Discovery Through Bottlenecks in Multimodal Models

Mia Chiquier

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2025

© 2025

Mia Chiquier

All Rights Reserved

Abstract

Discovery Through Bottlenecks in Multimodal Models

Mia Chiquier

Modern machine learning systems generate photorealistic images, classify data with superhuman accuracy, and synthesize human-like text, yet understanding and controlling their behavior remains challenging. While interpretability techniques and conditioning mechanisms have made progress, we propose a fundamentally different approach: building systems where information flows through inherently interpretable bottlenecks, unifying discovery and control by construction. This thesis presents three bottleneck methods for multimodal systems that exploit a key duality: the same mechanisms that reveal interpretable patterns also serve as precise control interfaces. In fine-grained classification, LLM-based evolutionary optimization discovers discriminative features in interpretable language bottlenecks, enabling targeted control over classification decisions. In motion domains, bidirectional models learn interpretable muscle activation patterns that both explain motion generation and enable conditional editing based on muscle activity goals. In visual domains, counterfactual generation discovers fine-grained discriminative features through direct image editing, revealing subtle differences between visually similar groups while enabling modifications beyond the reach of language prompts. Together, these methods demonstrate how representational constraints can transform opaque machine learning systems into interpretable, controllable frameworks applicable to domains where the features and control objectives worth discovering are not known in advance.

Table of Contents

Acknowledgments xii

Chapter 1: Introduction and Background 1

1.1 The Discovery-Control Gap 1

1.1.1 Evolution of Control Methods 2

1.1.2 Multimodal Applications 4

Chapter 2: Muscles in Action 5

2.1 Related Work 7

2.2 The Muscles in Action (MIA) Dataset 9

2.2.1 Data Collection 9

2.2.2 Data Preprocessing 10

2.3 Method 11

2.3.1 Muscle and Motion Mappings 11

2.3.2 Modification in Muscle Space 12

2.3.3 Architectures 12

2.3.4 Conditioning 14

2.4 Experiments 15

2.4.1 Baselines 15

2.4.2 Results 15

2.5 Editing 18

2.6 Conclusion 19

Chapter 3: Evolving Interpretable Visual Classi ers with Large Language Models. . . 20

3.1 Related Work 22

3.2 Discovering Visual Classi ers 23

3.2.1 Model 23

3.2.2 Learning and Optimization 25

3.2.3 Evolutionary Search 26

3.2.4 Classi er Bank Initialization 27

3.3 Experiments 28

3.3.1 Datasets 28

3.3.2 Baselines 30

3.3.3 Fine-Grained Classi cation (iNaturalist) 33

3.3.4 Novel Objects (Kiki-Bouba) 34

3.3.5 Analysis of Learning 35

3.3.6 Auditing Dataset Bias 36

3.4 Discussion 37

Chapter 4: Teaching Humans Subtle Differences with DIFFusion. 38

4.1 Related Work 40

4.2 Method 42

4.2.1 Diffusion Preliminaries 42

4.2.2 DIFFusion 43
4.3 Experiments 47
4.3.1 Datasets and Baselines 47
4.3.2 Editing Results 48
4.3.3 Teaching Results 50
4.3.4 Method Analysis 51
4.3.5 Visualizing Dataset Bias 52
4.4 Discussion and Limitations 53

Chapter 5: Discussion and Conclusion 54
5.0.1 Commonalities and Differences Across Domains 54
5.0.2 The Visual Domain Challenge 55
5.0.3 Limitations and Future Directions 55

References 56

Appendix A: LLM-Mutate 72
A.1 Full Table 72
A.1.1 Training Details 72
A.1.2 Time Complexity 73
A.1.3 Qualitative Results of our method, CBD, and Gradient-Based Approach . . 74

Appendix B: Muscles in Action 75
B.1 Qualitative Examples: Motion-to-Muscle 75
B.2 Qualitative Examples: Muscle-to-Motion 75

B.3 Qualitative Examples of Editing 75

B.4 Architecture Implementation Details 76

B.5 Conditioning Implementation Details 77

B.6 Electrode/Sensor Placement 77

B.7 sEMG-Video Alignement (Further Details) 77

B.8 Dataset Statistics 78

Appendix C: DIFF-usion 81

C.1 Implementation Details 81

C.1.1 Baselines Implementation Details 81

C.1.2 User Study Details 82

C.1.3 Training Details 82

C.1.4 Datasets 84

C.2 Additional Experiments 84

C.2.1 SR vs. LPIPS Curves 84

C.2.2 Results w/o Domain Tuning 85

C.2.3 Perfect Inversion 85

C.2.4 Varying Dataset Size Results 86

List of Figures

2.1 Bidirectional Mapping between Muscles and Motion. Visible human motion is created by, and constrained by, our muscles. We learn to predict which muscle groups a person uses during motion (left column), as well as to reconstruct motion from muscle activation (right column). We illustrate how these two mappings can be combined to recommend new motions, similar to the input muscle, this time subject to particular muscle group goals (bottom row). 6

2.2 Sensor Placement. We illustrate the placement of our 8 sEMG sensors on a subject. We label the 8 measured muscles. 7

2.3 The MIA Dataset. We illustrate two canonical frames from our dataset for each of the 15 exercises. 8

2.4 Encoder and Decoder Architectures. We illustrate the architectures for our Motion-to-Muscle encoder and our Muscle-to-Motion decoder. 10

2.5 Motion-to-Muscle Qualitative Results. We illustrate two separate qualitative results. The rst row of frames corresponds to a visualization of the predicted activations, and the second row of frames corresponds to a visualization of the ground truth activations. For the plot beneath the frames, the dotted line corresponds to the ground-truth values, and the solid line corresponds to the predictions. Yellow corresponds to relaxed, and red corresponds to exed. 13

2.6 Muscle-to-Motion Qualitative Results. We show reconstructed 3D pose and 2D pose from muscle activity, where 3D pose is normalized with respect to a bounding box, while the 2D pose is absolute with respect to the image frame. Since sEMG signals don't contain location information, the 2D model cannot reconstruct the subject in the right location. 14

2.7 Qualitative Results. We illustrate three more qualitative results for the editing task. The rst row illustrates the ground truth 3D skeleton projected onto the frame. The second row shows the predicted muscle activation for the dorsal muscles (laterals and hamstrings). Since we are visualizing the back of the person, the meshes are ipped. The third row shows the scaled predicted muscle activations. The fourth row illustrates the recommended motion which the decoder generates from the scaled predicted muscle activations. 17

3.1 Learning Interpretable Classi ers. Can we nd text attributes for a concept by looking at the images without their class names? LLM-Mutate is a framework that learns sets of maximally discriminative visual attributes per class without access to class names or any form of prior knowledge. 21

3.2 Method. LLM-Mutate is an evolutionary algorithm that learns sets of discrete language attributes per class. The mutation and cross-over operations– that introduce new parameter hypotheses– are replaced by a large language model that uses in-context learning over past attributes and their scores to iteratively generate better attributes. 24

3.3 Attribute evolution. We show examples of the attribute evolution for both the pre-training and joint-training stages of learning. At the beginning, the rst generated set of attributes have little to do with the class, and by the end of the joint-training, the learned attributes are speci c to the Greenleaf Manzanita. 27

3.4 Qualitative Results. We show qualitative results for the iNaturalist Lichen family and a KikiBouba dataset. The results illustrate two sample images per class and the learned attributes. The learned attributes for the Lichen hardly refer to color, as this is a common feature to all Lichen, and instead focus on structural properties. . 29

3.5 Predictions. We show three different prediction examples. For each example, we show our method's prediction (rst column), as well as classi cation by description [83] (CBD)'s prediction (second column), and CBD's attributes for the ground truth class (third column). For each column, we show the normalized probability per attribute. Below the input image, we show the probability distributions across classes for both our method and CBD. The results show that our learned attributes are more detailed and discriminative of the species within the family, compared to the description by classi cation (CBD) baseline. Furthermore, our method's class probability distributions tend to be more concentrated than CBD's. 32

3.6 Comparison of Attributes by Method. We show qualitative examples of our learned attributes, classi cation by description's attributes (CBD), and our gradient-based approach attributes. CBD often produces reasonable attributes, but they are not discriminative, resulting in poor recognition accuracy. Gradient-based methods often produce poor attributes due to optimization dif culties. 33

3.7 Con dence. We show the mean margin of the scores between the top and runner-up prediction, which measures the typical con dence of each model. 35

3.8 Dataset Bias. The squirrel tail is the only species in the family that lives in drought habitats, and the learned attributes are names of plants that live in the desert. The ability to explicitly audit bias is an advantage of our interpretable method. 36

4.1 DIFF usion Counterfactuals. We illustrate the counterfactual results from our methods on the Butter y dataset, the Black Hole dataset, and the Retina dataset. In the Butter y dataset, the Viceroy has a cross-sectional line (yellow), a smaller head with less dots (magenta), and more "scaley" dots (blue), compared to the Monarch. In the Black Hole dataset, SANE has more uniform wisps (yellow) and less of a prominent photon ring (blue) as compared to MAD, with these distinguishing features discovered through our method rather than known a priori. In the Retina dataset, normal retinas lack the horizontal line bumps (yellow) present in retinas with drusen. 38

4.2 DIFF usion method. Our method consists of four parts. (i) Inverting the real image with DDPM-EF to obtain noise maps. (ii) Performing conditioning space arithmetic using positive and negative embeddings obtained from the training set. (iii) Generation via diffusion sampling, starting from the inverted noise conditioning on the manipulated conditioning vector. (iv) Optional domain tuning, in which we ne-tune the diffusion model for domain adaptation. 43

4.3 Qualitative Results. We present our qualitative results, where each row corresponds to one direction of our binary datasets. The rst column contains the inputs, and each subsequent column contains the results from each baseline, with the last column containing the result from DIFF usion. In particular, the magni ed boxes in the magenta frame show that our method is able to pick up on small discriminative cues. For example, when converting from MAD to SANE, the whisps become ampli ed and more uniform in brightness, and when converting from Monarch to Viceroy, a cross-sectional line is added on the wing. Note: The value in the top left corner of each image represents the probability predicted by the oracle classi er, as explained in Section 4.3.2. Values below 0.5 are classi ed as class 0, and above as class 1. Table 4.1 contains the class names and order. 44

4.4 (a) Varying number of images. Average LPIPS vs. number of images used per class. LPIPS stabilizes around 50 images for most datasets, re ecting improved identity delity and subtle class-distinctive feature shifts with increased embedding samples. (b) Difference Overlay. We visualize the difference between the input image and the counterfactual from DIFF usion. From SANE to MAD we notice a highlighting of the photon ring (green). From MAD to SANE we notice that the ring becomes less pronounced (magenta), and wisps appear (green). 49

4.5 User Study Results. We plot the results from user studies across users who studied our counterfactuals, users who studied the best baseline counterfactuals, and users who studied unpaired images. For both Butterfly and Black Hole datasets, we observe that the users who studied our counterfactuals significantly outperformed the other groups. The violin plots illustrate the distribution of user percentages, where the width of each grey shape represents the density of data points. 51

4.6 Interpolation. By varying the manipulation scale $\lambda \in \{0.5, 0.0, -0.5, -1.0, -1.5\}$, we can adjust the manipulation strength, allowing for smooth interpolation between the two classes. Notably, when $\lambda = 0$, we can reconstruct the original image while preserving the classifier's probabilities. 52

4.7 Dataset Bias. DIFFusion can reveal dataset bias. Squirreltail-to-Canada Wild Rye shifts emphasize environmental backgrounds over plant traits, reflecting iNaturalist's contextual bias, while Dachshund-to-Corgi edits prioritize foreground dog features, highlighting variable bias impact. 52

A.1 Comparison of Attributes by Method. We show qualitative examples of our learned attributes, classification by description's attributes (CBD), and our gradient-based approach attributes. CBD often produces reasonable attributes, but they are not discriminative, resulting in poor recognition accuracy. Gradient-based methods often produce poor attributes due to optimization difficulties. 74

B.1 Box Plot of Maximum sEMG Value per Muscle. We visualize the maximum sEMG value per muscle, across subjects. We notice that there is a significant range of values per muscle across subjects. 79

B.2 Nearest Neighbor. We perform visual clustering by taking advantage of the exercise labels in the dataset as well. We notice that the kicks and punches are clustered together, as well as the aerobic exercises (running and shuffle), and the strength training exercises (side lunges and squats). 80

C.1 Success Ratio (SR) vs. LPIPS curves. As discussed in Section C.2.1, we fix the guidance scales for each method, and the manipulation scale λ follows according to the implementation details in Section 4.2.2. We then vary η in increments of 0.1 within the range $\eta \in [0.0, 0.9]$, where η represents the percentage of timesteps skipped relative to the total denoising steps. 84

C.2 Perfect Inversion. We choose our inversion and sampling guidance scales by first reconstructing the original image with CFG. Then, we use these guidance scales for steering. 86

C.3 Varying Number of Images for AFHQ [191]. 88

C.4 Varying Number of Images for KikiBouba [192]. 89

C.5 Varying Number of Images for Butter y [189]. 90

C.6 Varying Number of Images for Black-Holes. 91

C.7 Varying Number of Images for Retina [190]. 92

List of Tables

2.1 RMSE per Exercise for the Encoder. We report the rMSE per exercise for muscle prediction. 16

2.2 RMSE per Exercise for the Decoder. We report the rMSE per exercise for motion prediction. 16

2.3 Temporal Analysis. We report the root mean squared error for the conditional baseline as well as for our conditional model as we change the length of the sequences. 16

3.1 We report the accuracy per dataset, for our method and baselines. 31

3.2 Accuracy across the hierarchical iNaturalist 26-class dataset. Rows represent the method for coarse-level classi cation, and column represents the method for ne-grained level classi cation. 31

4.1 Datasets and their classi cation tasks. 47

4.2 Performance comparison across datasets. SR = Success Ratio, LPIPS = Perceptual Distance. In bold are the best results, and in underline are the second-best results. . 48

4.3 User Study Results - Mean Accuracy (%) 50

A.1 We report all of the accuracies per dataset, for our method and baselines, across various class name types and with/without templates. * denotes the results that are averaged over the CLIP predetermined templates. 72

C.1 Results without Domain Tuning. We evaluate our method without ne-tuning on each dataset, measuring performance using Success Ratio (SR) and perceptual distance (LPIPS). Compared to Table 4.2, we observe substantial improvements in both SR and similarity for the Retina and Butter y datasets, as well as noticeable gains in reconstruction for Black-Holes and KikiBouba. However, the impact is minimal on the most natural-image datasets, AFHQ and CelebA-HQ. This suggests that the prior learned by the pre-trained diffusion model is strong enough to generate meaningful edits for these datasets without additional ne-tuning. 85

k

Acknowledgements

I never imagined I would do a PhD. In 2017, I wanted to start a clothing brand with my close friend Marianne Demarolle, using the hybrid image illusion I'd recently stumbled upon. I emailed Jianbo Shi for help, copying my monolithic attempt at its' implementation in MATLAB directly into the email, with the con dence only ignorance provides. He opened that email while strolling through Venice at ICCV with computer vision researchers, who would eventually become my academic family. After a summer of research, Jianbo planted the idea of a PhD in my mind, and recommended reading Carl Vondrick's work. I applied that fall.

The PhD is often described as solitary, but mine was collective from start to nish. Without structure, I found my community became the scaffolding holding everything together. They are all re ected in these papers: as coauthors, friends and family who became dataset contributors, thoughtful collaborators during problem-solving, or reviewers who read every paper thoroughly. Even my cat Jupiter and family dog Sherlock made it into the gures.

When I look back at the research, I see all of you.

Thank you to my family: Maman, Papa, Mamie, Papie. Your care, support and in uence on me is immeasurable. To my brother Ian, who became one of my closest (and funniest) friends during this journey. To Tonton Jean-Michel Chiquier and Tonton Albert LeFloch for instilling my taste for science. To Tata Véronique, Sébastien, and Solène for endless care. To my god-father Armand Ajdari, Gabrielle de Jasay, and their daughter, my god-daughter, Adélaïde.

Thank you to Jianbo Shi, for taking a chance and providing friendship and guidance throughout.

To my incredible PhD advisor Carl Vondrick, a true educator and one of the kindest, most brilliant people I've met. Thank you to Georgia Gkioxari, Katie Bouman, Jaakko Lehtinen, Lydia Chilton, and Victoria Mizzi for being exceptional mentors.

Thank you to Sonia Pearson, who is always there to problem solve from the most minor logistical questions to the biggest challenges, and for making me laugh at myself. Thank you for Jamaica, thank you for Marseille. Thank you for growing up with me. Thank you to Marianne Demarolle, for our biweekly home-cooked dinners and two-person book club that created such a deep feeling of family, and for being my partner in everything, from our beloved opera nights to NYC ferries and Rockaway Beach, from late nights in Paris at 19 to Brooklyn at 28. Thank you for your wit and unwavering presence. Thank you to Ella Bayi for your remarkable intuition and memory, for being my partner in both productive laziness and ridiculous humor. Thank you for the countless hours on couches and benches, and for how we inevitably lose all composure whenever we're together. Having you as a friend and neighbor for the past four years has made life in nitely better. Thank you to Alix Pouchol, for the thoughtful advice and practical wisdom, for our French vacations that ranged from winter chaos to summer bliss, and for bringing France to the city through raclette nights, mousse, crêpes, and galettes des rois.

Thank you to Serena Tsay for Taiwan, vegetables, and care. Ben Worsham for saving me from coyotes, feeding me, and creating community. David Vilalta for endless banter and books. Kaarish Shah Maniar for picnics and political theory. Sirine Kadi for Hamilton chats and absurd expressions. Kirill Chmel for blinis and peonies. Samir Gadre for fun research collaborations and city wanderings. Sruthi Sudhakar for morning debriefs and European adventures. Ruoshi Liu for wisdom on all fronts and inspiration to get Jupi! Basile Van Hoorick for pool cocktails and chats. Sachit Menon for shared existential crises and walks. Dave Epstein for grisaille and being my PhD companion from day one. Dídac Surís Coll-Vinent for being my academic big brother. Ege Ozguroglu for being my academic little brother. Mert Ketenci for looking out for one another. Jessie Burke for serene lunches by the pond. Natasha Deshko for the coffees, the wine, and everything in between. Olivia Aries and Vittoria Ciaraldi, for the distinguished Exeter trio,

renewed friendship, and the reminder that some connections transcend time and distance. Paola Martorell for depth and humor. Ellie Knaster for taste and remembering everything. Carmen Garcia Gallego for conversations in Paris, Lisbon, DC, Philly. Kevin Murning for synths and real-time audio experiments. Matthieu Crepy for exceptional cooking and shared music taste. Matt Goodman for being computer vision partners and friends since the early days.

Thank you to my lab-mates who have taught me so much. It has been a real pleasure working alongside such smart, kind, and open-minded people: Utkarsh Mall, Arjun Mani, Chengzhi Mao, Purva Tendulkar, Junbang Liang, Lennart Schulze.

Thank you to more faces and places that I love on campus: Vivian Liu, Martin Devaux, Simran Singh, Amitoj Singh, Loqman Salamatian, Ben Eyre, Nikita Rajaneesh, Todd Morrill, Jorge and Miriam from Vino Fino, Arts& Crafts, The Hamilton, Subconscious, the political science department for their happy hours.

Thank you also to William Kayat, Deniz Bes, Jimena Nestarez, Ariana McGinn, David Vidal, Paul-Julien Burg, Boyd Borjiet, Miloud Bessa , Linden Gledhill, Fabrice Cavarretta, Tana Cuturela, Gabriel Gilling, Raouf Taw k, Moni Arslan, Nicola Duemler, Leslie Crockett, Noah Shpak, Diana Farrell Pearson, and Alyosha Efros.

Thank you to the more recent New York faces I met and came to love: Manuela Montoya Arismendi, Pierre-Jean Quintin, Sonia Hussain, Tommasso Giulla, Merlin Heidemanns, Jonny Shore, Clara Berestycki, Paul Molander, Nico Lupo, Andrew Rodiguez Calderón.

Thank you to my cat Jupiter for doing absolutely nothing, perfectly.

Chapter 1: Introduction and Background

State-of-the-art machine learning models showcase extraordinary capabilities across diverse tasks: generative models produce photorealistic imagery and write complex code, while discriminative models demonstrate reasoning and classi cation performance that often exceeds expert human performance in specialized domains. This success creates opportunities to develop systematic methods for understanding and controlling these systems, yet doing so requires identifying the speci c features that drive effective performance. The objective is to develop methods that identify features which simultaneously reveal meaningful patterns in the data distribution and serve as control interfaces for directing systems toward desired outcomes. These features that double as controls are referred to as bottlenecks, meaning they impose interpretable structure into the model by construction. This opportunity becomes particularly compelling when we consider the vast computational capacity of modern systems. Generative visual models can produce endless image variations, motion generation systems can create diverse movements, and classi cation systems can separate categories across different modalities, presenting opportunities to discover which elements matter for particular goals while enabling precise control over outcomes. This thesis develops bottleneck methods that realize this potential by unifying discovery and control across motion, classi cation, and visual generation domains.

1.1 The Discovery-Control Gap

The typical paradigm for improving multimodal models' outputs today is prompt engineering, where the user iterates on the input prompt to improve results. However, this approach relies on trial and error. One alternative is post-hoc methods, which directly manipulate the model's internal representations rather than the input. A classic example for image editing is GANSpace

[1], which discovers salient editing directions in the latent space based on the dataset's distribution. This control method is useful for discovery because it highlights editing directions that could be important or difficult to describe. However, the user still needs to know which editing direction they want ahead of time, and they need to find it amongst the discovered directions. A similar limitation appears in discriminative tasks. Methods like GradCAM [2] can visualize a classifier's decision-making process through heatmaps that highlight relevant image regions. Yet the user still needs to know what to look for within the highlighted region to assess whether the classifier's decision-making is correct. As a result, both prompting and post-hoc methods share a deeper limitation: they require the user to know what they are looking to achieve in advance. This thesis addresses a more fundamental challenge: how can we build systems that help users discover what matters, not just control what they already know? We develop systematic methods that can both discover what drives effective performance across different tasks and provide precise control over those factors. We address this through architectural bottlenecks that force models to expose their decision-making process in structured, interpretable forms. This is critical in scientific domains where hypotheses must be discovered, educational settings where learning objectives emerge through exploration, and creative domains where goals evolve iteratively.

1.1.1 Evolution of Control Methods

Controlling powerful machine learning systems involves two interconnected challenges: identifying interpretable features and determining which features to prioritize for control. Existing approaches address these challenges in different ways, with each making distinct trade-offs. We can categorize methods by how they approach feature discovery and control:

Direct Control Methods: These approaches skip the discovery stage entirely, requiring users to manually specify control dimensions upfront. For generative tasks, this includes prompt engineering and manipulating conditioning mechanisms [3, 4]. For discriminative tasks, this encompasses manual feature engineering. These methods provide immediate control but offer no systematic discovery of which features matter most.

Supervised Control Methods These methods predetermine control dimensions during the speci cation stage, then learn to use them through supervised learning. For generative tasks, ne-tuning approaches [5, 6] optimize toward output characteristics de ned by training labels. For discriminative tasks, concept bottleneck models [7] learn to classify using pre-labeled features. By eliminating manual search, these approaches improve upon direct control, but they remain constrained to pre-speci ed dimensions rather than discovering what should be controlled.

Post-hoc Interpretability Methods: These methods separate the two stages explicitly: rst analyzing trained models to discover patterns, then applying those patterns for control. Early work demonstrated that GAN latent spaces contained semantic directions [1, 8], while modern implementations with diffusion models and CLIP [9, 10] can identify controllable directions through sophisticated analysis. However, this discovery-then-control pipeline still requires users to search through discovered features to determine which warrant prioritization.

Composable Methods These approaches come closest to unifying discovery and control by learning which prede ned components to compose. Function-based methods like DDSP [11] discover how to combine differentiable signal processing functions for audio generation. Structured approaches [12] learn to compose CAD-like geometric operations for 3D reconstruction. Discriminative methods like ViperGPT [13] learn to sequence visual reasoning modules for complex queries. While these methods can discover optimal component combinations, they remain constrained by predetermined function families or module vocabularies rather than discovering critical control dimensions directly from data distributions.

Our Approach: We develop methods that collapse the two-stage paradigm by making control dimensions emerge automatically from task necessity. By designing architectures where task-relevant information must ow through interpretable bottlenecks, we identify features precisely because they are critical for performance. This eliminates separate feature search or manual selection, unifying discovery and control through the architecture itself.

1.1.2 Multimodal Applications

To demonstrate the generality of uni ed discovery-control methods, this thesis presents three new bottleneck methods across distinct modalities, each exploring different aspects of integrated discovery and control.

We begin with physical motion systems, where bidirectional models with muscle activity as a bottleneck enable both motion generation and editing according to target muscle activations. By making muscle activity the bottleneck, we can systematically explore the relationship between neural motion control and physical movement while enabling applications in animation, rehabilitation, and robotics. This work establishes the foundational approach of using interpretable bottlenecks for objective-guided control. We will discuss this work in-depth in Chapter 2.

Building on these principles, we then address ne-grained visual classi cation tasks, where evolutionary approaches automatically discover the speci c features and decision rules that drive classi cation decisions. Rather than relying on predetermined bottlenecks, this work demonstrates how to discover the control dimensions themselves, in this case language bottlenecks, through objective-guided evolutionary search. This enables both understanding of what makes classi cation effective and precise control over classi cation behavior. We will discuss this work in-depth in Chapter 3.

Finally, we apply these discovery principles to visual generation domains, where we automatically discover key discriminative features from data distributions to control precise image edits. This work addresses both generative control (editing images) and discriminative control (identifying what visual features matter for ne-grained distinctions), demonstrating how objective-guided discovery can operate across both types of tasks within a single framework. The approach both discovers which visual elements drive ne-grained discrimination and provides practical control for image editing applications in design, marketing, and creative content creation. We will discuss this work in-depth in Chapter 4.

Chapter 2: Muscles in Action

The vision community has made great progress in modeling and analyzing human motion from video via tasks such as pose estimation [14, 15, 16, 17, 18, 19, 18], action recognition [20, 21, 22, 23, 24], motion transfer [25, 26, 27, 28] and more. However, motion understanding goes beyond the surface. Human motion is created by and constrained by muscles. Every action is a product of our brain sending electric signals to our nerves, which contract our muscles, in turn moving our joints. Although this process occurs within us, most of us turn to physical therapists and sports instructors for guidance on how to improve our motions to target or avoid particular muscle groups.

In this paper, we take a rst step towards building computer vision methods that represent the internal muscle activity that causes human motion. We present a system that, given a video of a person performing an action, learns to infer what muscles a person used. Walking is controlled falling, and any physical motion is a balance between muscle forces and gravity. This interplay leads to an inherent asymmetry: different muscles are engaged in the downward portion of a squat, for instance, than in the upward portion.

Our goal is to learn the complex relationships between physical forces by analyzing synchronized video and muscle activation data. We achieve this by developing a system that can predict muscle activity from motion, and vice versa. One application of this bidirectional system is generating new motions that are similar to an existing motion, while also adhering to speci c muscle recruitment targets, illustrated in Figure 4.1.

The typical method of measuring muscle activity is through the use of electromyography sensors, which exist in an invasive form, as well as an non-invasive form, called surface electromyography (sEMG). We collected a new dataset, which we will release, that consists of over twelve hours of synchronized single-view video and sEMG signals of eight muscles for ten subjects performing fteen different physical activities. By using commodity cameras and inexpensive sEMG

Figure 2.1: Bidirectional Mapping between Muscles and Motion Visible human motion is created by, and constrained by, our muscles. We learn to predict which muscle groups a person uses during motion (left column), as well as to reconstruct motion from muscle activation (right column). We illustrate how these two mappings can be combined to recommend new motions, similar to the input muscle, this time subject to particular muscle group goals (bottom row).

sensors, we make the problem practical and easy for others to build on. The eight muscles recorded are the left and right biceps brachii (biceps), the left and right latissimus dorsi (laterals), both quadriceps (quads), and both biceps femoris (hamstrings), denoted in Figure 2.2.

The primary contribution of this paper is a framework for modeling the association between human motion and internal muscle activity in video, and the rest of the paper will explain this contribution in detail. In section 2, we brie y review related work in human activity analysis, conditional motion generation, multi-modal learning, electromyography and physics-grounded human motion generation. In section 3, we describe our multimodal dataset in detail and analyze its characteristics. In section 4, we present a method to learn a bidirectional representation between the visual and muscle modalities. Section 5 shows experiments on both in-distribution experiments and subjects, as well as out-of-distribution experiments and subjects. In section 6, we showcase a demo application for learning the bidirectional representation between modalities. By releasing

our datasets and models publicly, we hope this paper will spur additional work that models the rich internal structure that drives human activity in video.

Figure 2.2: Sensor Placement. We illustrate the placement of our 8 sEMG sensors on a subject. We label the 8 measured muscles.

2.1 Related Work

Human Motion Prediction from Video. The field of computer vision has seen tremendous progress in inferring information about human motion from monocular video. One of the tasks is to regress human pose from video by regressing skeleton key-points and meshes [14, 15, 16, 17, 18, 19, 18]. A related task is action segmentation [29]. Other tasks that span from the pose estimation results include human motion transfer [25, 26, 27, 28] and even pose correction to make a given pose anatomically-correct [30, 31]. Closely related, Park et al. predict the 3D gravity direction from a moving first-person view using inverse dynamics [32].

Conditional Human Motion Generation. The field of conditional human motion generation is well-established, with a diverse set of conditioning mechanisms. There are works that condition based on past frames and/or future target frame(s) [33, 34, 35, 36, 37, 38, 39, 40, 41]. Others condition based on other modalities such as spatial trajectory [42], action class [43, 44, 41], natural

Figure 2.3:The MIA Dataset. We illustrate two canonical frames from our dataset for each of the 15 exercises.

language text [45, 46, 41], as well as audio [47, 48]. In this work, our motion generation is conditioned on an input motion, as well as muscle activity constraints.

Multi-Modal Representations. Multi-modal learning with video is a long-standing problem in computer vision. Some works predict other modalities from video, such as sound [49, 50] by using the natural temporal correspondence between video and sound, as well as video captioning [51]. Beyond prediction, multi-modal learning has been shown to improve video representations, for example from sound [52, 53], and language [51, 54]. The converse has also been explored - leveraging large datasets of unlabelled video has improved representations of other modalities [55, 53, 56, 57]. We are interested in predicting an entirely different modality from monocular video, muscle activation, as well as reconstructing motion from muscle activity.

Electromyography. To measure muscle activation, we use Surface Electromyography (sEMG) sensors, which are attached to electrodes placed on human skin to measure the electrical activity of muscle tissue. Reconstructing parts of human pose from sEMG data is an established task, but only for either arms [58, 59, 60, 61] speci cally, or legs [62] separately. In the forward direction, previous work has tackled predicting muscle activation, however, the input modality has not been video. Some works use torque or surface force measurements as input [63, 64, 65], while others

use goniometers to track pose [66] or motion capture tracking systems [67, 68, 69]. Our work seeks to predict muscle activation with no additional hardware at test-time besides video. Additionally, certain works predict muscle activation directly from 3D point clouds collected by depth cameras [70, 71]. However, these works rely on seeing the skin deformation on the human subject to infer muscle activity. We infer muscle activity from motion priors, not the visible increase in muscle size. This allows our model to work for clothed humans, or humans exercising at a distance.

Modeling Human Motion with Physics. Recent work has focused on generating motion that respects the physics of motion via physics simulations of human motion dynamics [72, 73, 74]. However, the simulated humanoids are constructed with assumptions about the physics of human motion. Additionally, sEMG studies have shown that for a given motion, different people vary in how they recruit muscle groups to execute that motion [75]. Single humanoid simulations will not capture this diversity in real humans' motion dynamics.

2.2 The Muscles in Action (MIA) Dataset

To explore the mapping between visual motion and muscle activity, we collected a dataset of synchronized video and sEMG signals. Our dataset contains 15 different exercises, which each of the 10 subjects perform.

2.2.1 Data Collection

Our dataset consists of 12.5 hours of synchronized video and sEMG signals, for eight muscles. These eight muscles include the left and right biceps brachii (biceps), the left and right latissimus dorsi (laterals), both quadriceps (quads), and both biceps femoris (hamstrings). The collected sEMG values correspond to the neuromuscular junction's total bioelectric energy.

The dataset consists of 15 exercises shown in Figure 2.6. Each subject performed each exercise for 5 minutes, and we asked them to vary the execution's speed, effort, and orientation. There are a total of 10 subjects in the dataset, 5 of which are females and 5 of which are males. We collected 75 minutes of data for each subject, totalling 12.5 hours of data. The subjects varied in body weight

Figure 2.4:Encoder and Decoder Architectures.We illustrate the architectures for our Motion-to-Muscle encoder and our Muscle-to-Motion decoder.

and muscle. To collect the sEMG data, we used eight M40 Muscle Sense bluetooth wireless EMG sensors from ANR Corp. To collect the video, we used a standard iPhone 10 camera. Please see the supplementary for details on the electrode and sensor placement method.

2.2.2 Data Preprocessing

The sEMG data's sample rate is 10fps, and the each data point from the sensor comes with a timestamp. The iPhone video records at 29.97 fps, and also has a time-stamp. We resample the video to match the frame rate of the sEMG data, and use these time stamps to align the muscle and visual modalities. We explain our exact methodology for this in the supplemental material.

Once the sEMG and the frames are aligned, we extract both 3D keypoints and 2D keypoints with the VIBE model and checkpoints [15]. The 3D keypoints are normalized with respect to a pre-computed bounding box, while the 2D keypoints are absolute with respect to the frame dimensions. For all experiments unless explicitly stated otherwise, the input sequence length is 30 frames and the output sequence length is 30 sEMG values per muscle, corresponding to 3 seconds. Once the dataset was split into intervals of 3 seconds, the train/test split was created by randomly choosing

20% of the 3 second intervals within an exercise per subject to be allocated to the test set, and the remaining 80% was allocated to the training set.

2.3 Method

Our approach aims to learn the bidirectional mapping between the visual modality and the muscle modality, which allows us to perform three tasks: a) infer muscle activity from video, b) infer pose from muscle activity, and c) provide recommended motions to people that will target certain muscles. In this section, we present this approach.

2.3.1 Muscle and Motion Mappings

The characteristics of muscle activity make the sEMG signal challenging to analytically process. We aim to overcome these challenges by leveraging the synchronization with the visual modality. By nding the correlations between a person's visible motion and the sEMG signal, we can learn representations that encode muscle activity with respect to motion.

Let G2 R) be the human pose of a person, extracted over keypoints, with dimensionality , for) frames in a video. Our goal is to predict the muscle activity that created the motion, which we denote as < 2 R") for " individual muscles, as well as to reconstruct motion G2 R) from muscle activity < . We aim to learn mappings that transform between these spaces through the functions:

$$<^{\wedge} = \backslash^{1}G^{\circ} \quad \text{and} \quad G^{\wedge} = |^{1}<^{\circ} \tag{2.1}$$

where $\backslash^{1}G^{\circ}$ is an encoder parameterized by and is a decoder parameterized by both of which are neural networks whose architecture we describe later. We learn the parameters for both models through the supervised learning problem:

$$\min_{\backslash,|} E_{1G,<} \, » L^{1} \backslash^{1}G^{\circ} - <^{\circ}, L^{1} |^{1}<^{\circ} - G^{\circ} ¼ \tag{2.2}$$

where we use a mean squared loss function to compare predictions to the ground truth in both modalities. We optimize both using stochastic gradient descent with the Adam optimizer [76]. Full implementation details are provided in the supplemental material.

2.3.2 Modi cation in Muscle Space

In this section, we explain how our bidirectional model can be used to generate new motions based on the edits in the muscle modality. Given a goal to minimize a use of the muscle, or increase the workout of a muscle, we generate a new motion, similar to the input motion, with a modi cation that adheres to the muscle activity goal. To do so, given a video, our encoder predicts muscle activation $\hat{<} \in R^{"\ )}$, composed of " sequences. Let $\hat{<}_: \in R^{)}$ be one muscle sequences in particular that we choose to scale, either up or down, with a scalar $B \in R$.

$$<_: = B\ \hat{<}_: \quad \text{and} \quad <_9 = \hat{<}_9\ 8_{9<:} \tag{2.3}$$

The new matrix $< \in R^{"\ )}$ is the edited $\hat{<}$ matrix. Our decoder decodes $<$ into a recommended motion G

$$G = {}^{1}<^{o} \tag{2.4}$$

This recommended motion G will be similar to the predicted reconstruction $\hat{G}$, except the recommended motion is in agreement with the muscle goals dictated by the edited predicted muscle activation $<$.

2.3.3 Architectures

We use a common architecture for both the encoder and decoder, with only minimal modi cations between them to adapt to their input and output modalities. See Figure 2.4 for an overview

Figure 2.5: Motion-to-Muscle Qualitative Results. We illustrate two separate qualitative results. The first row of frames corresponds to a visualization of the predicted activations, and the second row of frames corresponds to a visualization of the ground truth activations. For the plot beneath the frames, the dotted line corresponds to the ground-truth values, and the solid line corresponds to the predictions. Yellow corresponds to relaxed, and red corresponds to flexed.

of both architectures. We factorize the architectures:

$$\mathbf{G} = \mathbf{F}_2 \circ \mathbf{L}_1 \mathbf{G} \quad \text{and} \quad \mathbf{H} = \mathbf{F}_2 \circ \mathbf{L}_1 \mathbf{H} \tag{2.5}$$

where $\mathbf{L}$ is a local feature extractor and $\mathbf{F}$ is a global feature network.

The local feature extractor for the Motion-to-Muscle encoder receives keypoints $\mathbf{G} \in \mathbb{R}^{(\cdot)}$, where $\cdot$ is the number of keypoints, $\cdot$ is their dimensionality, and $\cdot$ is the number of frames. This matrix is then convolved with a filter $\mathbf{W}$ that has $\cdot$ channels:

$$\mathbf{L}_1 \mathbf{G} = \mathbf{W} * \mathbf{G} \tag{2.6}$$

The spatial dimension of the kernel $\mathbf{W}$ spans the entirety of the key-point dimension, and the temporal dimension spans roughly a second of time. Each kernel outputs a feature $\mathbf{F}$, where $\mathbf{F} \in \mathbb{R}^{(\cdot)}$. Since there are $\cdot$ channels, the resulting output from the temporal convolution layer is a sequence of $\cdot$ embeddings $\mathbf{z}_1, \ldots, \mathbf{z}_{\cdot}$ s.t. $\mathbf{z}_{\cdot} \in \mathbb{R}^{128}$.