explore all the optimal designs in a high-dimensional objective space. The other crucial difference between my work and that on general adaptive user interfaces lies in the overall goal. This dissertation examines only parametric optimization tasks, while previous publications have tackled different aims. For example, some studies have focused on assignment problems (menu items and keyboard assignments), problems better served by methods such as combinatorial optimizations [172]. Other research has considered the effects of adaptation, such as learning and fatigue, over a longer interaction span [213, 46, 65, 83]. The optimization problems examined in this dissertation are specific to shorter episodes, not long-term human adaptation.

### 3.3 Preexisting Optimization Tools

The dissertation project employed several varieties of Bayesian optimization. Some optimization tools providing similar functionality are available for ready use by developers with programming abilities. Single-objective optimization is provided through established libraries such as scikit-optimize\(^1\) for Python and the Statistics and Machine Learning Toolbox\(^2\) for MATLAB.

With regard to the greater complexity that multi-objective optimization entails, current packaged implementations vary in their level of maturity and capabilities. BoTorch [12] is a relatively full-featured and actively developed library implementing a particular variant of multi-objective Bayesian optimization [51]. Among the other packages that support multi-objective BO are GPflowOpt [117] (in the process of being revamped into Trieste\(^3\)) and MOBOpt [72]. TS-EMO [26] is a MATLAB library implementation employing a closely related technique. These various projects chiefly target developers familiar with the techniques and, correspondingly, offer them good configurability. In contrast, with Bayesian Compass my research team seeks to offer the same core functionality but at a higher level of abstraction that provides better support for HCI researchers and developers naïve to the underlying techniques.

While we apply alternative optimization techniques, our goal is similar to that behind the implementations of Sağ and Çunkaş [193], Google

---

\(^1\) Available via https://scikit-optimize.github.io/.


\(^3\) See https://github.com/secondmind-labs/trieste.
Vizier [79], and pymoo [22]. These efforts give greater focus to the visualization of the optimization process and its results, in conjunction with the aim of offering the user greater abstraction. Bayesian Compass is targeted specifically at HCI researchers and developers who wish to apply advanced multi-objective optimization in their design problems. To this end and in contrast to prior work, we seek to provide inspection tools along with relevant scaffolding and guidance specific to the HCI domain. Furthermore, we aim to build a MOBO-exploiting design workflow that is readily usable and applicable for a wide variety of tasks and that, in comparison to traditional design methods, offers the design process demonstrable benefits.
4. From a Single Objective to Multiple Objectives

“There are no solutions; there are only trade-offs.” — Thomas Sowell

While work on human-in-the-loop optimization has restricted itself mainly to solving single-objective problems, the prevalence of real-world applications that may even involve competing objectives highlights a need for feasible multi-objective approaches. For instance, a good input-device design should balance accuracy and efficiency while also accounting for other relevant metrics. A design optimized exclusively for accuracy may not be efficient, in that it could allow highly reliable and precise selection but at the cost of significant time and effort. Conversely, a design that prioritizes efficiency may lead to low accuracy.

One commonplace approach for straightforwardly addressing multiple objectives in optimization problems is to aggregate several objective functions into a single scalar objective function by assigning weights to each objective. We can formally describe the weighted-sum approach thus:

$$\text{Weighted}_{-}\text{Sum}\_\text{Objective} = \sum_{i=1}^{n} \text{weight}(i) \cdot \text{Old\_Objective}(i),$$

where $i$ is the index of the original objective functions and $\text{weight}(i)$ is the weight assigned for a particular objective function. By aggregating multiple objective functions into one, we can perform single-objective optimization to solve the problem.

However, such an approach has several major drawbacks. The first is that objective functions differ in their units, so comparison of their magnitudes becomes difficult. For example, task-completion time, a popular metric for efficiency, is measured in seconds or milliseconds, while accuracy in tasks such as pointing is judged in terms of distance, with centimeters or millimeters as the unit. This makes it challenging to interpret their aggregate-level meaning. Additionally, aggregating multiple objective functions into a single one results in information loss, in that the original individual objective functions’ values do not get considered in
the optimization process. The optimizer observes only the weighted-sum value, which may not accurately reflect the true performance of the system. Moreover, the difficulty of a designer or developer predefining good weights for all the objective functions may lead to a suboptimal user experience. Another limitation of the approach is that the problems must have a small number of objective functions, since weighting multiple objectives becomes increasingly complex as their number rises. Lastly, aggregating several objectives into a single objective function exacerbates the issue of non-linear tradeoffs. For example, when greater accuracy entails decreased efficiency or when higher user satisfaction is accompanied by poor system performance, the relationship between the two factors may be far from obvious. Since a simple weighting system cannot capture this nonlinear relationship between objectives, suboptimal solutions may result. In short, conducting single-objective optimization with a weighted sum is not a principled solution.

With the discussion below, I aim to answer two chapter-level research questions (CQs):

**CQ 1: How could we enable HILO to tackle multi-objective design tasks systematically?**

**CQ 2: What are the qualities and performance traits of multi-objective human-in-the-loop optimization as compared to traditional design approaches?**

### 4.1 Multi-Objective Optimization

To address the multi-objective design problem, a more principled solution than taking a naive weighted-sum approach is to employ multi-objective optimization with **Pareto-front learning**. When conducting single-objective optimization, all we need to do is compare distinct designs by one metric, concluding with a final design that has the highest single-objective value. On the other hand, when performing multi-objective optimization, we have a set of objectives. In this case, instead of seeking a single optimum, we search for a series of design points that represent optimal balances of several objective functions. The set of design points is also known as the Pareto frontier, Pareto efficiency, or Pareto-optimal designs. Figure 4.1 gives an example with the Pareto-optimal design set shown as the points in red; these points exhibit the optimal tradeoffs between the two objectives, $f_1$ and $f_2$.

Clearly, a single optimum cannot be identified. As we progress along the Pareto front, compliance with one objective diminishes as that with the other grows. This pattern highlights the tradeoffs. We can formally describe Pareto-optimal designs thus: Consider an interaction technique with the function $f : X \rightarrow \mathbb{R}^m$, where $X$ is the design space, which has
From a Single Objective to Multiple Objectives

**Figure 4.1.** Two example objective functions, illustrating the Pareto-optimal design set and the Pareto hypervolume. At left, points \( \{(f_1(x_i), f_2(x_i))\}_{i=1}^{12} \) are shown as dots, with red ones representing the Pareto-optimal design set and black ones representing dominated points. The gray region is the current Pareto hypervolume with respect to the reference point \( v_{ref} \). With pane b, a new observation is made at \((y_1', y_2') = (f_1(x'), f_2(x'))\), which dominates one point that was previously Pareto-optimal. The cyan region is the Pareto hypervolume increase after the observation (the green point). If the new observation is dominated by some previously observed point, there would be zero change in Pareto hypervolume.

the parameter space \( \mathbb{R}^n \), and \( Y \) is the set of feasible objective-function vectors in \( \mathbb{R}^m \) such that \( Y = \{y \in \mathbb{R}^m : y = f(x), x \in X\} \). Specifically, \( m \) here corresponds to the number of design parameters and \( n \) refers to the number of objectives. An objective vector \( y'' \in \mathbb{R}^m \) is preferred over (i.e., strictly dominates) another vector \( y' \in \mathbb{R}^m \) when all of its elements are greater than the second vector; this is denoted as \( y'' \succ y' \). Formally, the Pareto front is expressed as the set \( P(Y) = \{y' \in Y : \exists y'' \in Y, y'' \succ y', y'' \neq y'\} = \emptyset \).

From an intuitive standpoint, one can regard the Pareto front as capturing the need to sacrifice satisfaction of a specific objective function when pursuing improved performance for another objective function. In our example of an input-device design with two objective functions (efficiency and accuracy), if we randomly select two designs on the Pareto front, one of them must exhibit higher efficiency but lower accuracy than the other. The idea behind employing multi-objective Bayesian optimization is to search for the Pareto-optimal designs, the designs that lead to the Pareto-optimal objective functions.

How, then, do we enable BO to search for the Pareto-optimal design? For an answer, we have to introduce the concept of Pareto-front learning.
4.1.1 Pareto-Front Learning

In essence, Pareto-front learning is searching for the Pareto-optimal designs for a multi-objective optimization problem. Prior work has demonstrated how to reach this goal by maximizing the Pareto hypervolume. To understand the latter concept, consider some reference design \( v_{\text{ref}} \) that is inferior to all of the Pareto-optimal designs. In practice, this can be taken as the point corresponding to the lower bounds of each of the \( m \) objectives. We define the Pareto hypervolume with respect to \( v_{\text{ref}} \) as the hypervolume bounded above by the Pareto-optimal design set and bounded below by \( v_{\text{ref}} \). A new design point that improves upon at least one of the Pareto-optimal designs would therefore yield an increase in hypervolume. This concept is illustrated in Figure 4.1 with two objectives. Pareto hypervolume can be used as a measure of how good the current estimate of the Pareto-optimal design set is since the more dominant the Pareto front, the larger the Pareto hypervolume. Thus, in multi-objective optimization, the Pareto hypervolume functions as a good proxy for the quality of a set of Pareto-optimal points.

Intuitively, we would like the next point sampled to increase the Pareto hypervolume as much as possible, since that would correspond to a significant improvement in the estimate of the Pareto front. There are various ways of reaching this target. For example, Yang et al. and Daulton et al. proposed differential hypervolume increase [229, 51]. Shah et al. [198], in turn, proposed an acquisition function indicating the approximate expected improvement in Pareto hypervolume in the case where objectives are assumed to be correlated, which they referred to as the correlated expected improvement in Pareto hypervolume. There are other implementations [58, 72, 52], but they share the same central idea – to maximize the hypervolume. Pseudocode for Pareto-front learning is presented below, as Algorithm 2.

For the next two sections of this chapter, I applied MOBO to two distinct design cases. In the work presented here, I relied mainly on the methods of Daulton et al. [51] and Shah et al. [198]. Future work should explore the efficacy of other implementations and benchmark the resulting performance.

4.2 Designing with Multi-Objective HILO

MOBO with Pareto-front learning offers a principled solution for multi-objective human-in-the-loop optimization, thus potentially speaking to CQ 1. Furthermore, to assess the efficacy of MOBO and the experience of using it in design tasks (per CQ 2), I invited designers to work on a design-optimization task (optimizing the design of tactile icons) in the context of
Algorithm 2 Pseudocode for multi-objective Bayesian optimization

1: **Inputs:**
   Take a given \( d, v_{ref}, n \rightarrow 1 \), and iteration count \( N \).  
2: **Initialize:**
   Randomly sample \( d \) parameter sets for querying the user, and set \( \mathcal{D} = \{x_i, y_i\}_{i=1}^d \).  
3: **for** \( n \leq N \) **do**
4:   Construct a Gaussian Process from \( \mathcal{D} \).  
5:   Compute the current Pareto-optimal design set.  
6:   Use Sobol sampling to obtain \( x_{new} = \text{argmax}_x \text{EHVI}(x|\mathcal{D}) \).  
7:   Query the user to obtain \( \{x_{new}, y_{new}\} \).  
8:   \( \mathcal{D} = \mathcal{D} \cup \{x_{new}, y_{new}\} \)  
9: **end for**  
10: **return** \( x_n \) with the optimal \( y_n \).

---

**Figure 4.2.** Multi-objective HILO conducted by a designer.

---

a workshop I organized. The workshop anchored in CQ 2 was oriented toward two goals: 1) understanding the designers’ experience of working with MOBO and 2) understanding how MOBO-based design differs from typical manual optimization procedures. The main finding emerging from the workshop is that multi-objective optimization significantly reduces the total effort required for making decisions in the course of a complex, multi-objective design task. My workshop also revealed that MOBO can effectively assist designers in managing tradeoffs between objectives.

### 4.2.1 Interaction: Tactile Icons

The design task in the workshop setting was to optimize the set of haptic feedback from a haptic display. One can convey information to the user in a simple manner with a single tactor by using unique vibration cues (combinations of vibration duration and amplitude) to represent each distinct message. Naturally, a set of vibrations that features more unique
combinations carries more information, but it becomes more difficult to distinguish between individual cues as the number of unique vibration cues increases. To constrain the dimensionality of the design space, I limited the task to four parameters: the designer had to specify the vibration duration and amplitude’s minimum values and set the number of distinct levels for duration and amplitude.

While the first two of these design parameters (representing the minimum duration and minimum amplitude of vibration) were specified by the designer, the maximum duration of vibration (1 s) and the maximum amplitude (1.45 g) were set in advance. After the designer’s decisions on lower bounds came the selection of the number of levels for vibration duration, \(N\), and amplitudes, \(M\). The result was \(N\) duration levels, equally spaced between the minimum and maximum, and \(M\) levels of amplitude, distributed analogously. These conditions yield \(N \times M\) distinct possible vibration cues in total. Two objectives were taken into account: the information-transfer rate (IT) represents the channel’s capacity, and accuracy was judged in terms of recognition difficulty.

**The Background to Tactile Icons**

Investigating and optimizing ways of transmitting information via skin sensations has been an important enduring goal for haptics researchers, and it gained still greater relevance with the emergence of smartwatches [74, 116, 130, 147]. Prior work has investigated generating vibrations with various durations, frequencies, and amplitudes with a single vibration tactor [209, 211, 211, 27]; transmission of spatial patterns by means of multiple tactors [130, 128, 133]; and the use of motor-driven skin-drag displays for continuous spatial patterns [96].

These efforts notwithstanding, scholarship has produced only extremely limited conclusive guidance as to what constitutes the “best” design for a given context of use [208]. One critical reason for this lack of clarity is that optimizing a tactile display depends not on one performance metric but on several. The goal of most studies has been to minimize recognition error and maximize IT. These are mutually conflicting objectives, yet prior research has not effectively accounted for multiple objectives. According to information theory, there must be an increase in entropy – corresponding to a large set of possible haptic cues – if the rate of information transfer is to increase; however, a larger number of stimuli also leads to lower recognition accuracy, as noted above. In the absence of a principled algorithm for efficiently seeking Pareto optima, the methods available usually consider only one objective or involve manually eliminating non-suitable designs [140, 37, 41, 133]. Because of the time and effort involved in conducting empirical experiments, most efforts have curtailed the exploration of design alternatives after a handful of iterations. Given these “budget constraints,” Bayesian optimization offers a clear advantage in facilitating
Table 4.1. Design parameterization for the haptic display.

<table>
<thead>
<tr>
<th>Design parameter</th>
<th>Range or set</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$: Minimum duration of vibration</td>
<td>[50 ms, 950 ms]</td>
</tr>
<tr>
<td>$x_2$: Number of discrete vibration-duration levels</td>
<td>{1,2,3,4}</td>
</tr>
<tr>
<td>$x_3$: Minimum amplitude of vibration</td>
<td>[0 g, 1.45 g]</td>
</tr>
<tr>
<td>$x_4$: Number of discrete amplitude levels</td>
<td>{1,2,3,4}</td>
</tr>
</tbody>
</table>

Design Parameters

The ranges for the four design parameters adjusted in the interface study (the minimum duration, number of duration levels, minimum amplitude, and number of amplitude levels) are listed in Table 4.1. Again, the discrete nature of the two parameters for the number of levels ($x_2$ and $x_4$) is worthy of note: the intervals were assigned via even segmenting of the continuous space for the specified number of levels.

Objective Functions: Information Transfer and Recognition Accuracy

Information transfer ($IT$) represents an estimate of the channel’s capacity to communicate with a given set of stimuli – i.e., the information, in bits, successfully transferred per stimulus. A standard way to measure and compute the IT value for a certain communication channel is to conduct an absolute identification study. For details of IT computation, I refer the reader to section II A of the paper by Tan et al. on this topic [208]. While $IT$ is a general metric for the effectiveness of a communication channel, one should take recognition accuracy into account also, to mitigate error effects. Recognition accuracy over a set of stimuli is calculated as $Accuracy = \frac{n_{correct}}{n}$, where $n_{correct}$ is the number of correct trials (the user’s response is matched with the stimulus); $n$ is the total number of trials.

4.2.2 The Workshop

I conducted the workshop to understand how the MOBO-assisted procedure compares to a traditional one for manual design optimization. The eight participants in the workshop all were interaction designers with industrial-design experience. Each designer was assigned two other participants, who acted in the role of users when the designer needed to test designs. Figure 4.2 illustrates the MOBO framework that was employed to assist in the design process. For clarity’s sake, my description below refers to the participants as “designers” when they acted in that capacity and as “proxy users” when they were testing another participant’s designs.
The Experiment Design

The experiment is a within-subject design with one independent variable. There were two conditions: the design procedure supported by MOBO, which is referred to here as the MOBO-assisted procedure, and the procedure of the designer’s choice, which I refer to as the designer-led procedure. For the experiment, I counterbalanced these two conditions.

The Apparatus and Prototype

The team built a 3D-printed smartwatch prototype (4 × 4 × 0.5 cm) with a single vibration motor, a Precision Microdrives 310-113 unit, as shown in Figure 4.3a. The user interface, depicted in the figure’s pane c, was developed in the application Processing. The interface is composed of a grid of boxes, where each box represents a distinct vibration cue corresponding to a given amplitude and duration. In this interface, the boxes are arranged in accordance with amplitude and duration such that there are $x_2$ columns and $x_4$ rows.

Procedure

The designers were each provided with a prototype smartwatch (see Figure 4.3a) and two proxy users. The study setup was as shown in Figure 4.3b. In the procedure for the users’ identification task, the interface (shown in Figure 4.3c) displays the vibration designs to the proxy users in each iteration, with each box in the interface representing a unique vibration pattern. These vibration designs are generated by either the designer or MOBO. Each iteration starts with “practice mode,” in which the proxy users are presented with a unique vibration cue at random and the corresponding box in the interface is marked in red. Practice mode was designed to familiarize the proxy users with the vibration set in question. It is followed by “identification mode,” an evaluation setting in which each vibration cue
is displayed to the proxy user, who then has to identify it. A span of three hours was assigned for each condition.

The designer-led procedure: The designers were asked to assign values directly for the four design parameters and present the resulting vibration set to the proxy users. In each design iteration, when the work in identification mode (after practice mode as presented above) was completed, the designer viewed the achieved recognition accuracy and the number of cues. After the three hours assigned to the designers had elapsed, they were asked to determine one final design.

The MOBO-assisted procedure: The MOBO implementation utilized was based on the method proposed by Shah et al. [198]. Designers configured the MOBO, then commenced the human-in-the-loop optimization involving the two proxy users. In this procedure too, the designers had three hours for completing the task. They had to select one final design from the Pareto frontier as the final outcome.

After the experiment (i.e., the tasks in both conditions), the designers were presented with the designs they had derived and the corresponding user-performance results for each. To understand their experience, the designers were asked to fill in the NASA-TLX and the System Usability Scale (SUS) questionnaire.

4.2.3 Results

Both the final designs and the user-performance levels achieved proved very similar between the designer-led procedure and the MOBO-assisted one. The mean values of the recognition accuracy and the number of distinct vibration cues arrived at via the manual procedure were, 0.863 ($SD = 0.078$) and 6.125 ($SD = 1.36$), respectively. The corresponding performances for the MOBO-assisted procedure were 0.883 ($SD = 0.08$) and 6.125 ($SD = 2.031$), respectively.

Analysis of the Design Strategies in the Designer-Led Procedure

I noticed that the designers created various strategies to tackle the design problem in the designer-led procedure. This is indicative of the challenge, complexity, and higher mental load in this condition.

Strategy 1 – divide-and-conquer and then an increase in the vibration cues’ complexity: Two designers undertook a divide-and-conquer approach by only tuning certain design parameters as the first step. Afterward, these designers gradually increased the complexity of the vibrations (i.e., with additional unique vibrations) in the course of the optimization process until both metrics met their expectations.

Strategy 2 – divide-and-conquer followed by decreasing the complexity of the vibration cues: In contrast, two of the designers applied a divide-and-conquer strategy similar to strategy 1 but then gradually
decreased the complexity rather than increasing it.

**Strategy 3 – divide-and-conquer and local search:** Two designers identified a reasonable starting design via testing with the proxy users. These reasonably set initial values were fairly close to their final designs. The designers then performed a “local search” strategy in which they slowly fine-tuned the design parameters until they had identified a set of satisfactory cues.

**Strategy 4 – a self-evaluation approach:** One designer evaluated the designs in a largely independent manner at first. After an hour of self-testing, this designer narrowed the field to three design candidates. The designer then invited the two proxy users to evaluate these candidates and selected the final one.

**Strategy 5 – a focus group:** The last designer's work involved a small “focus group.” This designer invited both proxy users to spend five minutes creating their preferred designs independently. Then, each of the three evaluated all of the designs created by the others. The group then jointly discussed the approach to improving the design, then embarked on another round of iteration and evaluation use. Afterward, the group selected two final design instances.

**Analysis of Usability and Workload**

Wilcoxon signed-ranked testing was applied to analyze overall usability and workload on the basis of the questionnaires’ results. Figure 4.4 shows the results for the individual questions. The mean SUS score was 54.375 (sd = 15.51) for the designer-led procedure and 64.375 (sd = 13.48) for the MOBO-assisted procedure. A Wilcoxon signed-rank test showed no statistically significant difference in overall usability between the two conditions (Z = -1.022, p > 0.05). Still, significant differences were found in the responses to specific questions. Differences were particularly evident for item 1 (Z = -2.06, p < 0.05) and item 4 (Z = -2.97, p < 0.05). This suggests that the designers felt both that they would like to work with MOBO more (item 1) and that they would need technical support in the meantime when designing with the optimizer (item 4).

Figure 4.5 shows the results for individual questions in the NASA-TLX questionnaire. The mean workload rating for the manual procedure was 62.67 (sd = 16.36), and it was 45.17 (sd = 12.4) for the MOBO-assisted procedure. A Wilcoxon signed-rank test revealed a significant difference between the workloads perceived for the two conditions (Z = -2.38, p < 0.05), indicating that the MOBO-assisted procedure reduced the overall workload. Examining the individual questions showed that MOBO decreased the task’s mental demands (Z = -2.197, p < 0.05), physical demands (Z = -2.06, p < 0.05), temporal requirements (Z = -2.366, p < 0.05), and frustration (Z = -2.527, p < 0.05) specifically.
Qualitative Analysis

While the MOBO-assisted procedure and designer-led procedure generally reached similar outcomes, MOBO’s assistance significantly reduced the designer’s overall effort (mental and physical). The qualitative analysis pinpointed explanations for this.

With the designer-led procedure, designers must carefully create a strategy for tackling the design problem. Doing so requires extra mental effort. In addition, the designers had to expend time on “interpreting” and “making sense of” the results from each iteration and then selecting the next set of vibration cues. The MOBO reduced the designer effort required for determining the next design instance. Designers’ responses in the interviews led to the same conclusions. One designer (D4) stated that the designer-led search “can go on forever. I can always change something and lead to a different performance. I always feel uncertain, not knowing if this change will improve [things] or not, and this is frustrating. Also, because I
need to deliver a design within a certain amount of time, so I was somehow stressed.” Along similar lines, D1 shared that “I was not sure whether this design is good enough, so I felt it to be more temporally demanding. On the other hand, when using [MOBO], I simply needed to assign one hour [...] to each participant and collected the results. It is much simpler and relaxing.”

From the user standpoint, all designers pointed out the benefits of having the derived Pareto-optimal designs. One (D1) observed, “If I changed my weights of the objectives and wanted to search for another design, I might need to invest another 30 minutes to reach that point. [The MOBO procedure’s output visualization] showed all the designs along the line (Pareto front) and I could just pick one from them. From this perspective, I find [the MOBO procedure] much more efficient because it searches not just one final outcome but multiple.” Going into greater depth, D5 explained, “I set some kind of priority at the beginning of the design. For example, the recognition rate is more important than the information transfer, and I want to achieve 95% accuracy. However, during the [manual] process, I might gain new knowledge about the interaction and would like to change the weight of the two objectives, which would force me to change the direction of the search. The [MOBO procedure] can avoid this kind of a hassle because it explores all the directions and provides all the possibilities.”

The Workshop Overall

The workshop component of the research contributed to answering CQ 2. Two major findings emerged from the qualitative data: Firstly, MOBO reduced the effort of searching and of proposing the next design candidate. Secondly, the Pareto frontier gave the designer more flexibility to determine a final design. In contrast, the designer-led procedure requires the designer to have an implicit direction (apply some sort of weight) during the search process. Such internal weighting may change as the process progresses, thus necessitating more effort.

4.3 Investigating Performance in Multi-Objective HILO Conditions

Though the workshop-based research demonstrated that the MOBO effectively reduced the workload for designers, it did not fully address CQ 2, particularly with regard to comparing the ultimate user performance yielded by MOBO with that obtained through traditional approaches. Also, the first study did not investigate how MOBO affects creative activities in the design process. Neither did it demonstrate how search behaviors diverge between MOBO and designer-led optimization procedures. Therefore, a user study was designed with the specific aim of addressing these
4.3.1 3D Touch Interaction

The study involved a complex interaction, 3D touch, which encompasses parameters related to input and haptic output both (see Figure 4.6). In virtual-reality (VR) or augmented-reality (AR) applications, interactive objects (or targets) are often placed beyond the range within immediate reach of the physical hand. For instance, a user who is browsing a virtual living room might need access to an object situated on the other side of the room. To enable selecting targets at various distances, one can employ a transfer function to translate the physical hand’s position to a cursor position. This concept is widely applied in 2D interaction; for example, the transfer function of a mouse maps hand position to the location of the cursor on the monitor. For this study, I chose the Go-Go technique, a classic mechanism intended to address precisely this problem, as the base interaction. To bring the problem even closer to off-the-shelf devices, I added a vibrotactile motor supplying tactile feedback. Also, two further parameters were introduced, to control the vibration feedback.

The Background to 3D Touch Interaction

Pointing at a target is an essential and ubiquitous interaction in any VR or AR interface [7]. Hence, extensive research has explored a wide range of pointing and selection methods for VR and AR [25, 180]. As any input interaction should, well-designed 3D selection affords action that is both fast and accurate. Research attests that there is a vast range of control-to-display transfer functions, all with slight differences in their design space [7, 179, 69, 150, 118]. Since searching a high-dimensional design space while evaluating the user’s performance is complex and challenging, prior work has relied either on a tremendous amount of trial-and-error [35] or on heuristics [160, 232]. As for the technique chosen for this study, within certain bounds the Go-Go technique [179] follows a 1-to-1 linear mapping, so the virtual hand (i.e., cursor in virtual reality) moves linearly with the physical hand’s movements. Exceeding the given threshold, the technique follows a nonlinear mapping; the virtual hand’s motion away is scaled quadratically from the physical movements. This mechanism lets users stably select the targets that are relatively close to the body yet also enables reaching those targets that are farther away. The Go-Go technique operates with two parameters.

Design Parameters

The Go-Go technique’s two parameters are $D$ and $k$. The first of these is the threshold for ranges between the linear and nonlinear parts of the transfer function. When the physical hand is within the range $D$, the
From a Single Objective to Multiple Objectives

Figure 4.6. A diagram of the 3D touch interaction. (a) illustrates the cursor position when the length of the real arm vector is less than the threshold distance D. (b) showcases the cursor position when the real arm vector is beyond the threshold distance D. (c) shows the vibration feedback. More details are given in Publication II.

The transfer function is linear; otherwise, it is quadratic. Secondly, \( k \) dictates the scale of the quadratic transfer function. The paper introducing the technique provides a detailed explanation [179]. I retained \( D \) and \( k \) as the design parameters for the study’s 3D touch interaction. In light of the pilot-study results, the parameters were set in these ranges: \( D \in [0, 1] \) and \( k \in [0, 0.5] \).

In addition to the transfer function, I introduced tactile feedback to the interaction. To enhance user performance, I strove to offer a tactile signal when the target was reached. Hence, there are two parameters for vibrotactile feedback: the activation–vibration distance, denoted as \( G \), and the vibration amplitude, or \( A \), as shown in Figure 4.6 (pane c). To avoid further complications, I chose a fixed vibration duration of 300 ms. On the basis of the pilot study’s results, I set the ranges of these two parameters to \( G \in [15\, \text{cm}, -5\, \text{cm}] \) (15 cm before and 5 cm after touching a target) and \( A \in [2.6\, \text{g}, 0\, \text{g}] \).

**Objective Functions**

Similarly to those for general input devices, the two objective functions considered here are connected with efficiency and accuracy. The first addresses completion time, here the average time between the moment at which the user starts moving the cursor and when the selection action is complete. The second objective involves spatial error; the function refers to the maximum overshoot range.

For a roughly commensurate range between these two objective functions, I converted the aforementioned measurements into two metrics: speed and accuracy. In essence, the completion times and spatial error were normalized linearly to speed and accuracy. I transformed the completion-time range \([1,600\, \text{ms}, 900\, \text{ms}]\) into speed range \([-1, 1]\), and I transformed the spatial error range \([1\, \text{cm}, 0\, \text{cm}]\) into the accuracy range \([-1, 1]\). After normalization, 1 is the best value and -1 denotes the worst performance; hence, the problem has become a case of maximization.
4.3.2 The User Study

As the workshop described above was, this study was carried out to compare two conditions. Where the second study differs markedly from the aforementioned workshop with the MOBO-driven procedure and the designer-led procedure is that there was not a separate role of designer in this design. Instead, the designers themselves were the designs’ users (see Figure 4.7). They had to fine-tune the design parameters for maximizing their own performance. In the optimization-led procedure, the study participants engaged in HILO; that is, the optimizer chose how to set the design parameters in each iteration. After the optimization procedure was completed, the participant was presented with a set of Pareto-optimal designs. In the manual procedure’s implementation, search was driven by manual exploration; the participant had to decide on the next design candidate, unaided.

The Experiment Design

The study employed a between-subjects design. There was one independent variable with two conditions, involving the optimization procedures introduced above. Each participant was assigned to either the group for the designer-led condition or that for the optimization-driven one. Three sets of measurements were gathered: user performance (completion time and spatial error), perceived creativity (gauged via the Creativity Support Index, or CSI [43]) and workload (probed with the NASA-TLX instrument [85]), and the results of search-behavior analysis (for search distance and...
From a Single Objective to Multiple Objectives

hypercubes visited).

The Apparatus and Prototype
The 3D touch interaction was implemented with Unity 3D\(^3\) and deployed on the Oculus Quest 2\(^4\). For the vibration motor added to the system,\(^5\) the vibration was driven by a DRV2605L driver board powered by an Arduino Uno microprocessor. The target arrangement primarily followed the lines of previous Fitts’ law tasks [38].

To support the participants in the designer-led process, I provided them with a panel of parameter sliders, with which they could easily and intuitively tune the parameters’ values. The new value was applied to the interaction immediately, so the participants could try it out without any lag. If they reached a design that they deemed worthy of formal evaluation, they could press a button labeled “evaluation.” Doing this initiated a full evaluation of the design, consisting of 36 selections. Once formal evaluation was complete, two charts were shown to the participant to report on the resulting performance.

Procedure
Members of the designer-led group were instructed to tune the design manually and told that they needed to arrive at three optimal designs with the tools offered by the research team. Participants who were instead exposed to the optimization-led condition worked with an optimizer. After the optimization was complete, these participants likewise were asked to pick three final designs from among the Pareto-optimal designs presented. Finally, I evaluated the final performance of the designs developed, in a separate session.

4.3.3 Results
The analysis of the results comprised three elements: examining user performance, assessing the subjective ratings for the experience, and comparing search behavior between the two conditions.

User Performance
As Figure 4.8’s pane a shows, for the designer-led and optimization-driven procedure, respectively, the mean completion times were 1,120 ms (sd = 119.4) and 1,185 ms (sd = 97.2), and the mean values for spatial error were 2.2 cm (sd = 1.2) and 1.5 cm (sd = 0.7). From \(t\)-tests performed on completion-time and spatial-error data, a significant difference in spatial error \((t(38) = 2.237, p < 0.05)\) became evident, indicating that the

\(^3\) Further information available at https://unity.com/.
\(^5\) The model is presented at https://www.precisionmicrodrives.com/product/310-117-10mm-vibration-motor-3mm-type.
**Experience and Workload**

Figure 4.8, pane b, shows the perceived satisfaction, confidence, agency, and ownership for the two optimization procedures. Further analysis performed with a Mann–Whitney U-test revealed significant differences in the Agency ($t(38) = -5.523, p < 0.001$) and Ownership ($t(38) = -3.892, p < 0.001$) factors. To analyze the level of perceived creativity support further, I examined the responses to the CSI questionnaire. I found a significant difference in the overall CSI score ($t(38) = -2.503, p < 0.05$). For each question, a statistically significant difference in the Expressiveness factor was evident ($t(38) = -3.222, p < 0.001$).

Examining the workload perceived by the participants as probed via the NASA-TLX questionnaire, I found significant differences between the two approaches in both Mental Demand and Effort metrics ($p < 0.05$ for both scales). This sense that the designer-led procedure imposed a greater mental burden is consistent with the takeaway from the workshop discussed in the preceding section of the chapter.

**Search Behavior**

Lastly, I analyzed the search behavior within the two procedures. The designer-led group took, on average, 51.8 minutes ($sd = 10.0$) to complete the task, while the equivalent figure for the optimizer-driven procedure is 78.0 minutes ($sd = 6.3$). Furthermore, the designer-led procedure included visits to 271 distinct designs, on average ($sd = 192.4$). In all, the participants exposed to that condition tried out 259 designs ($sd = 194.5$) and formally evaluated 12.5 ($sd = 5.5$). Meanwhile, the optimizer-driven procedure visited only 40 designs. It is especially interesting that the designer-led procedure consumed less time for the full process yet visited 75
6.7 times more design points than the optimizer-led procedure did. This finding demonstrates that humans have a mechanism for rapidly distinguishing between a design that merits full-fledged evaluation and one that does not.

The final quantitative assessment examined how human designers and the optimization system explored the design space, by calculating how many hypercubes were visited. The design space for this case was $[0,1]^4$. We can split each dimension evenly into $m$ levels by means of a division parameter $m$. This separates the space into $m^4$ hypercubes of equal size.

I defined a hypercube as “visited” if any set of design parameters fell within the range of said hypercube’s bounds. The team conducted $t$-tests to check for inter-condition differences in the number of hypercubes visited. Analysis identified significant differences for both $m = 2$ ($p = 0.0001$) and $m = 3$ ($p = 0.0019$) segments. This result indicates that the optimization-driven procedure consistently explored more areas of the design space than the designer-led process. This is one explanation for the optimizer achieving better performance; the human designers were not able to explore the design space with as much breadth as the optimizer, so some optimal design(s) may have gone unexplored.

**Qualitative Analysis**

Six designers stated that they would like to have more agency and the opportunity to express their ideas when designing with the optimizer, especially in cases of disagreement with the design suggestions generated by it. As one participant explained, “I knew what I wanted. I wanted the gap [value] to be reduced, but the AI didn’t give me that design.” Another participant offered this suggestion: “I wish I can just tell the AI I don’t like it [the design].” With similar comments, a designer highlighted the frustration of dealing with “bad” designs proposed by the optimizer – an evaluation “trying out a design that I knew wouldn’t work is a waste of time.” Generally, the participants were not satisfied with the diminished sense of ownership they experienced in the optimization-driven process. They described feeling as if they had been “working for the AI on those trials,” having felt “bored,” and finding that this was “not intellectual work.”

Irrespective of the shortcomings, optimization did reduce the participant effort consumed in the design process. Participants cited the difficulty of optimizing this interaction themselves, offering as examples “the need to figure out how each parameter works” and “trying to further increase the performance.” They also reflected on the challenge of handling two objectives, with comments such as “in fine-tuning, I tended to work on reducing completion time more than spatial errors.” Participants assisted by the optimizer felt that they had expended significantly less mental effort: “I feel relaxed as the AI is doing the design part.”
4.4 The Work in Summary

In response to previous human-in-the-loop optimization work, which focused mainly on tasks with only one objective, this chapter lays out how to extend HILO from a single objective to multiple objectives. To this end, I raised and tackled two questions: how to enable multi-objective HILO (CQ 1) and what qualities characterize performing multi-objective HILO (CQ 2).

To respond to the former, I proposed applying Pareto-front learning to achieve multi-objective HILO. Instead of searching for a single design that yields the best user performance, Pareto-front learning identifies a series of designs that lead to the Pareto-optimal performance levels, representing the best tradeoffs of multiple objectives. Although publications in the machine-learning field have proposed various implementations of MOBO and these have been evaluated by means of test functions, no prior work has implemented multi-objective optimization in a human-in-the-loop setting. Filling this gap, two of the dissertation’s component publications demonstrate the efficacy of Pareto-front learning: Publication I describes a design workshop for the task of designing a set of vibration cues, which highlighted the designer-perceived workload and the range of design strategies exhibited. Publication II presents an empirical study conducted in the context of designing a 3D touch interaction. That study focused more specifically on examining the ultimate performance, perceived creativity support, and the search patterns.

Together, the two publications answer CQ 2. Analysis showed that the designers taking part experienced significantly less effort and a lower workload with MOBO than with manual (or designer-led) procedures. There are several reasons. Firstly, they did not need to come up with strategies to tackle the complex design challenge. As participants reported, exploring a large design space on their own was mentally taxing. Furthermore, they had to balance multiple objectives, which further complicates matters. In contrast, MOBO offers extensive flexibility for dynamically picking a final design on the Pareto frontier. Performance analysis revealed that MOBO yielded performance at least comparable to that obtained by means of the designer-led procedure. This can be attributed to MOBO’s ability to search a wider range of areas in the design space in a principled manner, thereby generating a greater variety of designs.

However, the workshop and follow-on study also illuminated the designer-led procedure’s greater sense of agency and ownership, stemming from the designer’s direct control over the exploration; participants articulated greater perceived expressiveness too. Participating designers recommended a mixed design method in which the human designer could rely on optimization for effectively exploring the design space but take control as necessary (e.g., skipping bad designs). I would encourage future
work exploring this direction. It is worth stressing the widespread applicability that such advances afford: the avenues presented are suitable, for instance, for single-objective cases as well as the multi-objective scenarios examined here.

The two publications described in this chapter jointly inform the dissertation’s first and second contribution: multi-objective human-in-the-loop design optimization via Pareto-frontier learning and investigation of the perceived qualities of multi-objective HILO, respectively.
5. From Individual to Population

“Design must always consider the needs of the mass, the majority, the group - not just the individual - for it is in satisfying the needs of the group that the needs of the individual are satisfied.” — Don Norman

Following on from the introduction of the method enabling multi-objective HILO, this chapter speaks to the next aim: extending HILO to reach optimization beyond the individual user. In design practice, designers often seek to address the needs of a group or population rather than an individual user. Two major issues arise in shifting from optimization for individuals to group-level optimization.

The first of them is transferability: Can a design optimized for a single user carry over to other users? This depends greatly on the interaction and on behavioral diversity across users. If all users show extensive similarity in their preferences and performance, the optimized design may be transferable; that is, a design that is “good” for one user should work well for others. However, if there are considerable variations among users, the optimal designs for two users are likely to be different. In that case, the optimal design cannot be transferred non-problematically, and a design that excels for one person might serve another user poorly. The methods encountered thus far in the field have not been able to address this problem. Therefore, there is a need to identify and hone a method able to generate design instances that are optimal for a group/population – i.e., the best-compromise designs for all individuals involved.

Another issue that has to be addressed is the time-efficiency of the optimization process. Previous studies have shown that a full-fledged optimization procedure can take an hour or even more to identify the Pareto-optimal designs for an individual. The issue becomes even more pressing in cases of design tasks that involve larger numbers of parameters or objectives, which require more iterations and time before convergence. Such lengthy optimization procedures are impractical for end users, who cannot afford to spend an extended period of time on finding an optimal setting before starting to use the interaction technique. The reason for
the protracted nature of the optimization is that BO’s learning typically does not utilize any prior information. One possible way to boost efficiency is to take advantage of previously gathered optimization data to form an informative prior. The practicality of this suggestion has remained unclear, though, with its further investigation posing a fundamental challenge. Therefore, the research presented in this chapter pursued a time-efficient optimization method that exploits group-level data.

With this chapter, I aim to answer two central questions arising from the issues described above.

**CQ 1: How could we derive the optimized design for a group of users?**

**CQ 2: How could we generate an initial model that can efficiently adapt to an individual user by making use of group-level optimization data?**

### 5.1 Group-Level Optimization

My response to these questions takes the form of a novel approach referred to as group-level human-in-the-loop optimization. Group-level optimization differs from typical HILO in that the optimization process does not simply consider an individual user’s data; it works with data from a group of users. To achieve this, I have introduced two extensions to the HILO framework. The first of these, **Global GP**, involves a model of aggregated group-level data that can be inspected to identify Pareto-optimal designs for the whole group. The second extension is **Warm-Start GP**. Its model relies on a set of optimization data for initialization that affords efficient design customization for new users. Table 5.1 presents a fuller comparison between standard BO and my group-level approach.

#### 5.1.1 Procedure of Group-Level Optimization

As given earlier, the central idea for these two extensions is to aggregate the data gathered from the optimization of a population user group. Different from a standard HILO where an individual user simply goes through an optimization process on their own, group-level HILO has a different procedure:

1. Running individual optimizations on a group of users;

2. Constructing the group-level models (the Global GP or the Warm-Start GP). If the goal is group-level optimization, use the Global GP to generate the group-level optimal designs.
3. Deploying the derived design (Global GP) or the derived Wram-start GP on the new users.

The proposed group-level HILO extensions support both single-objective and multi-objective purposes. However, it is required to be consistent throughout the steps. The group-level surrogate models should share the same number of objective functions as in the deployment step. In the following content, I only focus on multi-objective cases.

5.1.2 The Global GP: Deriving Group-Level Optimal Designs

The first extension, Global GP, is intended to address CQ 1, which involves deriving the Pareto-optimal designs for a group of users. Once I have constructed a global model, supplying all observed design parameters and objective values from all participants so as to form a single GP model, we can query that GP model to obtain the predicted mean and variance of a given design. This prediction can be viewed as the group-level expected performance for a given design instance. After constructing the model, I derive the optimal design instances by making predictions via the group-level surrogate model. To identify the Pareto-optimal designs generated by Global GP, I conduct a fine-grid search of the entire design space. Assigning \( c \) equally spaced points along each dimension gives us \( c^n \) points in total. We can query Global GP for all of these \( c^n \) points and store all the output. Finally, I identify the Pareto-optimal designs from among these \( c^n \) predictions. For the experiments reported upon in this chapter, the research team chose \( c = 16 \).

This method is computationally expensive but since it runs as a post-processing step, it is practical and feasible to perform and provides a comprehensive summary of the optimal design parameter sets. Figure 5.1 illustrates the high-level process of constructing the Global GP and extracting the Pareto-optimal designs.
<table>
<thead>
<tr>
<th>Features</th>
<th>Group-level HILO</th>
<th>Group specified in advance</th>
<th>Flexibly defined</th>
<th>User sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model initialization</td>
<td>Learn from scratch</td>
<td>Learn from scratch</td>
<td>Learn from scratch</td>
<td></td>
</tr>
<tr>
<td>Objective</td>
<td>Best design</td>
<td>Best compromise among users (Global GP)</td>
<td>Best design for an individual for a group</td>
<td>Objective</td>
</tr>
<tr>
<td>Model initialization</td>
<td>Learn from scratch</td>
<td>Learn from scratch</td>
<td>Learn from scratch</td>
<td></td>
</tr>
<tr>
<td>Objective</td>
<td>Best design</td>
<td>Best compromise among users (Global GP)</td>
<td>Best design for an individual for a group</td>
<td>Objective</td>
</tr>
</tbody>
</table>

Table 5.1. The differences between standard Bayesian optimization and Group-in-the-Loop optimization.
5.1.3 Warm-Start GP: A Rapidly Adapting Surrogate Model

The other novel extension responds to CQ 2, which pertains to the time-efficiency of HILO. Standard HILO usually begins without any informative prior distribution. This leads to random exploration of design options in the early stages, sometimes called the “cold-start” problem. To overcome this issue, I explored creating a “warm-start” GP model to serve the initial stages. We can implement the core principle (i.e., exploiting previously collected optimization data to form an informative distribution that captures the interaction characteristics) in practice by carefully selecting a subset of data points from the individual-level optimization processes and fitting them to our Warm-Start GP model. This SM can provide useful prior knowledge for efficient optimization in subsequent individual-level optimization processes. Figure 5.2 illustrates the general procedure of using Warm-Start GP in HILO.

For the most part, the extension follows the steps implemented in sparse GP. We adapted the method proposed by Titsias [212]; they applied an approximation of the marginal likelihood for the entire dataset with a subset of size $K$ in accordance with the work of Seeger et al. [196]. The main idea here is to pick the most “informative” $K$ data points by adding one training point from the full dataset at a time with the aim of maximizing the approximate marginal likelihood over the complete dataset. To reduce the computation complexity, some heuristics are introduced: Instead of directly including the full dataset as initial candidates for the sparse subset, the procedure considers only a reduced subset of candidates; such a subset is sampled randomly. In our application (detailed further along), that randomly selected subset had 100 data points; i.e., it was half the size of the full 200-data-point set. Within this reduced dataset, the likelihood-maximization process functions for greedy selection of the candidate point for the sparse subset of size $K$, which was set to be 5.

In summary, our method is designed to generate a warm-start prior that can adapt to a new user and, thereby, obtain personalized optimal design instances sample-efficiently. Our method follows sparse Gaussian Processes [17, 34, 32] to obtain the representative data points from the full dataset.
5.2 Evaluating Group-Level Optimization via Simulations

While our methods provide a means to obtain group-level optimal designs and initializations, they assume the users in a group to have some shared characteristics. Different levels of similarities within the group affect the performances of the proposed methods. Before conducting the experiment with a human-in-the-loop setup, I validate the Global GP’s and the Warm-Start GP’s performances when facing various diversity levels using simulations. Furthermore, I compare these performances to the conditions that are without our enhancements. The results showed that when the functions within a group are relatively similar, our methods can effectively achieve higher objective performance and more efficient adaptation. Even when the functions are quite diverse, our methods still provide marginal benefits or comparable performances to the baseline.

5.2.1 Test Functions

All the following simulations share a base function; I shifted the base function differently to simulate different users. The base function has four parameters and two objective functions. We adopted two widely used test functions: Branin\(^1\) (2-dimensional input) and Sphere\(^2\) (3-dimensional input) as our objective functions. The first two parameters \((x_1, x_2)\) contribute to the value of the Branin function, while the last three parameters \((x_2, x_3, x_4)\) contribute to the value of the Sphere function. Note that parameter \(x_2\) contributes to both the Branin and the Sphere values. This was deliberately set to create a trade-off scenario: there is not an \(x_2\) value that can maximize two functions at the same time, and hence, MOBO must search for the Pareto frontier rather than an optimal design. We normalized these two functions so that all the parameter values are bounded in the range of \([0, 5]\) and all the objective values are bounded in the range of \([-1.1, 1.1]\). Further details of the parameters and the objective functions can be found in Table 5.2 and Table 5.3.

---

\(^1\)https://www.sfu.ca/~ssurjano/branin.html

\(^2\)https://www.sfu.ca/~ssurjano/spheref.html
### Table 5.3. The simulation task’s objective functions and their ranges.

<table>
<thead>
<tr>
<th>Objective functions</th>
<th>Name</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_1$</td>
<td>Branin</td>
<td>[-1.1, 1]</td>
</tr>
<tr>
<td>$y_2$</td>
<td>Sphere</td>
<td>[-1.1, 1.1]</td>
</tr>
</tbody>
</table>

#### 5.2.2 Setting up Simulations

To simulate the differences between the users, I need to generate a set of functions that have different Pareto-optimal parameter values. We achieve that by randomly shifting the base function. Each shifted function can be seen as a unique user because it has its own set of Pareto-optimal design parameter values. The shifting happens in all the parameters ($x_1$, $x_2$, $x_3$, $x_4$), and the amount of the shifting is sampled from a uniform distribution of certain sample ranges. Specifically, the shifting is denoted as $x'_n = x_n + \delta$. $x_n$ is the original parameter value, $x'_n$ is the new parameter value after shifting, $n \in [1,4]$, and $\delta \sim U(-\frac{\text{range}_2}{2}, \frac{\text{range}_2}{2})$. If the sample range is big, the shifts of this group will be farther away from each other and result in more diverse functions. If the range is small, the resulting shifts and functions will be more similar.

We then simulate various groups of users by varying the sample ranges. We have 5 groups of functions (users), whose sample ranges are 0.1, 1, 2, 3, and 4, respectively. The smallest sample range (0.1) naturally results in a group of functions that are highly similar to each other. The largest sample range (4) leads to a group of high-diversity functions.

For each sample range (i.e., 0.1, 1, 2, 3, and 4), I randomly generated 20 unique functions (users). 10 of them were used to run individual MOBO, and the optimization data was used to generate the Global GP and the Warm-Start GP; I named this group of functions “the GP groups”. The remaining 10 functions (users) are then used to evaluate the Global GP and Warm-Start GP; I named this group “the evaluation groups”. For running the individual optimization, MOBO is set to have 3 initial random samplings followed by 10 optimization iterations.

To evaluate the Global GP, I first constructed the Global GP, and then performed a grid-search over the whole design space to identify the design that has the highest average of the two objectives. We evaluated the obtained parameter sets within the evaluation groups. Then, I created a baseline group, which determined the parameter values randomly.

To evaluate the Warm-Start GP, I derive 10 sparse GP points from the GP groups’ data. Then I ran 5 iterations on the evaluation groups with the Warm-Start GP and recorded the hypervolume. The baseline group is running MOBO from scratch with 2 initial samplings and 3 optimization iterations.
5.2.3 Simulation Results

The simulation results are shown in Figure 5.3. Both Global GP and Warm-Start GP resulted in promising performances when the sample ranges are small (e.g., groups 0.1, 1, and 2). This indicates that our methods would bring advantages to the users if the group has a certain level of similarity among users, which is similar to our findings in the following user studies. Even when the groups share only little similarity (e.g., groups 3 and 4), the final performances are slightly better or comparable to the baseline conditions. This suggests that our methods will at least provide marginal benefits or deliver similar performance to standard Bayesian optimization. The results further suggest that group-level optimization can be applied when the users may share certain characteristics.

With these positive simulation results, I then conducted the two studies, as shown in the following sections. These studies showed that both the Global GP and the Warm-Start GP lead to significantly better performances than baselines, which confirms that the users have some shared characteristics in the selected interactions. Further investigation is needed to understand the levels of similarities among users within different interactions. If there is an interaction where the users have no similarities at all, it is naturally impossible to derive a group-optimized design regardless of the methods because such a group-optimized design simply does not exist. Under such cases, only individual optimization will be suitable to address the problem, which requires a longer optimization duration.

5.3 Designing with Global GP

The first user study validated the Global GP method for deriving a set of Pareto-optimal designs that represent a group of users. The study involved three steps: A series of individual-level optimizations was conducted, resulting in observations of <design instance, objective values> pairs. Secondly, a Global GP model was constructed with all the data...
gathered, and a fine-resolution grid search was performed to derive the group-level Pareto-optimal designs. Finally, the Pareto-optimal designs were evaluated with a group of users and compared with the baseline condition. The results showed that the proposed technique produced better user performance.

5.3.1 Group-Level Optimized 3D Touch

The target interaction was the Go-Go technique [179], and the setting of the interaction was the same as that described in subsection 4.3.1.

5.3.2 The User Study

The three steps employed for the user study generally followed the description in subsection 5.1.1. Firstly, I ran individual-level optimizations for all users in a given group. After that, I constructed a Global GP model with the data obtained from those optimizations and, on that basis, derived the group-level Pareto-optimal design. Finally, the group-level optimal designs were evaluated with users. The design parameters and objective function remained the same as outlined in subsection 4.3.1.

Participants
The research team recruited 20 participants for the study and assigned them at random to two groups. The first group participated in every step of the study, and I subsequently labeled it the “experienced” group accordingly. The second group, involved in only the final step, was labeled as the “novice” group.

Apparatus
The apparatus was similar to what is presented in subsection 4.3.1. However, since the participants were not required to select design instances manually, all relevant interfaces related to manual optimization were absent.

Procedure
Step 1 (user-specific optimization): The first step consisted of individual-level optimization for all members of the experienced group. There were 40 iterations in total. Each iteration provided a pair of <design-parameter values, objective-function values>. The full procedure took 90 minutes per participant. From Step 1, 400 data points (10 participants × 40 data points) were gathered.

Step 2 (Global GP and group-level optimal designs): We constructed the Global GP model with all data points gathered in Step 1. Then, a fine-grained grid search over the whole design space was performed. Each design parameter was discretized evenly to 16 points, resulting in $16^4$ de-
sign instances. Global GP made predictions for all $16^4$ instances. Lastly, I derived the set of Pareto-optimal designs within this set of $16^4$ points. The final set of Pareto-optimal global designs is presented in Figure 5.4. These global designs were subdivided into two subsidiary sets: a speed-oriented subset and an accuracy-oriented subset. The speed-oriented designs prioritized efficiency (completion time) over accuracy (spatial errors), with the accuracy-oriented designs’ priorities aligned the other way around. We synthesized the results into two final designs, one speed-oriented and the other accuracy-oriented, by averaging over all parameter values from these subsets.

**Step 3 (evaluation of the group-level optimal designs):** The goal for the final phase of the study was to evaluate the performance of the group-level optimal designs against the baseline Go-Go technique configuration. To investigate the efficacy of the global-level optimal design thoroughly, I evaluated the designs with the aforementioned two groups of users: the experienced group (included in the first step) from which initial data were obtained, and the novice group (completely naïve to the interaction). The evaluation setting was structured as a $3 \times 2$ mixed-design experiment with two independent variables. Each participant tested all of the design instances, so the design factor was within-subject. The participant group was a between-subjects factor.

Note that the Go-Go technique as originally conceived does not feature vibration feedback. For a fair comparison, I augmented the technique by issuing a vibration cue as the user contacts the target with the virtual cursor ($x_3 = 0$ cm). This is a common default setting in many VR interactions. The vibration amplitude for the Go-Go technique ($x_4$) was set to 1 g, the amplitude that was most strongly preferred in pilot testing. With this setting, there are three conditions to be evaluated; Table 5.4 outlines them.
Table 5.4. The three design conditions evaluated in Phase 2.

<table>
<thead>
<tr>
<th>Condition</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed-oriented</td>
<td>0.05</td>
<td>0.098</td>
<td>5.77 cm</td>
<td>2 g</td>
</tr>
<tr>
<td>Accuracy-oriented</td>
<td>0.092</td>
<td>0.037</td>
<td>10.76 cm</td>
<td>0.91 g</td>
</tr>
<tr>
<td>Go-Go Technique</td>
<td>0.667</td>
<td>0.167</td>
<td>0 cm</td>
<td>1 g</td>
</tr>
</tbody>
</table>

Figure 5.5. Results of the comparative study on three designs. The result indicates that the designs generated by the Global GP outperform the Go-Go Technique in completion time. Further, the accuracy-oriented design also outperforms the other designs. Please refer to Publication 3 for further details.

5.3.3 Results

The mean completion time and spatial error for each condition and participant group are presented in Figure 5.5. The experienced users’ mean completion time for the speed-oriented design, the accuracy-oriented design, and the Go-Go technique were 1,038 ms ($sd = 77.64$), 1,081 ms ($sd = 87.72$), and 1,111 ms ($sd = 80.32$), respectively. Novice users’ corresponding mean completion times for the three conditions, respectively, were 1,124 ms ($sd = 126.28$), 1,117 ms ($sd = 127.10$), and 1,167 ms ($sd = 120.88$). As for the mean spatial error, the experienced users’ averages for the speed-oriented design, accuracy-oriented design, and Go-Go technique were 2.34 cm ($sd = 1.76$), 0.97 cm ($sd = 0.49$), and 1.55 cm ($sd = 0.85$), respectively. The equivalent figures for the mean spatial error of the novice users were 2.03 cm ($sd = 1.17$), 0.96 cm ($sd = 0.68$), and 1.83 cm ($sd = 0.78$), respectively.

Mixed-design ANOVA was performed to examine the effect of the interfaces and user-experience levels. Significant within-subject effects were found for both completion time ($F(2,36) = 7.483, p < 0.005$) and spatial errors ($F(1.432,25.781) = 19.284, p < 0.001$). Analysis of the between-subjects effects found no differences connected with user-experience levels (all $p > 0.05$). Examining completion time showed that both the speed-oriented and the accuracy-oriented design outperformed the Go-Go one (all $p < 0.05$). In the analysis of spatial errors, the accuracy-oriented design emerged as significantly better than the other designs (all $p < 0.001$); however, the speed-oriented design did not prove useful for reducing spatial errors. Both group-level optimal designs can be characterized as yielding better or
comparable performance levels.

It is important to note that, in this project, the presented approach aggregated all the previously gathered data points into one unified GP model. Future work should also consider more advanced ways of aggregating data and generating group-level designs. One direction is to store each user's data as separate surrogate models, which will then lead to a group of GP models. Then, for each particular design instance, we can generate a group of predictions, one derived from a single model. With this extension, future work can further derive diverse kinds of global-level optimal designs, such as “best-case” (prioritizing the optimization for the expert user group with the best performance), “worst-case” (prioritizing the optimization for the novice user group with the worst performance), and “average-case” (prioritizing the optimization for the average user group) designs.

5.4 Designing with the Warm-Start GP

Another study was conducted to validate the efficacy of the Warm-Start GP extension by a touch-button design task. Similar to the previous study, there were three major steps. First, I conducted individual optimizations on a group of users. With the gathered observations, I derived the Warm-Start GP via the method described earlier. Finally, I assess the effectiveness of the Warm-Start by comparing it to the standard BO process.

5.4.1 The Fast-Adaptation Touch-Button

The design task involved touch-button-pressing, which is a fundamental mechanism of interaction with touch-sensitive devices. Interacting in this manner involves the user's finger and a button on a touchscreen. Contact triggers a function attached to the button, alongside generation of a key-click vibration to inform the user of the activation. In order to provide context for considering this interaction, the subsections below discuss its design parameterization and target objective functions.

The Background to Touch-Button Interaction

Research has revealed that the optimal point for triggering a button is not at the exact moment when the finger makes contact with the button surface [134, 115] but somewhere within its travel range. Additionally, there has been extensive research into the design of haptics for touch-buttons, various aspects of which have been proven to influence the user's typing speed on a soft keyboard [236, 177]. Designing appropriate haptic feedback for target selection is complex, in that the feedback designs can lead to widely varying perceived sensations and user performance. For instance, the ideal moment for vibration might not be identical to that for
activation (as Figure 5.7 illustrates).

While previous research has demonstrated optimization of button design with a single objective [41, 189, 132, 236], the relevant efforts were not conducted with a human-in-the-loop setup or the results served only one specific objective. In contrast, the aim of this study was to derive a fast-adaptation model of a touch-button for temporal pointing interaction. Temporal pointing tasks require users to provide certain discrete inputs within a narrow time window [127]; for instance, they might have to activate a function at a particular moment in a game or synchronize input experiences in day-to-day use [224].

**Design Parameters**

The study includes five design parameters, shown in Table 5.5 and Figure 5.7. The button-activation threshold ($x_1$) is the force level that activates the touch-button, and the vibration threshold ($x_2$) is the force level that triggers the device to emit the vibration signal. The rest of the parameters were added to create a rich variety of haptic cues. Initial vibration amplitude ($x_3$) and final vibration amplitude ($x_4$) dictate the amplitude of the vibrations, and $x_5$ determines the vibration’s duration. If $x_3$ and $x_4$ are not identical, the vibration is set to linearly increase or decrease over the span of $x_5$. 

![Figure 5.7. Illustrative design example of the target interaction. Please refer to Publication III for more details.](image)
Table 5.5. Design parameterization of the touch-button.

<table>
<thead>
<tr>
<th>Design Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$: Button activation force level</td>
<td>[15 g, 1515 g]</td>
</tr>
<tr>
<td>$x_2$: Vibration activation force level</td>
<td>[15 g, 1515 g]</td>
</tr>
<tr>
<td>$x_3$: Initial vibration amplitude</td>
<td>[0 g, 3.2 g]</td>
</tr>
<tr>
<td>$x_4$: Final vibration amplitude</td>
<td>[0 g, 3.2 g]</td>
</tr>
<tr>
<td>$x_5$: Vibration duration</td>
<td>[0 s, 1.5 s]</td>
</tr>
</tbody>
</table>

Table 5.6. The design objectives of the touch-button.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal Error Mean</td>
<td>The temporal pointing is more accurate if this value is smaller.</td>
</tr>
<tr>
<td>Temporal Error Standard Deviation</td>
<td>The temporal pointing is more precise if this value is smaller.</td>
</tr>
<tr>
<td>Subjective User Rating</td>
<td>The vibration cue matches the click interaction more if this value is higher. Values from 0 to 100.</td>
</tr>
</tbody>
</table>

Objective Functions

Three objective functions considered the temporal performance and the user’s subjective rating. Two separate objective functions were measured regarding the temporal performance: the mean value of temporal errors and the standard deviation of the temporal errors. The participant’s subjective rating was provided by the participants after one iteration was complete. The three objectives are detailed in Table 5.6.

5.4.2 The User Study

With this study, I sought to compare efficacy between Warm-Start GP and the baseline (standard BO), with the design task described above. The results indicate that optimization efficiency indeed improves significantly when Warm-Start GP provides a starting point.

Participants

In total, 22 people were recruited for the study. The participants were randomly allocated to two groups thus: Ten participants completed all phases of the study. In other words, their data were used to generate the Warm-Start GP model, and they also were involved in the final evaluation phase. I refer to this group of users as the “experienced user group.” The remaining 12 participants took part in only the final evaluation, so their data did not inform construction of the Warm-Start GP model. I denote this group as the “novice user group.”
Figure 5.8. A simplified sketch of the study interface during button-pressing (a): the participant is asked to activate the button when the red bullet reaches the yellow target area, at which point the bullet turns blue. After 24 presses, the user is asked to rate the vibration cue (b). Pane c shows the study’s interaction in action.

Apparatus
I implemented a touch-sensitive prototype device (6 cm × 12.5 cm × 1 cm). The most important parts of this prototype were a force-sensing resistor (FSR 402\(^3\)) and a vibration motor (Precision Microdrives 308-102\(^4\)). The device is shown in Figure 5.6. Its vibration motor was controlled by a driver (SparkFun DRV2605L\(^5\)) and an Arduino microprocessor. The study interface, implemented in Processing, is presented in Figure 5.8, panes a and b.

General Tasks
As in the previous study, there were three main steps. The first and the third involved optimizations that required participants’ involvement; they were instructed to perform a temporal pointing task. The leftmost pane of Figure 5.8 depicts the interface. A red “moving bullet” flies from the right side of the interface to the left side, along the bar. Participants were instructed to “activate the button when the bullet reaches the center of the yellow target zone.” The red bullet turns blue when the button is activated, to notify the participant about its status.

In each iteration, participants interacted with the system at two levels of difficulty: “Easy” (with the bullet’s movement speed set to 625 pixels/second) and “Hard” (with a movement rate of 1,000 pixels/second). Users encountered the two difficulty levels in random order. At each difficulty level, the participant performed 12 button presses.

At the end of each iteration, the participants rated the statement “The vibration cue synchronizes (matches) with the button pressing interaction,” which was displayed via the interface. This statement was presented with the scale depicted in Figure 5.8’s pane b: each participant gave a subjective rating from 0 (“Strongly disagree”) to 100 (“Strongly agree”).

\(^3\) Details are available at https://www.interlinkelectronics.com/fsr-402.
\(^5\) See https://www.sparkfun.com/products/14538.
Figure 5.9. The results from the evaluation of Warm-Start GP. More details are provided in Publication III.

Procedure

**Step 1 (user-specific optimization):** Participants were asked to complete the standard MOBO procedure, and Step 2 – generation of the warm-start model – applied the data collected from all these participants: 500 observations (50 design instances × 10 participants).

**Step 2 (deriving the Warm-Start GP model):** The research team applied the Warm-Start GP approach to derive a subset of $K$ data points for forming a Warm-Start initialization model.

The size of $K$ strongly affects the efficacy of the adaptation: if there are too few points, the prior would not be helpful in the later optimization, yet were there too many points to form a prior, initialization would start with a very large initial hypervolume. This is potentially problematic since the data points selected might completely dominate new observations in the next step, in which case there would not be any adaptation at all. To pick a reasonable number of points, I created three warm-start models, with 5, 10, and 15 initial data points, and I then compared their performance by means of a simulation. The results showed that using five Warm-Start data points offers the best setting – it resulted in the greatest hypervolume increase within 15 iterations.

**Step 3 (evaluation of Warm-Start GP):** Both groups of users, the experienced user group and the novice user group, were involved in Step 3 of the study. Both sets of participants were exposed to two conditions, the standard BO and the BO with Warm-Start GP as the initial model. The tasks given to the participants were the same as in Step 1.

5.4.3 Results

Figure 5.9 plots the hypervolume increases produced in both conditions. Two-way repeated-measures ANOVAs were run to analyze the effect of two independent variables – **initialization** (with or without Warm-Start GP)
and iterations – on the hypervolume increase for both user groups.

For the experienced user group, there was no significant interaction between the effect of initialization and that of iterations ($F(14,126) = 0.38$, $p > 0.05$). Simple main-effects analysis showed that the hypervolume was significantly larger with Warm-Start GP than without it ($F(1,9) = 23.43$, $p < 0.001$). This analysis also revealed significant differences between iterations for the experienced user group ($F(14,126) = 31.88$, $p < 0.001$). Pairwise T-tests were conducted to compare the hypervolume between the two initialization conditions at every iteration. Significant differences evident across all iterations (all $p < 0.05$) indicate that Warm-Start GP initialization led to a consistently larger hypervolume.

For the novice user group, a significant interaction between the effect of initialization and of iterations ($F(14,154) = 8.25$, $p < 0.001$) was found. Simple main-effects analysis showed the hypervolume to be significantly larger when the procedure started with Warm-Start GP than without it ($F(1,11) = 19.24$, $p < 0.001$). Also, there was a significant effect for iterations ($F(14,154) = 32.17$, $p < 0.001$). Pairwise T-tests were run to compare the hypervolume between the two initialization conditions at every iteration. Warm-Start GP produced a significantly larger hypervolume from the first to the 11th iteration (all $p < 0.05$). This is evidence that Warm-Start GP effectively supported faster adaptation for the novice user group too.

### 5.5 The Work in Summary

For this chapter, I explored another important direction of HILO, extending it to support group-level optimization. To this end, I specified two important goals: 1) group-level optimal designs and 2) a group-level rapidly adapting the initial model. For reaching these goals, I propose methods that gather optimization data from a group of users and aggregate said data to derive designs or models.

The first method presented, Global GP, involves fitting all observations from user-specific optimizations into a single GP model to form a group-level representative model, after which a grid search is conducted to identify group-level Pareto-optimal design instances. I was able to demonstrate this method’s efficacy for 3D touch interaction, wherein the Pareto-optimal design resulted in better user performance than the baseline condition. The second method, Warm-Start GP, uses a subset of data from user-specific optimizations to create an SM that serves as a more efficient initial model for the optimization process. My evaluation of this method with a touch-button design task showed the hypervolume increase to be significantly greater with Warm-Start GP than in its absence.

These methods extend HILO from single-objective cases to multi-objective cases and expands the scope of optimization from a single user to a group of users.
users. Thereby, they enable HILO for more realistic design problems. That outcome constitutes the third contribution identified for this dissertation, group-level human-in-the-loop optimization.
6. From Physical Prototyping to Emulation

“If a picture is worth a thousand words, a prototype is worth a thousand meetings.”
— IDEO.org

In this chapter, I address another significant challenge of HILO – namely, the high cost of physical prototyping. For a better understanding of this challenge, a review of the typical design processes is in order. Prototyping is a crucial step in most design processes. In other words, prototyping involves transforming a design idea into a physical representation for user testing and evaluation.

I can consider prototyping in terms of two classes of activity: physical prototyping and non-physical prototyping. Many design prototypes do not involve a physical prototype; for instance, one can usually render various design instances for graphical user interfaces in real-time without any physical constructs. The preceding chapters demonstrate how HILO can tackle such applications; e.g., the optimizer can change the properties of the interaction instantly. In contrast, physical prototyping is significantly more costly and time-consuming. Creating a physical prototype involves transforming a design idea into a complete 3D model, then investing time in building the prototype. Soldering the circuit, assembling the digital elements into a working device, etc. requires substantial effort and can take hours to days. On account of the high cost of physical prototyping, it is impractical to conduct HILO for interaction conditions that require fabrication. These interactions have remained within reach only for traditional design processes such as UCD, and design exploration has been limited to a small portion of the design space.

The main research question addressed in this chapter is how to facilitate HILO in design tasks that currently require physical prototyping (CQ 1). To address this challenge, I propose the use of physical emulation, a software and/or hardware system able to simulate the responses of a physical device or system. Physical emulation eliminates the need for fabricating a physical prototype at each iteration, thus significantly reducing the financial outlay and time required for prototyping. I
begin the discussion with background information on physical interaction specifically with buttons, physical prototyping, and emulation in HCI research. Then, taking push-button interaction as an example, I demonstrate how physical emulation can function as a cost-effective tool for HILO.

6.1 Background

Physical Buttons (Push-Buttons)
Physical buttons are devices that translate mechanical force into an electrical signal. This chapter focuses specifically on push-buttons as used in keyboards or key panels. Design parameters such as the physical properties of the keycap (width, angle, and key depth) and the materials used (e.g., plastics) are among the many factors influencing the button’s haptic and tactile characteristics, commonly known as tactility [110, 143]. While tactility is crucial for the typing experience and performance of professional gamers, programmers, typists, and hobbyists alike [131, 2, 49, 174], designing push-buttons can be challenging, because of the high cost of creating the physical prototypes needed in testing their tactility, for which the mechanical structure is vital. This chapter presents physical emulation as a possible solution. It holds potential for enabling human-in-the-loop optimization of push-button design.

Physical Prototyping
Prototyping plays a crucial role in the HCI and design fields, enabling designers to express their ideas, compare design instances, and test hypotheses [221]. Accordingly, research has delved into the functionality of prototypes [90] and the experience of working with them [31], and advances in technology have enabled the emergence of rapid prototyping as a tool for designers’ communication and demonstration of sophisticated concepts [228]. Techniques such as 3D printing [200] and laser cutting [109] have made it possible for designers to fabricate physical forms at a lower cost, while microprocessors and IDEs [6] have made it easier for engineers and developers to work with circuits, sensors, and actuators, thereby facilitating work with fully functional prototypes and even “personal fabrication” [16].

Traditionally, designing a new push-button requires fabrication in full, inclusive of internal circuits and mechanical parts (spring and structure). Despite the relevance of push-buttons in HCI, the discipline has devoted very little effort to physical prototyping of buttons, with one possible reason being the complexity of the prototyping process. The most relevant research in this area has focused on prototyping devices that deliver “passive haptics.” For example, Lin et al. [136] used planar-compliant structures to
create various types of passive haptic feedback when an object is pressed, and metamaterials that alter the structure’s geometry can deliver several sorts of passive feedback and functions [95]. He et al. proposed 3D-printed spring designs for provision of varying passive haptic feedback [87]. It is important to stress that, although the aforementioned research has opened some paths to physically prototyping push-buttons, it remains extremely difficult to replicate all the haptic feedback that physical buttons can offer; the subtle “clicky” tactile feedback is especially hard to generate. Crafting prototypes of push-buttons continues to be a costly process that requires significant expertise in working with 3D/2D modeling, devices for printing or cutting out the parts, etc. These issues highlight the need for emulation.

**Physical Emulation and Haptic Rendering**

“Emulation” generally refers to using a hardware and/or software system to imitate the responses of another system. While it is often addressed in connection with the gaming domain’s software emulation [135, 157], my focus is on physical emulation, in which a system (the emulator) imitates the physical and mechanical responses of various objects/interfaces. There has been extensive research exploring shape-changing interfaces – devices that are able to render different shapes [185]. For instance, inForm [67] is a device able to do this in real time, and inForce [158] is an extension to inForm that can render force feedback with the same form factor.

While prior research has pursued rendering of rich and realistic haptic feedback via emulation devices [106, 14, 181], the body of work on emulating push-buttons remains quite limited. Here, I address this important void by demonstrating the use of a button emulator to aid in push-button design. As a button gets pressed, the rapid compression applied to the internal mechanical structure causes rich (force and tactile) feedback. Researchers have attempted to utilize the Phantom device, a six-DOF pen-type force-rendering platform [146, 191, 71], to generate rich force feedback. Phantom can emulate various levels of resisting force and softness of materials, to generate such force feedback, but it lacks the ability to generate vibrotactile feedback, and its relatively low rendering rate (60 Hz) makes it unsuitable for emulating push-buttons. Softness displays [155, 205, 159] and pseudo-force devices [70, 93, 103, 207] have been explored as alternatives. These too are not suitable for directly emulating push-buttons, since their response rate is insufficiently high. The work closest to this application has involved using vibrotactile feedback to emulate various force responses [175, 113]. At this juncture, it is worth noting that prior research into emulation, with its emphasis on understanding the mechanism of haptic perception or emulating certain feedback, has not utilized emulation to support HILO.
6.2 Emulation: The Case of a Push-Button Emulation Pipeline

This section of the chapter presents the example case of optimizing push-buttons. Thus, I demonstrate the efficacy of employing HILO with physical emulation. The approach to realistic button emulation must address the aforementioned effect of a push-button generating rich feedback within a very short time period. There are two elements to any accurate physical emulation. The first is a model, required for capturing and describing the physical phenomenon. In the case of a button model, it should accurately describe the level of resisting force when the button is pressed 1 cm downward etc. The model informs the goals for the simulation at any given point. Secondly, an emulation workflow is needed, to deliver the output requested from the model. Emulation workflows differ in their level of complexity. In a simpler case, a physical prototype that is able to generate the correct response suffices as the emulator. Other scenarios demand control methods for guaranteeing that the output is near the intended target specified by the model.

The novel technique I introduce below applies a force–displacement–vibration–velocity (FDVV) model to capture button-pressing. After describing this, I introduce an emulation pipeline that, in conjunction with the modeling, provides for accurate emulation. Lastly, I report on evaluation conducted to validate the realism of the emulation.

6.2.1 FDVV Modeling

Accuracy in capturing the tactility of a button requires a model that accurately describes the physical characteristics that are relevant to the process of button-pressing. Below, I outline preexisting force–displacement (FD) models, then present the improved approach.
**FD Models and Their Limitations**

Push-buttons have traditionally been described by way of their force–displacement function or force curve, with separate curves for actuation and release [56, 183, 131]. The FD curve can influence sensations, joint kinematics, muscle activity, and user performance [99, 110, 183, 187, 174]. Linear buttons’ internal structure is composed mainly of a spring, so they do not offer tactile “bump” when pressing or releasing. Tactile buttons, in contrast, have a mechanical structure that generates such a bump or a “snap.” In another variation, some tactile buttons emit a clearly audible clicky sound when reaching the snap point. Other important properties of a button are the travel distance (i.e., the total distance between the initial and the bottommost state of the keycap) and the activation point, or the depth at which the button is activated [112]. Although FD models can capture these properties, they fail to consider velocity-dependency and vibration characteristics, which are critical factors.

Another way to conceptualize push-buttons is as a mass–spring–damper system. The inertia caused by the button’s mass means that the resisting force depends not on displacement alone but also on velocity and acceleration. Previous studies have modeled the softness required for contact with the target surface at various velocities and accelerations, thereby demonstrating the importance of considering velocity- and acceleration dependence. Another problem with FD modeling lies in its limited ability to capture the aforementioned snap sensation caused by the button’s unique internal structure. The force–displacement–velocity–vibration model proposed here addresses that limitation too.

**The FDVV Approach**

The novel modeling method I propose as an extension of FD models is illustrated in Figure 6.2. It captures the velocity dependency relation by incorporating several FD profiles, for various speeds. In addition, vibration is obtained by a microphone during a button press. To accompany the FDVV model, I developed a novel end-to-end emulation workflow covering the operations from capturing a model to emulating it.

### 6.2.2 The Emulation Pipeline

**Step 1 (button capture):** The technique begins with a novel approach to measuring the FDVV characteristics of push-buttons. Firstly, to capture the velocity-dependence of button pressing, one measure presses in several velocity conditions. Secondly, instead of a rigid, static-velocity probing object such as a mechanical probe, a human finger does the pressing. This provides a more realistic response envelope of the sort encountered in users’ real-world button-pressing. Thirdly, vibrations during presses are recorded.
Step 2 (FDVV modeling): To obtain a lower-dimensional representation and enable efficient design and optimization, the procedure transforms the raw measurements from the previous step into an FDVV model. However, the raw measurements are inherently noisy and potentially require high-dimensional parameterization. For overcoming this issue, the data undergo several preparatory operations. The FDVV modeling comprises several steps of preprocessing, accordingly. The first one is filtering, which entails the application of an electrical low-pass filter during data acquisition and a Gaussian filter after the acquisition is complete. The second step is synchronization addressing the fact that data were gathered from the microprocessor and motion-tracker at different rates (1,000 Hz and 256 Hz, respectively). To synchronize the timestamps, the procedure up-sampled
the motion-tracker data. After synchronization, several smoothing operations were performed in succession. The next activity involved fitting the data to a low-dimensional B-spline model. The selection of this model type stemmed from its ability to approximate the data with minimal error while keeping the number of parameters low. For determining the optimal number of B-spline knots, the procedure minimized the Bayesian Information Criteria [119]. This allowed for a lower-dimensional representation of the data, reducing the complexity of the model and enabling more efficient design and optimization.

**Step 3 (using the physical emulator):** Figure 6.4 presents the physical emulator, which has four major components: a linear force actuator, a linear position sensor, a voice coil acting as a vibrotactile motor, and a servo motor. A microprocessor (Adafruit ItsyBitsy M0) drove the force actuator, the sensor, and the servo motor. To adjust the travel range, this microprocessor directed the servo motor to change the location of the travel-range controller, which further modified the travel range of the button. Another microprocessor (Arduino Uno) drove the vibrotactile voice coil with a wave shield. When emitting the vibrotactile cue was needed, the ItsyBitsy board requested Arduino Uno to trigger the vibrotactile motor (voice coil) such that the corresponding wave files play.

**Step 4 (iterative compensation control):** Making sure the force response of a push-button gets emulated accurately requires one to account for the transfer function of the force actuator. Yet no prior work on button or force emulation has considered this issue. To compensate for the transfer function, I propose an iterative compensation method, as depicted in Figure 6.5. The central idea is to adjust the signal amplitude associated with
force-actuation at each displacement point until the target resisting force is detected by the sensor as the key is being pressed against the keycap. This iterative process allows us to achieve accurate emulation of the force response in a manner that factors in the unique transfer function of the force actuator. The iterative compensation process can be mathematically described as

\[ u_{k+1}(p) = u_k(p) + \Gamma(error_k)(y_d(p) - y_k(p)), \quad p \in [1, n], \]  

(6.1)

where \( u_k(p) \) is the actuation signal at displacement point \( p \) in the current iteration. \( u_{k+1}(p) \) is the actuation signal in the next iteration at the same displacement; \( y_k(p) \) is the force detected by the sensor, and \( y_d(p) \) is the target force level at the given displacement point. Finally, \( \Gamma(error) \) represents the proportion of adjustment of the actuation signal that must be applied, arrived at from the error value in this iteration (\( error_k \)). The error from the current iteration is defined as

\[ error_k = \alpha \cdot \frac{\sum_{p=1}^{n} |y_d(p) - y_k(p)|}{n} + (1 - \alpha) \cdot \max_{p \in [1, n]} |y_d(p) - y_k(p)|. \]  

(6.2)

In this expression, two terms constitute the error. One is the average difference between the target FD curve and the measured curve at each measuring displacement. The other is the largest error between the target FD signal and the measured FD signal. The weight applied between the two is \( \alpha \).

![Diagram](image)

Figure 6.5. Iterative compensation finds a way to render an FDVV model via a given simulator plant.

### 6.2.3 Evaluation of the Emulation

To evaluate the perceived realism of the emulation, a controlled study employed ABX testing, which is a commonly used method in psychophysics to compare two possible sensory stimuli to a target [148, 45, 114, 156]. Participants experienced a real reference button (X), after which they were asked to press two synthetic buttons (A and B) and indicate which one offered a more realistic rendering of the target. Both A and B were
rendered by means of the emulator described above. The study compared FDVV-based models against traditional FD ones.

*Participants*
For the study, I recruited 12 participants (6 of them female).

*The Task and Apparatus*
The study compared the realism of six target push-buttons. I captured all these buttons in both traditional FD models and FDVV models. The emulator was placed inside a box. The participants had to reach into the box for performing the pressing action (see Figure 6.6). This prevented the users from being biased by visual cues. An interface informed the participants which of the two buttons (“A” or “B”) was the current rendering.

*Procedure and Experiment Design*
In each round, which exposed the participant to a specific reference button and two emulated buttons, the participants were informed that there were two emulated buttons: button A and button B. They were then instructed to freely experience the reference button by pressing it at different velocities. Then, they were instructed to experience the two emulations, buttons A and B. Participants were told that, when they were ready to provide their judgment, they needed to indicate which provided greater realism (A, B, or neither). Further, they needed to provide the perceived realism of the two emulated buttons by a seven-point Likert scale. The FDVV and FD models were randomly assigned labels A and B.
Results
The FDVV modeling delivered greater perceived realism. In their choices between models, the participants determined the FDVV model to be more realistic 77.31% of the time. Wilcoxon signed-rank tests afforded further analysis. For all buttons presented, there were statistically significant differences between the FDVV models and the traditional FD models. I refer the reader to Publication IV for more details regarding the analysis.

6.3 An Example of Optimization: HILO for Button Design

After evaluating the realism produced by the physical emulation pipeline, I set out to demonstrate the potential for human-in-the-loop button optimization using physical emulation with a specific design task of temporal pointing [127]. Temporal pointing tasks require user activation of a certain feature within a predefined time window, which is a commonplace action in many interactive applications (games etc.). With this optimization process, I did not consider the velocity-dependent properties of button activation; I assumed a consistent pressing speed across all participants.

6.3.1 The Design Parameters and Objective Function

There were eight design parameters in all. Six of them contributed to specifying the button model: $x_1$, $x_2$, and $x_3$ (displacements of these three control points) and $f_1$, $f_2$, and $f_3$ (actuation signals of these three control points). The ranges of these six parameters were $x_1 \in [0,1), x_2 \in [1,3), x_3 \in [3,6.2)$ and $f_1, f_2, f_3 \in [20,180]$. The final two parameters were the activation point, $p_a$, and the vibration point, $p_v$. Their ranges were $p_a, p_v \in [0.5,5.5]$.

The scenario was a case of single-objective optimization in which the objective was to minimize a user’s temporal error, i.e., the mean asynchrony [127, 188].

6.3.2 The Study Setting

I recruited 10 participants (4 of them female) for this optimization study. The participants were presented with the button emulator and an LED strip. They were instructed to press the button when the yellow bullet on the LED strip had reached the center of the blue target zone (Figure 6.7). There were two levels of task difficulty: “Easy” (the bullet moves slower) and “Difficult” (the bullet moves faster). For each iteration of each difficulty level, 27 trials were collected and then the system computed the mean asynchrony. The optimizer then updated the design based on the observed performance. Comparing against the optimized button, I selected a series of
Figure 6.7. The setting for the push-button optimization.

Figure 6.8. The results of the push-button optimization. The optimal button is the personalized optimal design. The Clear, Brown, Black, and Red switches are 4 mm mechanical Cherry MX switches rendered by our emulator. Lastly, the random button is a design generated randomly within the design space.

Cherry MX 4 mm switches\(^1\) and a randomly generated button as baselines.

### 6.3.3 Results

For the Easy level, the mean asynchronies produced for the optimal, clear, brown, black, red, and random buttons were 65.8 ms (SD = 6.15), 83.6 (SD = 7.65), 88.04 (SD = 7.82), 81.65 (SD = 6.19), 84.28 (SD = 6.75), and 100.78 (SD = 6.89), respectively. For the Difficult level, the asynchrony’s mean values, in corresponding order, were 77.3 ms (SD = 6.89), 93.43 (SD = 7.56), 97.48 (SD = 7.69), 96.9 (SD = 7.61), 96.65 (SD = 6.36), and 108.22 (SD = 8.77). Two-way repeated-measures ANOVA revealed the existence of a significant main effect of buttons on mean asynchronies, \(F(5,95) = 10.724\), with \(p < 0.001\). Pairwise post-hoc comparisons with Bonferroni correction showed that the optimal button design indeed outperformed the rest (\(p < 0.05\)).

\(^1\)See [https://www.cherrymx.de/en/cherry-mx.html](https://www.cherrymx.de/en/cherry-mx.html).
6.4 Summary and Discussion

Designing physical interactions often entails optimizing their physical and mechanical properties. However, varying the physical properties requires sets of physical prototypes, which may display a host of variations, for communication and testing of the ideas involved. Because the ensuing resource demands may make HILO difficult to apply for such design tasks, I explored emulation as a potential solution, thereby striving to answer CQ 1. Instead of creating a physical prototype for each iteration, the solution developed imitates the mechanical properties of particular designs via emulation. This work accounted for both of the important aspects to emulation: the model of the physical phenomenon and a workflow to support the emulation, which usually involves control methods.

To demonstrate the concept’s workability in practice, I implemented this novel approach for emulation in the form of FDVV models for button-pressing and an end-to-end emulation pipeline. I investigated the proposed method’s efficacy further by taking push-buttons as an example and conducting an evaluation experiment that demonstrated reaching greater realism than previously implemented approaches have produced. Finally, I conducted HILO with the emulator to optimize push-buttons for a temporal pointing task. The optimal button designs that emerged outperformed the baseline preexisting button designs.

This chapter ties in with the dissertation’s fourth contribution, human-in-the-loop optimization with emulation, through the proposed novel approach to applying HILO with emulation. The chapter also points to several interesting directions for future research. On one potential path, scholars could embrace physical emulation as a design tool rather than merely a means of generating specific feedback. This could involve exploring how designers might use emulation to prototype and iterate on physical designs more efficiently and effectively. Additionally, future research could investigate a wider range of applications for HILO with emulation, beyond push-button design alone. This work could include exploring possible uses of emulation to optimize other types of physical interactions or to support the design of more complex systems.
7. From the Real World toward Simulation

“What I cannot create, I do not understand.” — Richard Feynman

“Science is what we understand well enough to explain to a computer; art is everything else.” — Donald E. Knuth

My attention thus far in the dissertation has been directed to expanding the application scope of HILO to multi-objective cases, group-level optimization, and HILO’s practical implementation with physical emulation. In this chapter, my aim is to tackle the issue of the cost and effort associated with conducting human evaluations within the HILO process. Evaluation is costly for both the designers and the participating users: the designers need to plan the study, recruit people to participate, and conduct the HILO, while users have to dedicate immense effort to repetitive tedious tasks. Additionally, as the earlier chapters elucidate, performing HILO in itself is a time-consuming process.

The central question I set out to answer in this chapter is CQ 1: Can we free the HILO procedures of human efforts? To this end, I extend the definition of the term “human” in Human-in-the-Loop Optimization in this chapter: Previously, the word “human” referred to human participants, and here, the word refers to both “actual human participants” and also “synthetic human participants.” With this extension, I propose a novel simulation-based optimization framework to automate the evaluation process. Within this framework, there is an agent governed by a user model, which produces human-like behaviors to inform the evaluation of each design’s quality in a simulation environment. An optimizer then iteratively proposes the next design instance that is likely to yield the best possible performance of the agent. Since no human users are involved, this framework offers the designer high efficiency and automation. Furthermore, since the agent’s performance does not deteriorate over time (there are no fatigue effects), the framework provides greater robustness.

To demonstrate the efficacy of this framework here, I present a use case of
optimizing a 3D pointing interaction by means of the proposed simulation-based optimization framework. The results attest that the framework supports optimizing the pointing interaction with highly efficient automation. Overall, this framework represents a promising route to eliminating the cost and effort associated with employing human evaluations in the design process. Future research could investigate the scalability and generalizability of this framework with various design tasks and domains.

7.1 A Simulation-Based HILO Framework

This section of the chapter deals with the novel framework I propose for the simulation-based optimization of design interfaces. The discussion begins with an orientation to prior work. Shifting the optimization process from the real world to a simulation environment is not a new idea.

7.1.1 Background and Related Work

As is evident from the review in section 2.2, simulation-based optimization is commonplace when engineering practitioners find physical evaluation overly expensive or otherwise impractical [144]. Engineers have applied it for optimizing buildings [163], urban transportation design [170], aircraft [210], the aerodynamics of rocket design [202], and other designs. However, these engineering tasks display a fundamental difference from optimization of interface design: as noted earlier on, the former often relies on well-established models of the problem.

While HCI researchers have attempted to optimize interfaces by using simulations, their efforts are plagued by the fact that many interface interactions lack precise models. In fact, this is the main challenge they face. For example, optimization of layout-assignment problems often utilizes the well-established Fitts’ law model, which is suitable primarily for simple tasks, such as optimizing soft-keyboard layout [20], physical keyboards’ layout [62], and keyboard layout for gaze-based interaction [18]. Other HCI optimization work relies on assumptions about user behavior (e.g., the frequency of using particular functions), in areas such as applying combinatorial optimization methods to graphical layouts [171] and optimizing menus via reinforcement learning (RL) [213]. Likewise, some display optimization is based on heuristics rather than user models. For example, Luzhnica and Veas optimized a tactile pattern by avoiding two vibration points identified as too close to each other [140].

Frequently, these techniques’ dependence on well-established models or simple user-behavior assumptions renders them unable to generalize to different tasks or handle more complex settings that require physical motor control and interaction. These shortcomings highlight the need for
a simulation-based optimization framework that can handle a broad range of HILO tasks even in the absence of precise models or assumptions about user behavior.

### 7.1.2 The Simulation-Based Optimization Framework

In light of the need described above, I propose a novel simulation-based HILO framework intended both to 1) be generalizable to a wider range of tasks and 2) handle complex and realistic interfaces. Figure 7.1 illustrates the framework. The core concept is moving the HILO from the real world to the simulation domain. This framework has three key components:

- **A physical simulation environment**: A realistic environment that can simulate physical responses and phenomena is needed. This must cope with a wide range of tasks.

- **An RL-based user model**: Reinforcement learning is a general framework in which agents learn the policy for interaction with the environment via trial and error [206]. An RL-based user model permits interaction with a broad range of tasks, even complicated and realistic ones.

- **Bayesian optimization**: Since BO does not require assumptions about or knowledge of the problem or the user model, it is generalizable to numerous sorts of design-optimization problems.

In a divergence from previous simulation work in the HCI field, most of which has targeted a specific interaction, the elements in the framework I propose are not limited to certain interactions or tasks. Rather, they are generic solutions for a wide range of problems. Against the backdrop of the preceding chapters’ thorough introduction to Bayesian optimization, the discussion below focuses on presenting the simulation environment and the RL-based modeling involved.

It is worth mentioning that the proposed simulation-based optimization framework differs from the previous model-based optimization, such as optimizing layout based on Fitts’ law, in the form of evaluation. In the traditional approach where simplistic models are applied, the model directly generates the statistical evaluation result of a design instance. For instance, one can apply Fitts’ law to predict the completion time of a selection. Yet, such predictions are not based on moment-by-moment movements. On the other hand, the proposed simulation-based optimization framework generates moment-by-moment motions and interactions and then the system calculates the performance metrics. It permits more realistic replication of user evaluation and also allows for higher generalizability.
7.1.3 Physics Simulation

Physics simulation and engines have revolutionized the study and analysis of complex systems, across various fields [124]. Recent years have seen them gain particular importance for the training of robots and deep-RL policies. Of the various physics engines introduced (DART [129], PhyX\textsuperscript{1} ODE [203], etc. [59]), MuJoCo [215] stands out for its exceptional realism and efficiency. For the work presented in this chapter, I employed the simulation-based HILO framework by means of that leading physics engine.\textsuperscript{2}

7.1.4 Reinforcement-Learning-Based User Models

There is a long history of constructing user models to improve understanding and prediction of user behaviors [21], and recently RL has emerged as a generic framework for modeling a wide range of tasks [42, 152, 101]. In this general framework, an agent learns to select a series of actions for the optimal accumulated reward. Typically, RL is modeled as a Markov decision process with the following key elements:

- **The state space**, $S$: All possible statuses of the agent and the environment
- **The action space**, $A$: All actions (choices) available for the agent to select
- **A reward function**, $R(s,s')$: A function that generates the utility of performing a certain action from a given state

\textsuperscript{1}See https://developer.nvidia.com/physx-sdk.
\textsuperscript{2}For more information about MuJoCo, please consult its official Web site, https://mujoco.org/.
For a more detailed introduction to reinforcement learning, please refer to the presentation by Sutton and Barto [206]. Below, I present an example case wherein RL as described here can be utilized to model and simulate humans’ motor behaviors. Specifically, I model affordance perceptions through recognizing motions, which are acquired via the mechanism of RL. A preliminary case of design optimization utilizing RL models is presented after the affordance perception modeling in section 7.2.

### 7.1.5 Reinforcement Learning for Understanding of Affordance

An important step toward agent-in-the-loop optimization is to allow the agents to interact with the target interface as envisioned. That is, the agent should be able to perceive affordance. Despite its importance in HCI and a rich corpus from ecological psychology, researchers have never produced a theory that explains the formation of affordance. Simultaneously, the numerous supervised-learning-based computational models available remain largely detached from the world of motion planning. Therefore, such affordance models cannot be directly applied to guide agents’ motions. Addressing these gaps, I developed a theory explaining the formation of affordance. I argue that the perception of affordance is obtained via reinforcement learning. The key elements of this theory can be articulated thus:

1. **Affordances are learned when reinforcement signals are provided in response to motions.** An organism tries out different possible motions to learn which motion results in the highest reinforcement signal (i.e., the greatest reward). In the future, a rational agent will then avoid a motion that leads to little reward and repeats/favors a motion that brings a large reward [40].

2. **Affordance perception is guided by predicted rewards.** The solution to the problem of delayed feedback is dependent on prediction, which is grounded in reinforcement learning [206]. To decide on an optimal action at the moment, the brain must learn to predict the outcome of a given action [77].

3. **Affordances are learned by exploring and exploiting.** Based on the theory of reinforcement learning, a rational agent intelligently balances exploration and exploitation, aiming to maximize the reward.

4. **Affordance perception generalizes to unseen instances of a category.** An organism generalizes policies via feature similarities. In other words, the motions toward two similar objects are similar.

5. **Humans learn to associate linguistic categories (labels) with affordances.** Nearly always, affordance theory has presumed linguistic categorization of actions [216, 36]. Such linguistic labeling allows us to reason and communicate, which further boosts the development of cognitive representations.
From the Real World toward Simulation

Figure 7.2. The team developed a computational model of the affordance theory, implemented in the MuJoCo physics engine. It enables a virtual robot to interact with widgets.

Figure 7.3. The virtual robot interacted with widgets of several types: (a) button widgets; (b) slider widgets; and (c) a deceptive widget that, while resembling a slider, allows only push actions, similar to a button.

Evaluating the Affordance Model via Deceptive Widgets

For the evaluation, I devised a simulation implementing a robot agent, which is essentially a robotic arm with characteristics similar to the human (Figure 7.2). There were two types of widgets for the agent to interact: buttons (in Figure 7.3, pane a) and sliders (in pane b). Each widget affords a particular action – a button provides the possibility of “pressing” and a slider provides the possibility of “sliding.” In addition, I implemented a “deceptive widget,” which has the appearance of a slider but it only affords “pressing”; this widget (shown in the figure’s pane c) was deployed only in testing. When successfully triggering a widget, the agent receives a reward signal.

In the training phase, the agent interacts with the buttons and the sliders, via the process depicted in Figure 7.4, pane a. Evaluation was carried out after training, to confirm the robot’s ability to use these two widgets correctly. In 1,000 trials with each widget, the agent interacted with the button and achieved a success rate of 91.3%; meanwhile, the agent interacted with the slider and achieved a success rate of 94.5%.
Next, I labeled 1,000 successful movement trajectories with each widget: the trajectories for activating the button were labeled as “press”, and the trajectories for activating the slider were labeled as “slide.” With these labeled data, I proceeded to train a motion classifier. Here, 80% of the data items were randomly assigned to the training set, 10% to the validation set, and 10% to the testing set. The motion classifier achieved 85.8% validation accuracy and 88.1% testing accuracy, indicative of high-quality recognition.

In the last step, I evaluated the model by using the deceptive widget. The robot interacted with the widget and continued discovering the right action through successive interactions. The results, presented in Figure 7.4 (b and c), show the evolution of progress in the agent’s affordance perception. The agent initially perceived a greater affordance for sliding and, consequently, experienced low success rates. Through interactions and reinforcement learning, however, the robot gradually adjusted its policy and came to perceive the right affordance (i.e., pressing).

### 7.2 A Preliminary Example Case: 3D Touch Optimization

This section describes the validation of the efficacy of the framework whose core elements are outlined above. That simulation-based optimization framework was validated for the goal of optimizing the parameter settings of a transfer function for 3D touch interaction. The validation application followed three phases:

1. **Agent training** – training the agent to acquire the policy for the target interaction via reinforcement learning

2. **Optimization** – deploying the optimizer to derive the Pareto-optimal parameter setting(s)

3. **Evaluation** – comparing the performance when given the optimized
parameter setting and a random parameter setting

7.2.1 Configuration for the 3D Touch Interaction and Optimization

The interaction setup was established in the manner introduced in subsection 4.3.1. Again, the research team chose the Go-Go technique to serve as the transfer function. It makes use of two parameters, \( D \) and \( k \) (for details of \( D \), \( k \), and the definition of operation distance, please refer back to Figure 4.6). For the present setting, there were a few subtle modifications: Firstly, since there is no controller in the simulation, I used the fingertip position in place of the controller position; hence, the Go-Go technique computation’s figure for the distance between the controller and the center of the body uses the distance from the fingertip to the center of the body. Secondly, because there is no button for the agent’s completion of the selection when the cursor reaches the target, dwelling for five steps (timestamp increments) denoted making the selection. Figure 7.5 shows the interaction with the target in the simulation.

Iteration and Target Placement
The agent-training phase utilized targets randomly sampled from the \([1,2]\) operation distances range. A uniform distribution was used. Publication II supplies details related to the operation distance in the study conducted. For the optimization phase, I evenly discretized the target range into 21
distances (i.e., operation distances of \([1, 1.05, 1.1, ..., 2]\)). In each optimization iteration, every discrete distance was sampled 200 times, for a total of \(21(d\text{istances}) \times 200\) times selections.

**Design Parameters**
For the parameters, I used the settings \(d \in [0.3, 0.4]\) and \(k \in [0.5, 1.5]\).

**Objective Functions**
Two objective functions were considered, for the hit rate and the efficiency. As noted above, there were 4,200 selections per optimization iteration. The hit rate, defined as the number of successful selections in each iteration divided by 4,200, has the range \([0, 1]\). Efficiency was a normalization function of completion time. Because there is no convenient time unit (second, minute, etc.) in the simulation environment, I took the simulation step as the unit for time. Specifically, I converted \([160, 20]\) steps to a \([0, 1]\) range for the efficiency metric. Therefore, for both objective functions, a higher value reflected better performance.

**Settings for Bayesian Optimization**
I set up MOBO to use expected hypervolume improvement (EHVI) \([52]\). The initial sampling, with a value set to 5, was followed by 30 optimization iterations.

### 7.2.2 Agent Settings

In the system developed, the agent has two components: the biomechanical model and the policy model. The biomechanical model chosen is the one introduced and used for User-in-the-Box \([94]\).

**State**
Here, the state is composed of three elements together constituting the observation of the biomechanical model: The first is vision: The model has a fixed forward-facing visual field. I took the full set of RGB values from the 2D view as part of the state. The second element is proprioception, represented via the 3D coordinates of the fingertip position and the cursor position. The final component, the target position, refers to the 3D coordinates for the center of the target.

**Action**
The action space is identical to that in the User-in-the-Box work. It includes all the muscles of the right arm of the agent.

**The Reward Function**
When the agent carries out an action \(a\), the environment generates a reward \(r(s, a)\). The reward has two parts, a distance penalty and task-
completion reward. The **distance penalty** is based on the distance from
the center of the cursor to the center of the target. This distance is mul-
tiplied by a scalar and normalized to the value $\in [-0.01, 0]$. The closer the
cursor is to the target, the smaller the penalty received. When the target
is successfully selected (i.e., once the cursor has reached the target and
stayed there for five steps), the agent receives a **task-completion reward**
of value 10; otherwise, the reward is 0.

**Training the Policy Model**
I employed Proximal Policy Optimization [194] to teach the policy of this
interaction. I directly used the implementation in the stable-baseline3
library. The library’s documentation supplies further details of the
hyperparameter settings.
To derive a generalized policy that is able to interact with a range of
potential design settings, at the beginning of each iteration, a set of new
design parameter ($d$ and $k$) values are uniformly sampled from the given
range (as shown in Design Parameters).
I set the maximum step of each episode (the horizon) to 160 steps. If
the agent fails to select the target within 160 steps, it does not receive the
task-completion reward, and the environment is reset.

**7.2.3 Results**
Below, I present the results in terms of the sequence followed. I begin with
the first phase, addressing the training progress of the agent model. Then,
I present the optimization results and the optimized parameter setting.
The subsection ends with a comparison between the optimized setting and
random settings.

**Phase 1: Agent Training**
I trained the agent for $10^8$ episodes. Figure 7.6 presents the progress of
training. At the left is the episode length over the course of the train-
ing. Episode length gradually decreased. This is evidence that the agent
learned to complete the task more efficiently. The right-hand part of the
figure shows the episodic reward per episode, which rose throughout the
training process.

**Phase 2: Optimization**
Figure 7.7 presents the objective functions resulting from the optimiza-
tion. There are two objective functions in our application (hit range and
efficiency), and we can see a strong correlation between them. Namely, a
design that allows for better access to the targets (higher hit rate) also

---

allows for a more efficient selection. Although multi-objective optimization was employed, there is only one optimized design on the Pareto frontier, which is the design that yields the highest hit rate as well as the greatest efficiency. The optimized design is \( d = 0.351 \) and \( k = 1.23 \).

Also note that, although we did not observe the formation of a Pareto frontier in the outcome of the optimization, we can see that the performances of the sampled designs are generally located in the upper right corner, but not scattered over the whole objective function space. This shows that the optimization is effective as the derived designs lead to positive but not random performances.
Phase 3: Evaluation
I evaluated the performance of the optimized design ($d = 0.351$ and $k = 1.23$) and randomly sampled designs for 10 iterations. For the optimized-design condition, the same parameter setting was applied over all 10 iterations. The sampling for the random condition took a new parameter setting for every iteration. The results are presented in Figure 7.8. For hit rate, the optimized design reached 0.961 ($s.d. = 0.035$) while the random designs reached 0.656 ($s.d. = 0.16$). The optimized design reached an efficiency value of 0.54 ($s.d. = 0.019$), and the random designs’ value was 0.383 ($s.d. = 0.88$).

In t-tests run to compare the performance levels, significant differences emerged for both objective functions (both $p < 0.05$). This analysis showed that the optimized design outperformed the baseline conditions in both efficiency and hit rate.

Discussion and Future Work
The positive results of the preliminary evaluation notwithstanding, more work is needed. An critical next step is to employ more diverse design objective functions so that the optimized results form a Pareto frontier instead of a single optimal design. The evaluation of different tradeoffs along the Pareto front will then inform us of the efficacy of generating optimal designs for different objective preferences.

Another important topic for consideration is evaluating the optimized design in the real world. At present, the optimized design has been evaluated only with the agent; the differences between the agent and human participants have not been fully investigated. The design that proved optimal for it might not be the optimum for certain users. It is important to investigate the discrepancy between the synthetic performance of the agents and the actual performance of the human participants. Along this line of research, future work should also investigate how to generate more realistic synthetic behaviors.
Also, the agent in the simulation conducted had a fixed set of physical characteristics. Users in the real world exhibit variations in numerous physical features – limb length, muscle strength, etc. Furthermore, users may differ in their strategies for approaching the task. The simulation did not consider these between-user variances. An important next step would be to train a policy model that is generalized for different user types, which should allow for more generalized optimal designs.

Last but not least, future research should consider applying this simulation-based optimization framework to other interaction techniques. The use of RL allows for modeling a potentially wide range of interactions. However, as the complexity increases, it is more challenging for the RL-based approach to generate human-like behaviors due to the difficulties in crafting realistic reward functions and physical characteristics. It is important to understand the limitation of RL-based agents for simulating human motor behaviors, which will then be useful for us to fully understand the application scope of such simulation-based optimization.

7.3 The Work in Summary

For the chapter-level goal of addressing how to make HILO more accessible to practitioners by eliminating the cost of human user studies, I have presented a novel simulation-based HILO framework that supports the efficient optimization of interaction designs by deploying an agent as the evaluator of the design instance. This work forms part of the last contribution listed for the dissertation, a simulation-based human-in-the-loop optimization framework. The key difference between this framework and previously introduced optimization systems using simulation is that it is not constrained to any specific interaction or user model. It can be applied to a wide range of interaction tasks.

The framework factored in three key elements: a physics simulation, an RL-based user model, and BO. The workflow is another important aspect of the work. In the example case presented, the first step of the workflow was to train the user model to acquire a policy model capable of performing the target interaction, the second step was to deploy an optimizer to derive the optimal designs, and the evaluation of the designs derived formed the final step. While further evaluation of the framework is necessary, for a better understanding of its limitations, the results of the preliminary study clearly show that the optimized design for 3D touch interaction outperforms the baseline.

Overall, the work constitutes a promising avenue for practitioners’ efforts to optimize interaction designs without incurring the high cost and effort of running human studies. Through the power of physics simulations and RL-based user models, this simulation-based HILO framework offers an
efficient and accessible approach to interaction-design optimization.
Design optimization involves exploring a large design space to identify the optimal parameter setting. This necessary search task remains complex and challenging. A realistic design challenge usually involves multiple objective functions that a designer must consider during the optimization process. Furthermore, the design parameters are often continuous, presenting a nearly infinite set of possible combinations. The relationship between the resulting objective function and a particular design instance is often unknown, and evaluating the design instance with human participants is an expensive undertaking. Manual optimization, a process in which designers seek a good design by means of various design methods, intuition, and exhaustive trial and error, is a highly demanding and effortful approach to this problem, and it still does not guarantee an optimal outcome.

HILO holds great potential as a solution for design optimization, offering more principled, generic, and automated applications. However, several critical limits have constrained its application scope thus far.

The first constraint arises from HILO having been applied exclusively to single-objective rather than multi-objective design tasks. To address this limitation, I have proposed applying multi-objective BO based on Pareto-frontier learning in HILO. This process identifies a set of designs that lie on the Pareto frontier.

Secondly, researchers have not comprehensively investigated HILO’s strengths and limitations. In the dissertation project, a workshop and user study informed understanding of how designers perceive the HILO process. The findings indicate that the participating designers experienced significantly less mental and physical effort in searching for the optimal designs with HILO, relative to manual approaches.

In addition, HILO has been applied with regard to individual users only, as opposed to a group of users. To address this restriction in scope, I have proposed group-level HILO, which aggregates optimization observations from a group of users and derives a model accordingly. Simulations and two user studies attest to the efficacy of the two extensions devised for this
Discussion and Conclusion

purpose, Global GP and Warm-Start GP.

The fourth important restriction is that HILO must be implemented for the interaction in such a way that the design process need not involve fabrication. To overcome this shortcoming, the project utilized physical emulation for HILO. Through this mechanism, designers need not fabricate or prototype each design instance.

Finally, my research addressed the issues arising from the requirement for every iteration in HILO to include a human evaluator, which is costly and effortful for both the designers and the other participants. The work behind the dissertation extended HILO from the real world to physical simulations to circumvent the hassle and effort of human evaluation.

In conclusion, this dissertation contributes to the field of human-in-the-loop optimization by introducing a set of methods that broaden the application scope of HILO. The methods presented here address the critical constraints of HILO, paving the way for future work on applying HILO to a wider range of design problems.

8.1 The Findings Overall and Their Implications

In summary, the dissertation demonstrates the effectiveness of HILO for reducing designer effort and arriving at better design outcomes. Furthermore, the project has expanded the scope for HILO’s application by implementing various computational methods that perform better than traditional design processes. Together, these showcase HILO’s potential as a versatile solution: the findings suggest that automating design optimization via computational methods is a feasible and reasonable way forward. State-of-the-art methods such as Bayesian optimization are able to overcome the considerable challenges arising from the complexity and noise associated with human–computer interactions, and they can lead to promising user performance in response to diverse design challenges.

There is a clear implication that design practitioners should embrace these means of computational optimization. Additionally, the dissertation highlights expansion in the space ripe for HILO-related research, in that the optimization step has already become automated in many branches of engineering. With advances to computing capability and machine-learning tools, it is now possible to tackle interaction optimization via these techniques. The flexibility of BO invites further enhancements, and future research could push the boundaries of HILO even further.
8.2 Limitations and Future Work

The research presented here contributes to work on several topics. With this fertile ground come several limitations that future steps must address.

8.2.1 Advanced Optimization Methods

To enhance the efficacy of the HILO process further, researchers should strive toward more advanced optimization algorithms. Bayesian optimization is a general framework within which there is ample room for such enhancements.

More Efficient Optimization

One critical limitation of HILO is that it can be time-consuming. For example, with the 3D touch interaction presented in Chapter 4, a two-objective problem featuring four design parameters, it took 60 to 90 minutes (40 iterations) for multi-objective BO to arrive at final designs. A need for more iterations and time is to be expected if the problem has more parameters or objective functions. Alongside the Warm-Start GP demonstration provided in the dissertation, extensive research has examined how to enable more efficient BO [10], with one mainstream approach being meta-learning [217]. Meta-learning in the context of BO is a set of techniques designed to boost the efficiency of solving an unseen optimization problem by incorporating auxiliary data from similar tasks. Several implementations of BO that employ meta-learning have been proposed, some including multi-task GP, deep neural models, etc. I encourage researchers to look into the opportunity of applying them to HILO problems.

High-Dimensional Optimization

One important restriction still hampering state-of-the-art BO-based HILO is its highly limited ability to handle a larger number of design parameters. For instance, design tasks that have more than five parameters prove computationally expensive, and it may become challenging to address a task with more than 10 parameters at all. Real-world design problems often require the designer to deal with a larger number of design parameters. To address the associated limitation, researchers recently have introduced high-dimensional BO approaches [154], which typically involve learning a low-dimensional latent parameter space that BO can operate on, thus enabling the optimization of designs that have more parameters. It is worth exploring whether such techniques can be extended to high-dimensionality HILO problems, thereby enhancing the ability of the approach to handle more complex design problems.
Automated Machine Learning for BO

One step inherent to HILO is setting up the optimizer, which involves tuning hyperparameters at various levels of abstraction. This unavoidable step may discourage designers who do not have a programming background. To make HILO more accessible to practitioners despite this challenge, we can draw inspiration from the field of automated machine learning. Researchers in this emerging field, also known as AutoML, aim to automate the machine-learning pipeline, including hyperparameters’ tuning [89]. In essence, AutoML algorithms search for the best machine-learning models and hyperparameters for the particular task and data given. We might be able to reduce the difficulty of setting hyperparameters by integrating AutoML techniques into HILO; further automating the optimization process in this manner could render HILO more accessible to a wider range of practitioners while also aligning it with the trend of automating machine-learning workflows more broadly.

Hierarchical Group-Level Optimization

The Global GP implementation presented in the dissertation has shown promise for generating Pareto-optimal designs suited to a larger group of users. In many realistic design scenarios, though, the target user population may consist of several groups, each with a unique set of needs and preferences. We could approach these groups via a hierarchical structure, with subgroups and, in turn, individual users within each of those. In such cases, it is important to consider group-level optimization for generating hierarchical group-optimized designs via HILO. This would involve developing a framework that can capture the design requirements and preferences distinct to each group well, then generate optimized designs that satisfy those specific needs, all while guaranteeing general feasibility and compatibility. One possible path to this end is through the use of a multi-level covariance kernel [231] and multi-task GP [23], which can capture hierarchical structures in the data and enable efficient group-level optimization. Such an approach possesses the potential to enhance the practical utility of HILO in real-world design applications significantly. Further research and development in this area, which could encompass exploring other hierarchical machine-learning methods, could inform a more comprehensive and robust HILO framework, one able to cater to a wider range of design problems and user populations while still exploiting the power of Global GP.

8.2.2 Making HILO More Usable for Designers

While the HILO framework has demonstrated its effectiveness for varied design tasks, several areas remain in which the framework could be made more usable for designers. Future research plans should consider
Discussion and Conclusion

enhancing HILO's practicality/usefulness by means of more realistic and complexity-attuned user studies. Better human–machine collaboration is another vital goal.

Investigation of More Realistic Design Challenges
The workshop and user studies conducted in connection with the dissertation shed valuable light on the effectiveness of HILO in a controlled environment with a limited number of users. For a full understanding of the practicality and possible benefits of HILO, future research should aim for more realistic, in-the-wild experiments. Some experiments might involve pairing designers with a larger number of users and allowing them to explore various applications of HILO. Evaluating the subjective feedback of designers and users in a real-world setting would more readily permit judging HILO's effectiveness. Furthermore, it is important to evaluate designer perceptions of the approach's usefulness in practical design conditions (which lack extensive technical support etc.). Such experiments could yield valuable insight for both understanding the practicality of HILO and refining the framework such that it responds better to real-world design problems and the needs of actual user populations.

Better Designer–Optimizer Collaboration
From the qualitative analysis of the studies, it became clear that the designers sometimes felt disconnected from the design process and that the optimizer was driving the design decisions, not vice versa. At the same time, the studies showcased human designers’ unique ability to quickly identify areas of the design space that do not merit exploring, which is difficult for the optimizer to do. This prompts us to ask an important question: how we can improve collaboration between human designers and optimizers? Answering this demands two lines of attack. Firstly, we need to develop collaborative interactions that permit humans and optimizers to make decisions jointly, and, secondly, we must identify a computational method that supports such collaboration appropriately. This should bring a deeper understanding of how human designers make decisions and how we can effectively integrate their knowledge and preferences into the optimization process. Among the possible approaches are to develop more interactive optimization methods that allow human input in real-time and to create hybrid optimization methods that blend human decision-making with automated optimization. Further research in this area could help bridge the gap between human designers and optimization algorithms, thus leading to more effective and efficient design processes.
8.2.3 Work toward More Realistic Simulation and Emulation

The dissertation provides a starting point via the proposed simulation-based and emulation-based HILO, accompanied by reports on a few applications and preliminary studies. To take the work further, researchers should consider investigating other techniques to address the gulf between simulation/emulation and the real world.

Evaluating the Simulation-Based Optimized Designs with Human Participants

The simulation-based optimization framework proposed in this dissertation was evaluated in a preliminary experiment using a synthetic user agent. While this provided valuable insight, studies with human participants are vital for enriching our understanding of the framework’s functionality in real-world scenarios. Importantly, the possibility of real users’ behavior and interactions deviating from synthetic users’ necessitates testing the framework with a broad spectrum of users. Conducting a full-fledged study with human participants should help to furnish researchers with a more comprehensive evaluation of the proposed framework and its potential impact on practical design applications.

Advanced User Models

A crucial direction for future research is to develop more human-oriented models for design optimization. The models employed in the dissertation project rely on policy-based reinforcement learning, which may generate movements that are not entirely similar to those of humans. Researchers could address this issue by employing reward-shaping [123] and imitation RL [91], for closer alignment of current models’ behavior with human behavior. However, since humans adapt much more quickly than RL agents, meta-learning [219] or model-based learning approaches [153] are needed, for higher efficiency. In addition, factors such as learning and fatigue affect human performance. These are not easily replicated by RL agents. Therefore, it is important to develop realistic models that can better capture human behavior and its variations in real-world design scenarios.

Transferring Designs from Simulation/Emulation to Products

Simulation- or emulation-based optimization work wrestles with the critical issue of translating the optimized design from the simulated or emulated environment into the real world, a process commonly known as “sim-to-real transfer” [235]. The ultimate goal of this process is to realize the optimized design in the physical world, such that end users can use it. This simple goal notwithstanding, transferring designs from simulation to reality is a complex process faced with various technical and practical challenges. Among them are accounting for uncertainties, inaccuracies,
and divergences between the simulation domain (the simulation world’s physical environment and equipment) and the real world. Addressing these challenges requires developing robust, effective sim-to-real transfer methods that can assure the optimized design’s functionality and reliability in real-world settings. Further research in this area could help strengthen the connection between simulation and reality, thus facilitating the efficient and effective realization of optimized designs in practical applications.

**Exploring Other High-Precision Emulation**

Lastly, there is room for further work associated with the emulation-supported HILO approach I have proposed for the button-pressing task. Exploring other applications is necessary for evaluating the effectiveness and versatility of the approach suggested. One critical factor for successful emulation-based HILO is high-precision emulation, which can be achieved through the development of more accurate models, better emulators, and/or more powerful control methods. Further advances in this direction could promote identifying other applications that could benefit from the proposed approach. Such work too would pave the way for the development of more accurate and efficient emulation-based HILO frameworks.

### 8.3 Conclusion

With this dissertation, I aim to address the complex and challenging task of design optimization through the use of HILO. While HILO has shown potential as a general solution for design optimization, its scope of application has remained restricted thus far, on account of several limitations. To overcome them, I have introduced a set of computational methods with the potential to augment HILO’s capabilities. Work applying Pareto-frontier learning for multi-objective HILO problems demonstrated both positive and negative qualities of HILO. While demonstrating the methods’ effectiveness, the research showed that incorporating HILO can reduce the designer’s sense of agency and ownership. Another technique I examined is group-level optimization; this generates group-optimized designs and rapidly adapts group-level warm-start models. My work also demonstrated the potential of emulation deployed as an alternative solution to reduce the cost of fabrication, coupled with a simulation-based optimization framework to eliminate the costs entailed by human evaluation. This simulation-based framework’s three generic elements allow it to be applied with ease to other design optimization tasks. All of these efforts contribute to enhanced applicability in the design domain.

The outcomes presented here advance the fields of design and HCI through the following contributions:
Discussion and Conclusion

1. Knowledge: The dissertation showed that HILO is a viable and effective solution for design optimization problems. Various experiments and studies carried out attest that HILO produces better design instances than traditional design methods do. The knowledge produced can serve to inform and guide future research in the field of design optimization.

2. Methods: The computational methods proposed in this dissertation expand the application scope of HILO. By enabling HILO to tackle multi-objective and group-level optimization problems, these methods make it a more versatile and powerful tool for designers. Additionally, the use of physical emulation and simulation has showcased HILO's benefits and potential in relation to the time and other resource demands of prototyping and user studies.

3. Use cases: The use cases presented exemplify the proposed computational methods’ suitability for a range of optimization tasks that span such design domains as input devices, wearable haptic interfaces, VR/AR interactions, and physical interfaces. By demonstrating the effectiveness and applicability of HILO for a wide range of design-optimization scenarios, these applications and my presentation of them together lay a foundation for solid research and development in the field.

This dissertation contributes demonstrably to the fields of HCI, design, and machine learning through valuable insight enriching HILO research. The computational methods introduced attest to HILO’s potential for various design-optimization tasks, and the use cases’ documentation attests to their effectiveness in extending HILO further. The findings and results reported upon in the dissertation open many interesting research questions for further discussion, providing useful information and inspiration for future research in this area.
References


References


References


