NORTHWESTERN
UNIVERSITY

Computer Science Department

Technical Report
Number: NU-CS-2026-17

May, 2026

Ground, Reason, Align: Advances in Multimodal Foundation Models

Haosen Sun

Abstract

Scaling foundation models has reshaped multimodal AI, but three problems remain surprisingly underexplored: *efficiently locating* relevant moments in hour-long videos, *reasoning about* dynamic task progress from a single observation, and *reliably aligning* language models with human values at inference time. This thesis addresses each directly. The first contribution, **T***, reframes temporal search in long-form video as spatial search over grid images so that lightweight detectors can guide iterative zooming-in refinement; applied to GPT-4o and LLaVA-OneVision-OV-72B, T* improves accuracy from 50.5% to 53.1% and from 56.5% to 62.4% on LONGVIDEOBENCH with only 32 frames. The second contribution, **ProgressLM**, introduces PROGRESS-BENCH and a human-inspired two-stage reasoning framework for progress estimation in robotic manipulation; our trained model ProgressLM-3B surpasses GPT-5 (NSE: 13.8 vs. 18.9; PRC: 90.1 vs. 89.4) despite being over 100× smaller. The third contribution, **ODESteer**, presents a unified theoretical framework showing that activation addition is a first-order Euler discretization of an ODE and that steering directions correspond to barrier functions from control theory; ODESteer achieves improvements of 5.7% on TruthfulQA, 2.5% on UltraFeedback, and 2.4% on RealToxicityPrompts over *state-of-the-art* baselines. Together, these three works push toward multimodal AI systems that can efficiently search long video, reason about where a task stands, and stay aligned with human values during deployment.

Keywords

Long-form video understanding, Temporal search, Progress reasoning, Vision-language models, Activation steering, LLM alignment, ODE-based framework, Barrier functions

NORTHWESTERN UNIVERSITY

Ground, Reason, Align: Advances in Multimodal Foundation Models

A THESIS

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

MASTER OF SCIENCE

Field of Computer Science

By

Haosen Sun

EVANSTON, ILLINOIS

May 2026

ABSTRACT

Scaling foundation models has reshaped multimodal AI, but three problems remain surprisingly underexplored: finding relevant moments in hour-long videos, reasoning about dynamic task progress from a single observation, and keeping language models aligned with human values at inference time. This thesis addresses each directly, introducing new benchmarks, methods, and theoretical insights.

The first contribution introduces **T***, a temporal search framework for long-form video understanding. We formalize temporal search as the *Long Video Haystack* problem—identifying a minimal keyframe set from tens of thousands of candidates—and introduce **LVHaystack**, a benchmark of approximately 450 hours of video with 15,434 human-annotated instances. T* resolves the prohibitive cost of VLM-based search by reframing it as spatial search over grid images, enabling lightweight detectors to guide iterative, zooming-in refinement. Applied to GPT-4o and LLaVA-OneVision-OV-72B, T* improves accuracy from 50.5% to 53.1% and from 56.5% to 62.4% on LongVideoBench, with only 32 frames.

The second contribution presents **ProgressLM**, a framework for enabling vision-language models to reason about task progress in robotic manipulation. We introduce PROGRESS-BENCH, a benchmark of over 3,000 instances that systematically varies demonstration modality, viewpoint correspondence, and answerability. Inspired by human episodic retrieval and mental simulation, we develop both training-free and training-based approaches. Our trained model, PROGRESSLM-3B, surpasses GPT-5 on PROGRESS-BENCH (NSE: 13.8 vs. 18.9; PRC: 90.1 vs. 89.4) despite being over 100$\times$ smaller.

The third contribution proposes **ODESteer**, a unified theoretical framework for LLM alignment via activation steering. We show that conventional activation addition corresponds to first-

order Euler discretization of an ordinary differential equation (ODE), and that identifying a steering direction is equivalent to designing a barrier function from control theory. ODESteer employs multi-step, adaptive activation updates guided by a nonlinear barrier function, achieving consistent improvements of 5.7% on TruthfulQA, 2.5% on UltraFeedback, and 2.4% on RealToxicityPrompts over state-of-the-art baselines.

Together, these three works push toward multimodal AI systems that can efficiently search long video, reason about where a task stands, and stay aligned with human values during deployment.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of many people, and I am grateful to all of them.

I am most grateful to my advisor, Professor Manling Li. Over the past year and more, working alongside her has taught me what it means to be a researcher. Her enthusiasm for scholarship and her commitment to rigorous, responsible work are an inspiration I will carry throughout my career. Through her guidance, I have learned to appreciate the craft of research and to face unfamiliar problems with patience and confidence. I could not have asked for a more thoughtful or dedicated advisor. I am also grateful to Professor Chi-Keung Tang, who first set me on the path of research. He is a kind and patient mentor, and his belief in me from my undergraduate years gave me the foundation on which everything else has been built. I thank Professor Huajie Shao for his help and guidance during the application season; his approach to research has deeply influenced how I think about my own work. I also thank Professor Han Liu for serving on my thesis committee; his insights and engagement strengthened this thesis.

I feel fortunate to have been surrounded by wonderful friends throughout this time. Their companionship and support have meant more than I can say.

Most importantly, I want to thank my parents. Their love and steadfast support have been my greatest source of strength. Thank you for always encouraging me to pursue my dreams.

List of Abbreviations

AFRR	Answerable False Rejection Rate
AI	Artificial Intelligence
CAA	Contrastive Activation Addition
CoT	Chain-of-Thought
FLOPs	Floating Point Operations
GRPO	Group Relative Policy Optimization
ITI	Inference-Time Intervention
LLM	Large Language Model
NSE	Normalized Score Error
ODE	Ordinary Differential Equation
PRC	Progress Rank Correlation
QA	Question Answering
RL	Reinforcement Learning
RepE	Representation Engineering
SFT	Supervised Fine-Tuning
SOTA	State of the Art
UDA	Unanswerable Detection Accuracy
VLM	Vision-Language Model
VQA	Visual Question Answering

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Large-scale foundation models have changed what AI can do. Vision-language models (VLMs) now describe images, answer visual questions, and follow complex instructions. Large language models (LLMs) reason and generalize across tasks at a level that would have seemed unlikely a few years ago. But three problems have proven stubborn: efficiently finding information in hour-long videos, enabling models to reason about dynamic processes over time, and keeping language models aligned with human values. This thesis addresses all three.

1.1 Motivation

Consider what these systems need to do in practice. A robot assistant has to estimate how far along a task is from a single visual observation. A video QA system has to find the relevant frames out of tens of thousands. A deployed language model has to stay helpful, truthful, and non-toxic even when pushed. Each problem sits at a different level of the AI stack, but they share the same basic constraint: useful behavior without unreasonable cost.

The three works in this thesis share an approach: each replaces brute-force or opaque methods with something grounded in an existing, well-understood framework—spatial search, cognitive models of reasoning, and control theory. The problems are different; the strategy is not.

1.2 Overview of Contributions

T*: Efficient Temporal Search for Long-Form Video Understanding

Long-form video understanding is bottlenecked by the prohibitive cost of processing thousands of frames with large vision-language models. Existing approaches either uniformly sample frames (missing relevant content) or apply VLMs frame-by-frame (computationally infeasible). Chapter 2 presents **T***, which reframes temporal search as spatial search: frame sequences are arranged into grid images and queried with lightweight object detectors, enabling iterative zooming-in refinement at a fraction of the cost of VLM-based search.

We also introduce **LVHaystack**, the first large-scale benchmark specifically designed for temporal search quality evaluation, comprising approximately 450 hours of video and 15,434 human-annotated instances. T* improves over strong baselines, including GPT-4o and LLaVA-OneVision-OV-72B, on LongVideoBench while using up to 4$\times$ fewer frames.

ProgressLM: Progress Reasoning for Vision-Language Models

While much prior work focuses on *what* is visible in an image, a fundamentally different and harder question is *how far along* a task has progressed. Chapter 3 presents **ProgressLM**, which studies progress estimation as a core reasoning capability for VLMs. Motivated by how humans reason about progress through episodic retrieval and mental simulation, we introduce structured prompting strategies and a training pipeline that produces PROGRESSLM-3B, a compact model that surpasses GPT-5 on our newly introduced PROGRESS-BENCH.

PROGRESS-BENCH is designed around three axes of difficulty: demonstration modality (visual vs. text), viewpoint correspondence, and answerability. Together they give a rigorous and interpretable evaluation framework for progress reasoning in robotic manipulation settings.

ODESteer: ODE-Based Activation Steering for LLM Alignment

Aligning large language models to exhibit desirable behaviors, such as truthfulness, helpfulness, and non-toxicity, is a central challenge in AI safety. Chapter 4 presents **ODESteer**, which provides a unified theoretical framework for activation steering based on ordinary differential equations (ODEs). We show that conventional activation addition is equivalent to a first-order Euler discretization of an ODE, and that identifying steering directions corresponds to designing barrier functions from control theory.

This theoretical insight leads to a practical method: ODESteer constructs an ODE guided by the gradient of a nonlinear barrier function (defined as the log-density ratio between positive and negative activations) and solves it numerically to perform multi-step, adaptive activation updates. Empirically, ODESteer achieves consistent gains of 5.7% on TruthfulQA, 2.5% on UltraFeedback, and 2.4% on RealToxicityPrompts over state-of-the-art baselines.

1.3 Thesis Organization

The remainder of this thesis is organized as follows:

- **Chapter 2** presents T*, covering the LVHaystack benchmark, the temporal search framework, and experimental evaluation on long-form video understanding.
- **Chapter 3** presents ProgressLM, covering the Progress-Bench benchmark, training-free and training-based progress reasoning approaches, and analysis of VLM failure modes.
- **Chapter 4** presents ODESteer, covering the ODE-based theoretical framework, barrier function interpretation of existing methods, and empirical results across multiple LLM alignment benchmarks.

- **Chapter 5** concludes the thesis and discusses future research directions.

CHAPTER 2

T*: EFFICIENT TEMPORAL SEARCH FOR LONG-FORM VIDEO UNDERSTANDING

2.1 Introduction

As video understanding research expands from seconds-long clips to hour-long recordings, a fundamental bottleneck emerges: how to efficiently identify the small subset of frames that matter for answering a given question. State-of-the-art vision-language models (VLMs) such as GPT-4o [1] and LLaVA-OneVision [2] require hundreds of tokens per frame, making exhaustive frame-by-frame processing of hour-long videos computationally prohibitive.

Existing approaches fall into two categories. *Uniform sampling* selects frames at fixed intervals regardless of content, frequently missing the relevant moments. *VLM-based search* methods [3], [4] iteratively query large models to select keyframes, but incur prohibitive inference costs. In this chapter, we present **T***, a lightweight temporal search framework that reframes temporal search as spatial search, and **LVHaystack**, a large-scale benchmark purpose-built for evaluating temporal search quality.

2.2 Task Formulation: The Long Video Haystack Problem

We formalize temporal search as the *Long Video Haystack* problem. Given a video $V = \{f_1, f_2, \ldots, f_N\}$ with N frames and a question Q, the goal is to find a minimal subset of k keyframes $V^K = \{f_1^K, f_2^K, \ldots, f_k^K\} \subseteq V$ satisfying two properties:

- **Completeness**: V^K must contain all information necessary to answer Q. If the correct answer from V is A, then the answer from V^K should also be A.

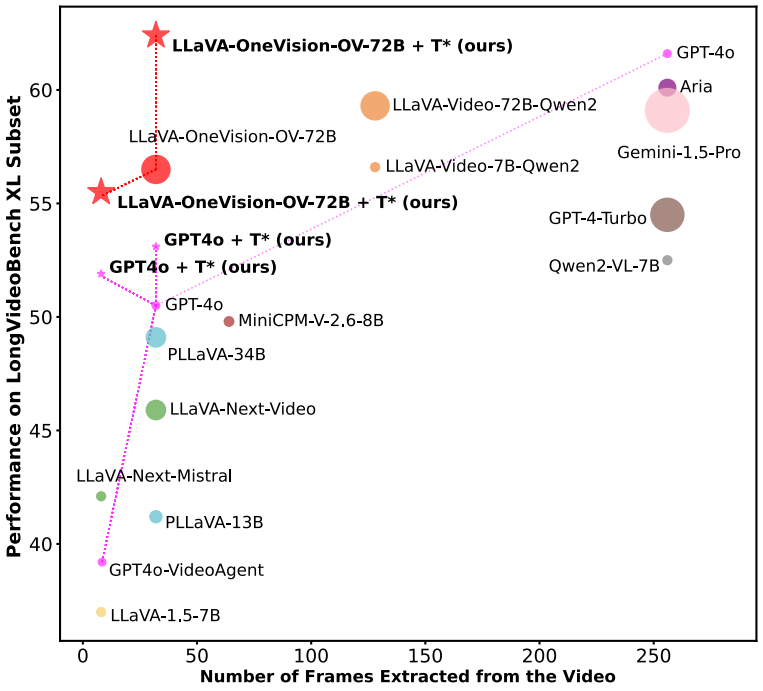

Figure 2.1: **Long-form video understanding performance on LongVideoBench XLong subset (15–60min).** Open-source model size is indicated by marker size. T* significantly improves SOTA models: GPT-4o from 50.5% to **53.1%** and LLaVA-OneVision-OV-72B from 56.5% to **62.4%**, both under a 32-frame budget.

- **Minimality**: V^K contains no redundant frames—every selected frame contributes essential information.

This formulation is analogous to the needle-in-a-haystack problem [5]: the "needles" are the handful of relevant keyframes scattered among thousands of "haystack" frames.

2.3 The LVHaystack Benchmark

Prior long-form video benchmarks [6], [7] focus on downstream task accuracy, providing no ground-truth annotation of which frames are needed. We introduce **LVHaystack**, the first benchmark designed specifically to evaluate temporal search quality.

Each instance in LVHaystack is a tuple $\langle V, Q, V^K, A \rangle$ with a video V, question Q, annotated minimal keyframe set V^K, and answer A. LVHaystack comprises two subsets:

- **LVHAYSTACK-EGO**: 15,092 instances from Ego4D [8] egocentric videos averaging 8.3 minutes per segment. Crowdworkers identified the minimal keyframe set needed to answer each question.
- **LVHAYSTACK-LV**: Instances from LongVideoBench [6] allocentric videos, with annotators verifying and refining original reference timestamps.

In total, LVHaystack covers approximately 450 hours of video with approximately 48 million frames. We propose temporal F_1 and visual F_1 metrics that compare model-selected frames against human-annotated keyframes within a 5-second temporal threshold.

2.4 The T* Framework

T* operates in three phases, illustrated in Figure 2.2.

Subset	LVHaystack-Ego	LVHaystack-LV
Video Type	Egocentric	Allocentric
# videos	988	114
# total length	423 h (25.7 min/video)	26.7 h (14.1 min/video)
# frames	45,700,000 (46,300/video)	2,200,000 (19,100/video)
# QA pairs	15,092 (15.3/video)	342 (3.0/video)
# keyframes	28,300 (1.9/question)	496 (1.5/question)

Table 2.1: **Dataset statistics of LVHaystack.**

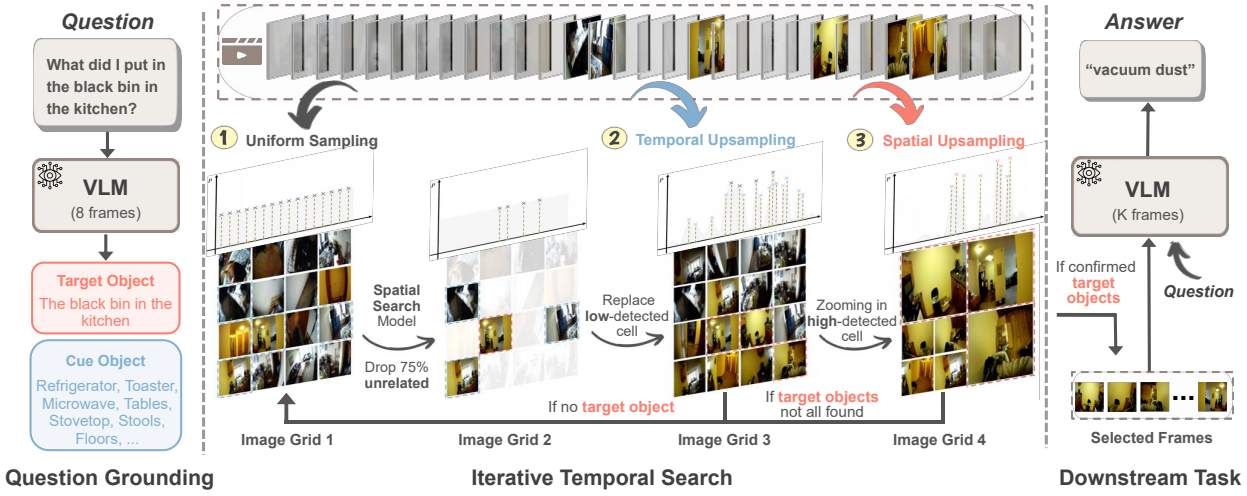

Figure 2.2: **The T* framework for efficient temporal search.** T* operates in three phases: (1) *Question Grounding*, where target and cue objects are extracted from the question; (2) *Iterative Temporal Search*, formulated as spatial search over grid images; and (3) *Downstream Task Completion*, where the top-K keyframes are passed to a VLM for answering.

2.4.1 Question Grounding

The first phase extracts semantic search targets from the question. N uniformly sampled frames $\overline{V_N}$ are passed to a VLM along with the question Q to identify:

$$\{T, C\} = \texttt{VLM}(\overline{V_N}, Q), \tag{2.1}$$

where T is the set of *target objects* (directly relevant to answering Q) and C is the set of *cue objects* (contextual indicators of where targets may appear). For example, for "What did I put in the black dustbin?", the target is the dustbin and cues include room corners and furniture arrangement.

2.4.2 Iterative Temporal Search

The core of T* is an iterative sampling-detection-reweighting loop. Frames are sampled from a probability distribution P (initialized as uniform), arranged into a $g \times g$ grid image G, and processed by a spatial detector:

$$I = \text{Sample}(P \odot N_v, g^2), \tag{2.2}$$

where N_v is a binary mask preventing re-sampling of visited frames. The detector scores each grid cell (i, j) as:

$$C_{i,j} = \max_{o \in \mathcal{D}_{i,j}} (c_o \cdot w_o), \tag{2.3}$$

where c_o is object detection confidence and w_o is object weight (1.0 for targets, 0.5 for cues). Detected keyframes are added to the result set F. The score distribution is updated with temporal locality propagation and spline interpolation, then renormalized. The loop continues until all targets are found or the frame budget B is exhausted.

2.4.3 Downstream Task Completion

The final K frames are selected by TopK(S) on the accumulated score distribution and passed to a VLM for question answering. This decouples the search phase from the answering phase, enabling T* to augment *any* VLM without modifying its parameters.

2.5 Experimental Results

2.5.1 Search Utility

We evaluate temporal search quality on LVHaystack using temporal F_1 at a 5-second threshold. As shown in Table 2.2, existing adaptive baselines top out at 2.4% temporal F_1 on LVHAYSTACK-LV even with 32 frames, and as low as 2.1% under tighter budgets, showing how difficult precise temporal localization is. T* using detector-based spatial search substantially outperforms all baselines in both temporal F_1 and downstream QA accuracy, with the 32-frame detector variant achieving the best overall performance.

2.5.2 Downstream Video Question Answering

Table 2.4 summarizes performance on long-form video QA benchmarks. T* achieves:

- GPT-4o: 50.5% $\rightarrow$ **53.1%** on LongVideoBench XLong with 32 frames
- LLaVA-OneVision-OV-72B: 56.5% $\rightarrow$ **62.4%** on LongVideoBench XLong with 32 frames

These improvements are achieved with up to 4$\times$ fewer frames compared to baselines and approximately 3$\times$ lower FLOPs compared to frame-by-frame VLM search.

Method	Frames↓	LVHaystack-Ego						LVHaystack-LV					
		Temporal			Visual			Temporal			Visual		
		Prec.↑	Rec.↑	F_1 ↑	Prec.↑	Rec.↑	F_1 ↑	Prec.↑	Rec.↑	F_1 ↑	Prec.↑	Rec.↑	F_1 ↑
Baselines: Static Frame Sampling													
Uniform [6]	8	1.0	3.4	1.6	58.0	63.0	60.2	1.4	6.3	2.2	56.0	72.0	62.7
Uniform [6]	32	1.1	14.8	2.0	58.5	65.6	61.5	1.4	24.9	2.7	58.7	81.6	67.3
Baselines: Adaptive Temporal Search													
VideoAgent [3]	10.1	1.7	5.8	2.7	58.0	62.4	59.9	1.2	8.5	2.1	58.8	73.2	64.7
Retrieval-based	8	1.2	4.2	1.9	58.5	61.7	59.9	1.5	6.3	2.3	63.1	65.5	64.1
Retrieval-based	32	1.0	13.8	1.9	58.5	65.4	61.4	1.3	21.8	2.4	**59.9**	80.8	**67.8**
Ours: T for Zooming-In Temporal Search*													
Attention-based	8	2.2	7.5	3.3	58.4	62.5	60.2	1.5	6.6	2.4	63.6	68.6	65.7
Training-based	8	1.4	4.9	2.1	58.0	61.5	59.6	1.5	6.6	2.3	59.8	71.1	64.5
Detector-based	8	1.7	5.8	2.7	63.8	70.1	66.8	1.6	7.1	2.5	58.4	72.7	64.3
Detector-based	32	**1.8**	**26.3**	**3.4**	**62.9**	**76.2**	**68.9**	**1.7**	**28.2**	**3.1**	58.3	**83.2**	**67.8**

Table 2.2: **Search utility results on LVHaystack.** 8-frame bests are underlined; 32-frame bests are **bold**. Detector-based T* achieves the best 32-frame performance across most metrics, demonstrating the effectiveness of visual grounding combined with iterative temporal search.

Method	Search Efficiency				Overall Task Efficiency		
	Grounding	Matching	TFLOPs↓	Latency (s)↓	TFLOPs↓	Latency (s)↓	Acc↑
Baselines: Static Frame Sampling							
Uniform-8 [6]	N/A	N/A	N/A	0.2	139.3	3.8	45.9
Baselines: Adaptive Temporal Search							
VideoAgent [3]	GPT-4×4	CLIP-1B×840	536.5†	30.2	690.7†	34.9	49.2
Retrieval-based	N/A	YOLO-world-110M×840	216.1	28.6	355.4	32.2	50.3
Ours: T for Efficient Temporal Search*							
Attention-based	LLaVA-72B×3	N/A	88.9	13.7	228.2	17.3	49.6
Detector-based	LLaVA-7B×1	YOLO-world-110M×49	33.3	7.5	172.6	11.1	50.8
Training-based	LLaVA-7B×1	YOLO-world-110M×38	**30.3**	**6.8**	**169.6**	**10.4**	**51.0**

Table 2.3: **Efficiency results on LVHaystack.** T* achieves strong performance with significantly less computation and lower latency than adaptive search baselines. (†) VideoAgent FLOPs exclude GPT-4 costs due to closed-source nature.

LongVideoBench						Video-MME					
		Video Length						Video Length			
Model	**#Frame**	**XLong** 15–60min	**Long** 2–10min	**Medium** 15–60s	**Short** 8–15s	**Model**	**#Frame**	**Long** 41min	**Medium** 9min	**Short** 1.3min	**Total** 17min
GPT-4o	8	47.1	49.4	67.3	69.7	GPT-4o	8	51.4	54.3	55.7	53.8
GPT-4o + T*	8	**51.9**	**52.4**	**72.7**	**70.0**	GPT-4o + T*	8	**55.9**	**57.3**	**56.4**	**56.5**
LLaVA-OV-72B	8	53.7	57.4	74.1	73.0	LLaVA-OV-72B	8	52.6	55.5	59.6	55.9
LLaVA-OV-72B + T*	8	**55.5**	**63.7**	**76.3**	**73.5**	LLaVA-OV-72B + T*	8	**57.7**	**57.5**	**61.7**	**59.0**
GPT-4o	32	50.5	57.3	73.5	71.4	GPT-4o	32	56.3	60.7	68.3	61.8
GPT-4o + T*	32	**53.1**	**59.4**	**74.3**	**71.4**	GPT-4o + T*	32	**59.3**	**63.5**	**69.5**	**64.1**
LLaVA-OV-72B	32	56.5	61.6	77.4	74.3	LLaVA-OV-72B	32	60.0	62.2	76.7	66.3
LLaVA-OV-72B + T*	32	**62.4**	**64.1**	**79.3**	**74.6**	LLaVA-OV-72B + T*	32	**61.0**	**66.6**	**77.5**	**68.3**

Table 2.4: **Downstream QA performance on LongVideoBench and Video-MME.** T* consistently improves GPT-4o and LLaVA-OneVision-72B across all video lengths and both benchmarks, under 8- and 32-frame budgets.

2.5.3 Shorter-Video Benchmarks: NExT-QA and EgoSchema

Table 2.5 evaluates T* on shorter-video benchmarks (NExT-QA at 0.7 min average, EgoSchema at 3 min average). Even in this regime where uniform sampling already performs well, T* achieves consistent improvements: LLaVA-OneVision-7B improves from 76.4% to **80.4%** on NExT-QA and from 63.6% to **66.6%** on EgoSchema with only 8 frames.

2.6 Analysis

Figure 2.3 visualizes the iterative search process. T* begins with a coarse uniform distribution and iteratively concentrates probability mass on temporal regions containing target objects. This coarse-to-fine strategy mirrors how humans scan long videos: first identifying approximate locations, then zooming in for confirmation.

Ablation studies (see Appendix A.2) show that the iterative distribution update mechanism is the primary driver of T*'s gains, with temporal locality propagation contributing additional benefit. Replacing the spatial detector with uniform scoring degrades performance to near-baseline levels,

Model	Frames	NExT-QA 0.7 min avg	EgoSchema 3 min avg
Baselines: Static Uniform Sampling			
InternVideo [9]	90	49.1	32.1
MVU [10]	16	55.2	60.3
LLoVi [11]	90	67.7	57.6
LangRepo [12]	180	60.9	66.2
LLaVA-OneVision-7B [2]	32	79.4	65.4
Baselines: Adaptive Frame Selection			
SeViLA [4]	32	63.6	25.7
VideoAgent [3]	8.4	71.3	60.2
LVNet [13]	12	72.9	66.0
VideoTree [14]	63.2	73.5	66.2
VidF4 [15]	8	74.1	—
Ours: Plug in T for Efficient Temporal Search*			
LLaVA-OneVision-7B [2]	8	76.4	63.6
+ T*	8	**80.4**	**66.6**

Table 2.5: **Results on NExT-QA and EgoSchema.** T* improves LLaVA-OneVision-7B on both shorter-video benchmarks with only 8 frames, outperforming prior adaptive methods.

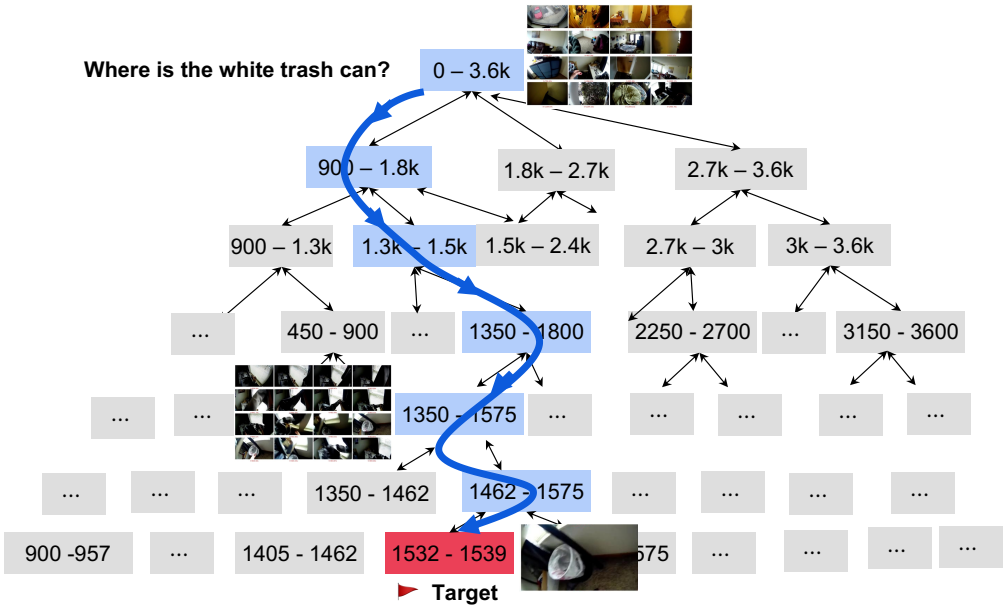

Figure 2.3: **Visualization of the iterative temporal search process.** T* progressively focuses on relevant temporal regions through zooming-in refinement, concentrating the frame budget on high-probability regions.

confirming that object-guided search is what drives the improvement. Scaling the grounding VLM from 7B to 72B parameters yields less than 1% improvement in QA accuracy; search quality is not bottlenecked by the grounding step.

2.7 Summary

T* shows that efficient temporal search does not require VLM-in-the-loop iteration. Reframing the problem as spatial search over grid images and using lightweight detectors for iterative distribution refinement, T* consistently improves long-form video understanding across multiple VLMs and benchmarks at a fraction of the computational cost.

CHAPTER 3

PROGRESSLM: PROGRESS REASONING FOR VISION-LANGUAGE MODELS

3.1 Introduction

Most vision-language research asks: *what is shown in this image?* A fundamentally harder and less studied question is: *how far along is this task?* Answering this requires the model to interpret a single observation not as a static snapshot, but as a moment situated within an ongoing, evolving process. This capability, *progress reasoning*, is central to robotic manipulation, procedural video understanding, and embodied AI more broadly.

Prior work on progress estimation either relies on task-specific regression models [16] or reformulates the problem as shuffle-and-rank [17] or pairwise comparison [18]. None of these approaches evaluate whether VLMs can acquire progress estimation as a *general reasoning capability* from a single observation.

This chapter presents **ProgressLM**, which addresses this gap through three contributions: (1) PROGRESS-BENCH, a systematic benchmark with over 3,000 instances; (2) a training-free prompting strategy inspired by human episodic reasoning; and (3) PROGRESSLM-3B, a compact trained model that surpasses GPT-5 on PROGRESS-BENCH.

3.2 The Progress-Bench Benchmark

3.2.1 Problem Formulation

Each instance in PROGRESS-BENCH consists of a task demonstration $\mathcal{D}$ and a single observation $\mathbf{o}$ sampled from an intermediate moment of execution. The demonstration describes the task

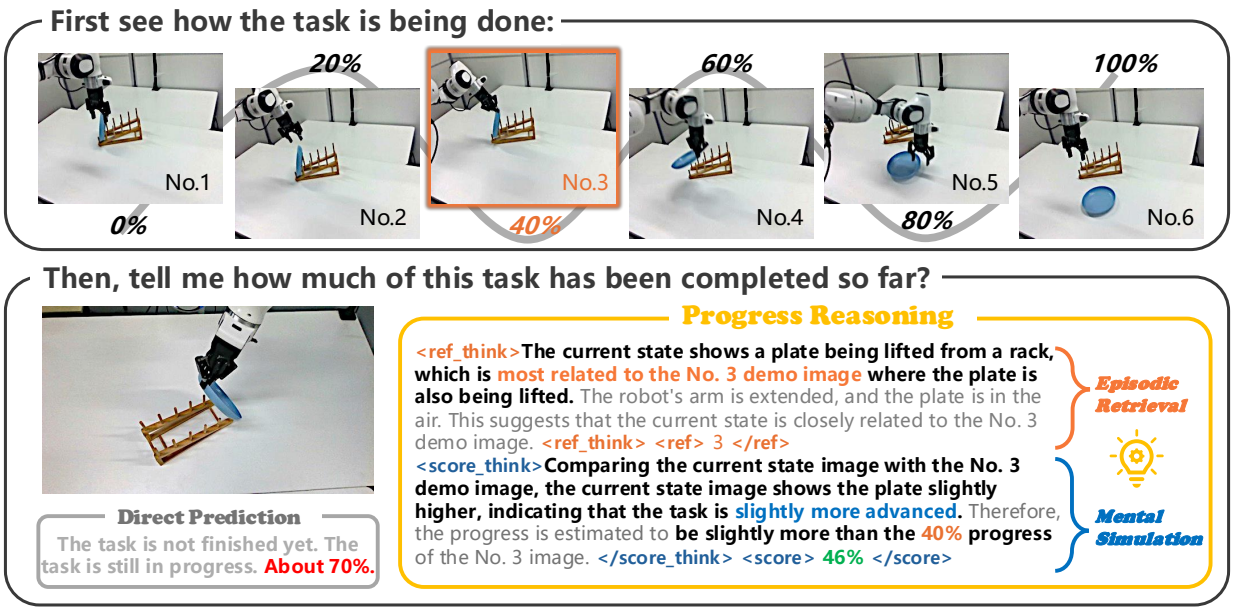

Figure 3.1: **Progress reasoning overview.** Given a task demonstration and a single observation, the model must predict how far the task has progressed. Our human-inspired two-stage reasoning framework—episodic retrieval followed by mental simulation—consistently outperforms direct prediction, especially with training.

from start to completion, presented either as a sequence of key frames or as stepwise text. Given $(\mathcal{D}, \mathbf{o})$, the model predicts a normalized progress score $p \in [0, 100\%]$. When the demonstration is insufficient or inconsistent for meaningful estimation, the model should output `N/A`.

3.2.2 Benchmark Construction

PROGRESS-BENCH is organized along three key axes of variation:

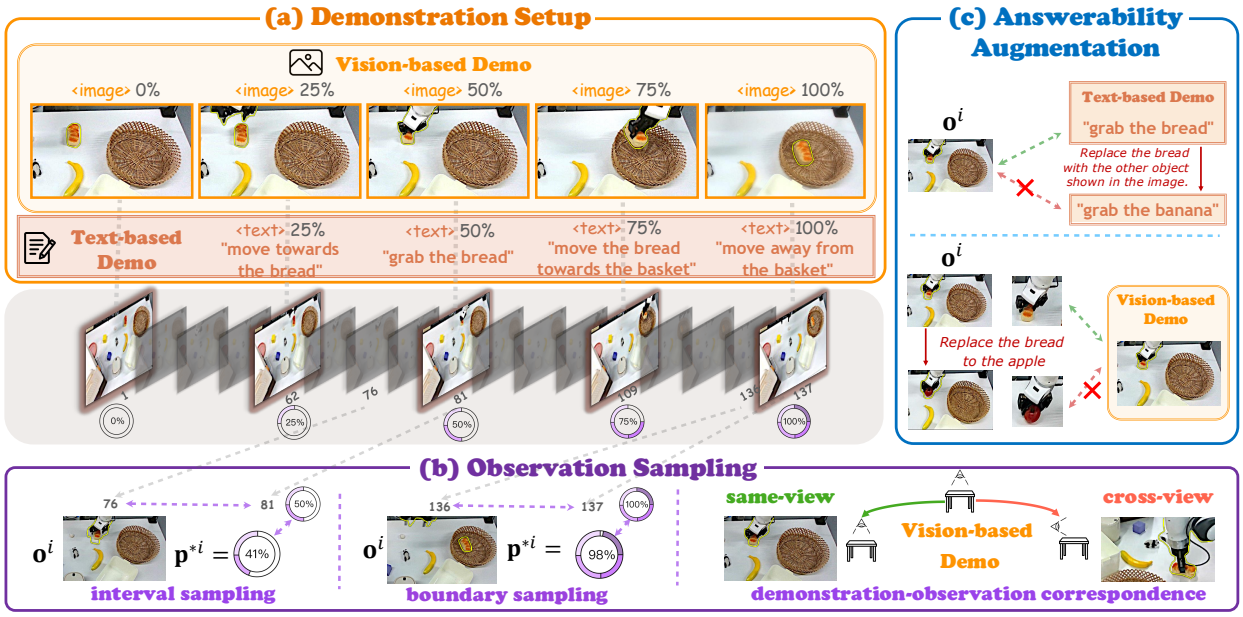

Figure 3.2: **PROGRESS-BENCH construction overview.** (a) Demonstrations in vision-based or text-based format; (b) observation sampling with viewpoint correspondence control; (c) answerability augmentation via semantic mismatch.

Demonstration Modality. We consider two settings. In the *vision-based* setting, demonstrations are sequences of key frames: $\mathcal{D}_v = \{(\mathbf{f}_{a_j}, \mathbf{p}_j)\}_{j=1}^{N}$. In the *text-based* setting, demonstrations provide action-only step descriptions: $\mathcal{D}_t = \{(\mathbf{t}_j, \mathbf{p}_j)\}_{j=1}^{N}$. Text-based demonstrations are harder because intermediate states are not directly observable.

Viewpoint Correspondence. Observations are sampled from the same camera view as the demonstration (*same-view*) or from a different viewpoint (*cross-view*). Cross-view settings require the model to abstract away from egocentric camera positions.

Answerability. Some instances are intentionally made unanswerable by introducing semantic mismatches between the demonstration and observation. The model must recognize these cases and output `N/A`.

Progress labels are assigned by linear interpolation between consecutive key steps:

$$\mathbf{p}^* = \mathbf{p}_j + \delta\,(\mathbf{p}_{j+1} - \mathbf{p}_j), \quad \delta \in (0, 1), \tag{3.1}$$

which is justified for smoothly executed expert demonstrations.

3.2.3 Evaluation Metrics

We report four complementary metrics:

1. **Normalized Score Error (NSE)**$\downarrow$: $\mathrm{NSE} = |\hat{p} - g| / \max(g, 1 - g)$, measuring pointwise prediction accuracy.

2. **Progress Rank Correlation (PRC)**$\uparrow$: Spearman rank correlation between predicted and ground-truth progress sequences along a trajectory, measuring temporal ordering consistency.

3. **Answerable False Rejection Rate (AFRR)**$\downarrow$: Fraction of answerable samples incorrectly labeled as unanswerable.

4. **Unanswerable Detection Accuracy (UDA)**$\uparrow$: Fraction of unanswerable samples correctly identified.

3.3 Towards Progress Reasoning in VLMs

3.3.1 Human-Inspired Two-Stage Reasoning

We model progress estimation as a two-stage process inspired by human cognition [19]:

1. **Episodic Retrieval**: Given the current observation, recall a representative moment from the demonstration that most closely resembles the current state as a coarse anchor.

2. **Mental Simulation**: Reason about how the task evolves from the retrieved anchor toward the current observation to produce a refined progress estimate.

3.3.2 Training-Free Approach

We instantiate this two-stage framework through structured prompting, without parameter updates. The output schema enforces four fields:

- `<ref_think>`: episodic retrieval reasoning,
- `<ref>`: the retrieved reference step,
- `<score_think>`: mental simulation comparing reference with current observation,
- `<score>`: final progress prediction.

3.3.3 Training-Based Approach: ProgressLM

We construct PROGRESSLM-45K, a training dataset of 45,000 instances drawn from robotic manipulation tasks that are fully disjoint from PROGRESS-BENCH, ensuring that all evaluation results reflect generalization to unseen tasks rather than memorization.

Cold-Start SFT. We fine-tune on PROGRESSLM-25K-COT, where each training instance provides ground-truth <ref> and <score>, and the model generates intermediate reasoning traces:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N}\sum_{i=1}^{N} \log P_\theta(\mathbf{r}^{i*} \mid \mathcal{D}^i, \mathbf{o}^i). \tag{3.2}$$

Reinforcement Learning. We apply GRPO [20] with a composite reward:

$$\mathcal{L}_{\text{RL}} = -\mathbb{E}_{\mathbf{r}\sim P_\theta}\big[\alpha R_{\text{format}} + \beta R_{\text{ref}} + \gamma R_{\text{score}}\big], \tag{3.3}$$

with $\alpha : \beta : \gamma = 1 : 6 : 3$. The reward structure encourages structured outputs, accurate reference retrieval, and precise progress estimation.

3.4 Experimental Results

We evaluate 14 VLMs on PROGRESS-BENCH, including GPT-5 and GPT-5-mini, and models from the Qwen and InternVL families ranging from 2B to 72B parameters. For each model, we consider direct prediction, our training-free prompting approach, and for the 3B-parameter Qwen2.5-VL-3B backbone, the full training pipeline (PROGRESSLM-SFT and PROGRESSLM-RL).

Table 3.1 reports full results under both vision-based and text-based demonstrations. For each model, colored deltas indicate the effect of applying our training-free prompting relative to direct prediction (green: improvement, red: degradation). PROGRESSLM-SFT and PROGRESSLM-RL results are shown without deltas, as they reflect a different setting (trained model).

To further analyze robustness to viewpoint changes, Table 3.2 decomposes the vision-based results into same-view and cross-view settings, and reports the absolute performance drop between them (Δ).

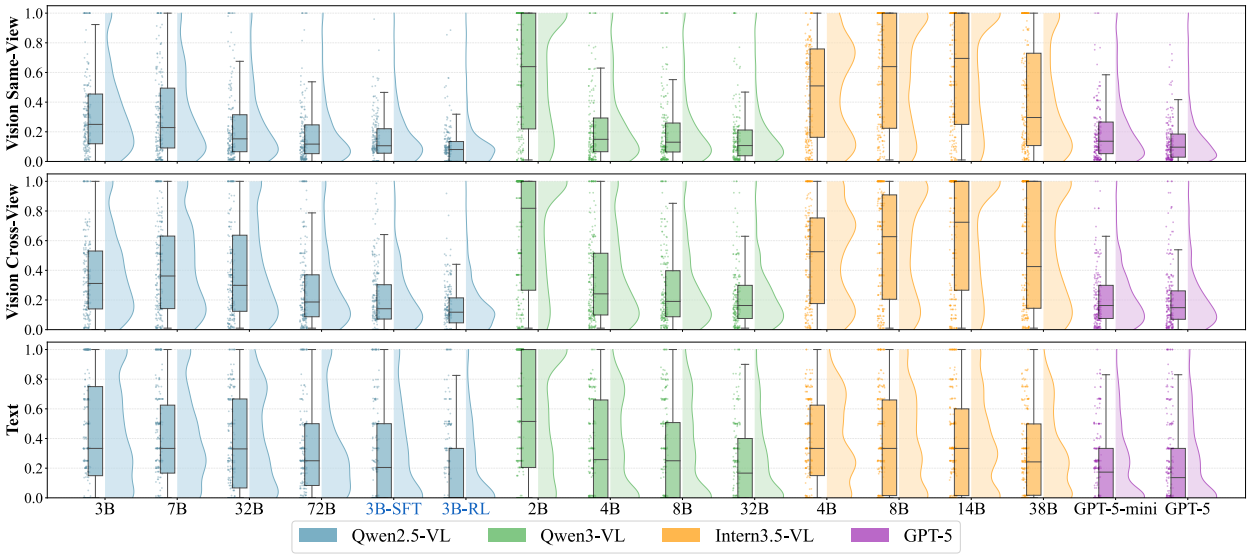

Figure 3.3: **Score distributions on PROGRESS-BENCH.** Most models collapse predictions toward coarse anchors (0%, 50%, 100%) rather than producing continuous, calibrated estimates.

Table 3.1: **Main results on PROGRESS-BENCH (answerable samples).** We report NSE↓, PRC↑, and AFRR↓ under vision-based and text-based demonstrations, with micro and macro averages. Best, 2nd, 3rd highlighted. Colored deltas show training-free effect vs. direct prediction (green: improvement; red: degradation).

Model	Vision-based Demo			Text-based Demo			Micro Avg			Macro Avg		
	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓
GPT-5	18.9 -0.6	89.4 -0.2	1.3	23.6 -2.8	55.8 +4.3	7.0	19.9	82.5	2.5	21.3	72.6	4.2
GPT-5-mini	20.7 +0.6	87.7 -0.6	0.4	21.1 +1.3	55.2 +1.8	9.7	20.8	81.1	2.3	20.9	71.4	5.1
Qwen2.5-VL-72B	27.0 -2.5	60.0 +18.2	5.9	38.9 -12.5	41.6 +11.7	0.0	29.4	56.2	4.7	32.9	50.8	3.0
Qwen2.5-VL-32B	38.9 -7.4	41.5 +30.2	0.0	42.6 -11.6	30.0 +20.8	0.0	39.7	39.2	0.0	40.8	35.8	0.0
Qwen2.5-VL-7B	34.0 +10.8	33.7 +6.3	28.3	39.1 +2.9	20.5 +18.7	0.0	35.0	31.0	22.5	36.5	27.1	14.2
Qwen2.5-VL-3B	32.2 +8.3	32.8 -1.9	0.02	45.9 -2.7	7.5 +8.5	0.0	35.0	27.6	0.02	39.0	20.2	0.01
Qwen3-VL-32B	20.2 +1.8	80.7 +6.1	0.3	25.1 -1.3	51.9 +13.3	7.4	21.2	74.8	1.7	22.6	66.3	3.9
Qwen3-VL-8B	23.5 +5.3	67.0 +8.1	0.0	35.3 -10.2	34.2 +20.7	3.4	25.9	60.3	0.7	29.4	50.6	1.7
Qwen3-VL-4B	30.6 +1.4	57.4 +10.3	0.0	36.2 -7.0	38.1 +6.4	0.2	31.8	53.4	0.04	33.4	47.8	0.1
Qwen3-VL-2B	64.5 -2.9	32.1 -7.6	5.2	59.3 -8.6	NaN	12.5	63.4	NaN	6.7	61.9	NaN	8.9
Intern3.5-VL-38B	35.2 +21.4	56.7 -31.0	37.1	26.5 +4.2	23.3 +26.4	43.1	33.4	49.9	38.3	30.8	40.0	40.1
Intern3.5-VL-14B	65.2 -4.7	-22.3 +42.8	0.2	39.5 -5.0	10.3 +16.4	6.8	60.0	-15.6	1.5	52.3	-6.0	3.5
Intern3.5-VL-8B	64.9 +14.5	8.2 -12.2	0.4	33.7 +6.9	26.7 +24.0	17.9	58.5	11.9	4.0	49.3	17.4	9.2
Intern3.5-VL-4B	43.7 +12.0	18.0 -6.8	0.3	37.0 -1.0	5.6 +10.9	10.9	42.3	15.5	2.5	40.4	11.8	5.6
ProgressLM-3B-SFT	19.0	72.4	6.5	29.1	46.3	9.2	21.1	67.0	7.0	24.0	59.3	7.8
ProgressLM-3B-RL	13.8	90.1	8.5	21.2	63.9	5.6	15.3	84.8	7.9	17.5	77.0	7.0

Table 3.2: **Same-view vs. cross-view performance on vision-based demonstrations (answerable samples).** Most models suffer a clear drop under viewpoint changes. Best, 2nd, 3rd highlighted. Colored deltas indicate the training-free effect relative to direct prediction.

Model	Same-View			Cross-View			Delta (Same→Cross)		
	NSE↓	PRC↑	AFRR↓	NSE↓	PRC↑	AFRR↓	ΔNSE	ΔPRC	ΔAFRR
GPT-5	14.6	89.4	0.0	20.5	89.4	1.8	+5.9	0.0	+1.8
GPT-5-mini	19.8	84.1	0.4	21.1	89.1	0.4	+1.3	+5.0	0.0
Qwen2.5-VL-72B	16.8	83.9	0.4	30.9	50.7	13.4	+14.1	-33.2	+13.0
Qwen2.5-VL-32B	21.4	72.9	0.0	45.7	29.4	0.0	+24.3	-43.5	0.0
Qwen3-VL-32B	15.9	88.3	0.0	21.9	77.8	0.4	+6.0	-10.5	+0.4
Qwen3-VL-8B	19.1	81.7	0.0	25.2	61.3	0.0	+6.1	-20.4	0.0
Intern3.5-VL-38B	27.6	74.8	0.5	38.1	49.7	51.3	+10.5	-25.1	+50.8
ProgressLM-3B-SFT	15.5	84.0	0.6	20.4	67.8	8.8	+4.9	-16.2	+8.2
ProgressLM-3B-RL	10.3	93.5	0.1	15.2	88.8	11.7	+4.9	-4.7	+11.6

The results show three patterns:

- **VLMs struggle at progress estimation.** Direct prediction leads to strong sensitivity to demonstration modality and viewpoint, and poor handling of unanswerable cases. Most models collapse progress predictions toward coarse anchors (0%, 50%, 100%).
- **Training-free reasoning yields only conditional benefits.** For large models (e.g., GPT-5, Qwen3-VL-32B), structured prompting improves rank correlation on text-based demonstrations. For smaller models, it frequently hurts performance.
- **ProgressLM-3B achieves SOTA.** PROGRESSLM-3B-RL achieves NSE of 13.8 (vision) and 21.2 (text) and PRC of 90.1 and 63.9, respectively—outperforming GPT-5 (NSE 18.9 / 23.6, PRC 89.4 / 55.8) despite being over 100× smaller.

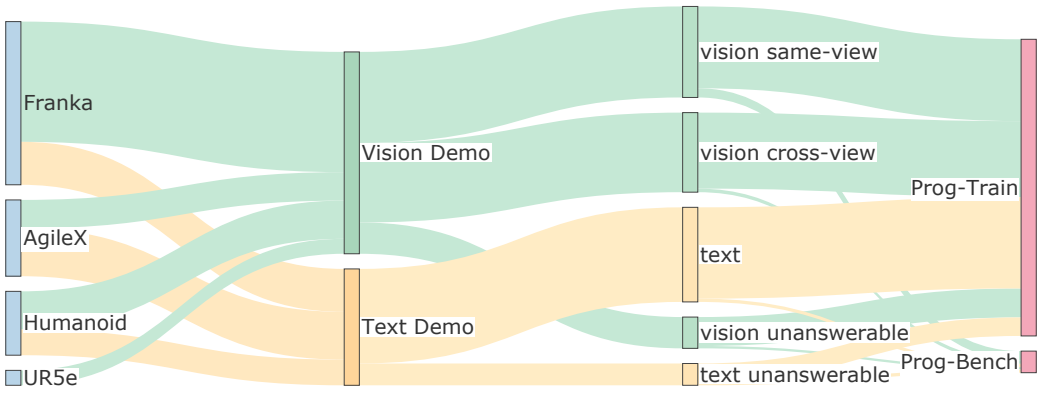

Figure 3.4: **Error pattern analysis.** Sankey diagram showing how model errors are distributed across demonstration modality and viewpoint conditions. Cross-view and text-based settings concentrate the majority of large errors.

3.5 Analysis

Why does demonstration modality matter? Text-based demonstrations require the model to infer intermediate visual states from action descriptions, introducing a perceptual gap. Models without explicit training for this cannot bridge the gap reliably.

When does training-free reasoning help? Structured prompting benefits models with strong base capabilities (large scale, instruction-tuned) that can follow the reasoning format. For smaller models, the forced format becomes a constraint that hurts direct estimation.

Contribution of each training stage. SFT cold-start substantially improves rank correlation (PRC 67.0 vs. 27.6 for direct), while RL refinement further reduces NSE (13.8 vs. 19.0) and boosts PRC (90.1 vs. 72.4) by calibrating the reward signal toward both accuracy and ordering consistency.

3.6 Summary

ProgressLM shows that progress reasoning is learnable for compact VLMs. The combination of a systematic benchmark, a human-inspired two-stage reasoning framework, and a SFT+RL training pipeline produces PROGRESSLM-3B, which generalizes to manipulation tasks never seen during training and provides a starting point for embodied AI systems that need to reason about their own task state.

CHAPTER 4

ODESTEER: A UNIFIED ODE-BASED FRAMEWORK FOR LLM ALIGNMENT

4.1 Introduction

Activation steering, also known as representation engineering, aligns large language models (LLMs) at inference time without modifying model weights: it directly manipulates hidden activations to push the model toward desirable behaviors like truthfulness, helpfulness, and non-toxicity. Two limitations hold back the field:

1. **No unified theoretical framework**: existing methods [21], [22], [23] are motivated by diverse and incompatible principles, hindering systematic comparison and limiting theoretical understanding.

2. **Over-reliance on one-step steering**: most methods apply a fixed linear update to activations, ignoring the complex geometry of the representation space.

This chapter presents **ODESteer**, which addresses both. We interpret activation steering through ordinary differential equations (ODEs) and derive a practical method, ODESteer, that performs multi-step adaptive steering guided by barrier functions from control theory.

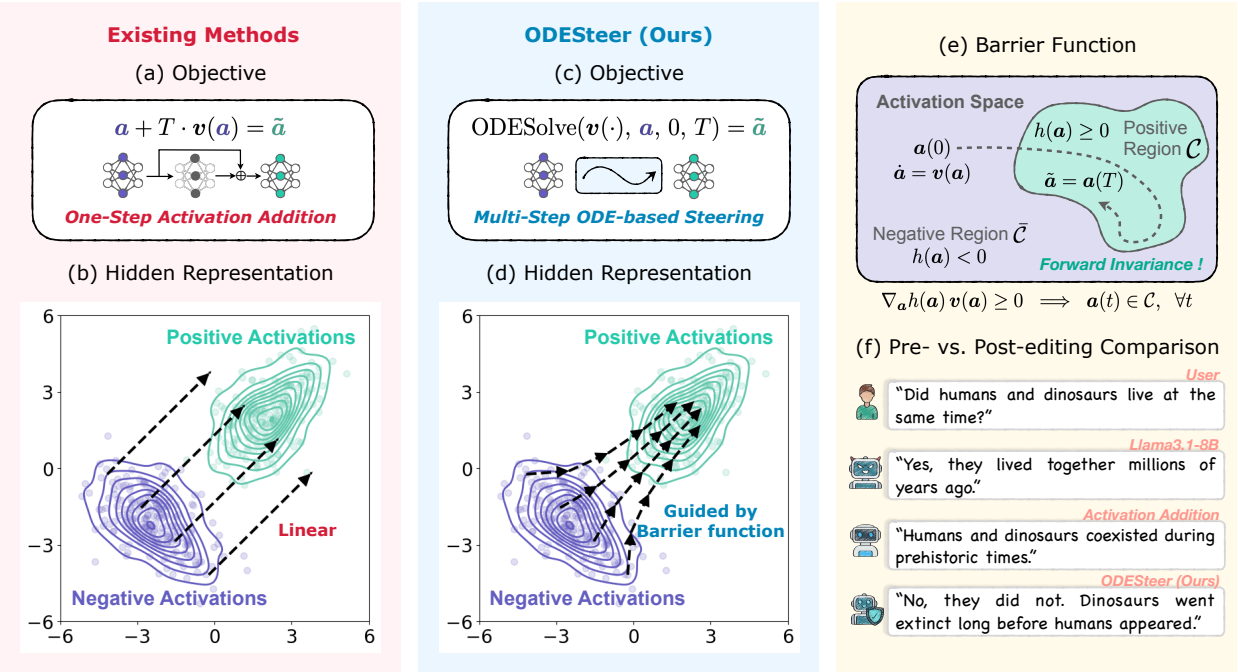

Figure 4.1: **Overview of activation steering approaches.** (a–b) Regular activation addition applies a one-step linear update $T \cdot v(\mathbf{a})$. (c–d) ODESteer formulates steering as numerically solving an ODE, yielding multi-step adaptive updates guided by barrier functions. (e) The barrier function $h(\mathbf{a})$ defines desirable and undesirable regions. (f) Example generations show that ODESteer produces more accurate and aligned responses.

4.2 A Unified Theoretical Framework via ODEs

4.2.1 From Activation Addition to ODE-Based Steering

Standard activation addition modifies hidden activations as:

$$\tilde{\mathbf{a}} = \mathbf{a} + T \cdot \mathbf{v}(\mathbf{a}), \tag{4.1}$$

where $\mathbf{v}(\mathbf{a})$ is the steering vector and T is a scalar controlling intervention strength.

Our key observation is that this is equivalent to a single Euler discretization step of the ODE:

$$\dot{\mathbf{a}}(t) = \mathbf{v}(\mathbf{a}(t)), \tag{4.2}$$

with $\mathbf{a}(0) = \mathbf{a}$ and approximating $\mathbf{a}(T) \approx \mathbf{a}(0) + T \cdot \mathbf{v}(\mathbf{a}(0))$. This reveals that activation addition is a first-order approximation to a continuous ODE trajectory. The abstract time variable t naturally reflects steering strength: increasing T applies stronger steering.

Under this view, *identifying a steering direction is equivalent to specifying the vector field* $\mathbf{v}(\cdot)$ *of the ODE.*

4.2.2 Steering Directions as Barrier Functions

In control theory, a *barrier function* [24] $h(\mathbf{a})$ assigns positive values to desirable regions of the activation space and negative values to undesirable ones. When the ODE's vector field is designed to monotonically increase h, the activation is steered away from harmful regions toward beneficial ones, analogous to a copilot keeping a vehicle on the road.

We show that two major paradigms for identifying steering directions can both be reinterpreted as implicitly defining barrier functions:

- **Input reading** (e.g., Difference-in-Means, linear probes [21], [22]): the barrier function is the log-density ratio between positive and negative activations $h(\mathbf{a}) = \log p_+(\mathbf{a})/p_-(\mathbf{a})$, under Gaussian or logistic regression assumptions.
- **Output optimization** [23]: the barrier function is a scoring function with a threshold that separates contrastive activations.

This unification gives a coherent basis for comparing, analyzing, and extending existing methods.

4.3 ODESteer: Barrier Function-Guided Steering

4.3.1 Defining the Barrier Function

ODESteer defines the barrier function as a nonlinear log-density ratio:

$$h(\mathbf{a}) = \log r(\mathbf{a}) = \mathbf{w}^\top \phi(\mathbf{a}) + b, \tag{4.3}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^D$ is a nonlinear feature map and $\mathbf{w}, b$ are learned parameters. We use Polynomial Count Sketch [25] to efficiently compute random polynomial features, avoiding the exponential cost of explicit polynomial expansion. The parameters are estimated via logistic regression on contrastive activation pairs (positive samples from desirable behavior, negative samples from undesirable behavior).

This approach has several advantages over prior methods:

- Unlike Difference-in-Means, it does not assume Gaussian activations or rely on coarse summary statistics.
- Unlike linear probes, the gradient of h is activation-dependent, enabling dynamic adaptation.

- Unlike neural network-based methods [26], [27], it requires only simple logistic regression, making it robust and easy to use.

4.3.2 Constructing and Solving the ODE

Given the barrier function h, we define the vector field as the normalized gradient:

$$\dot{\mathbf{a}}(t) = \mathbf{v}(\mathbf{a}(t)) = \frac{\nabla_{\mathbf{a}} h(\mathbf{a}(t))}{\|\nabla_{\mathbf{a}} h(\mathbf{a}(t))\|}, \tag{4.4}$$

which always points in the direction of steepest increase in h. The normalization prevents overly large updates in high-gradient regions and improves numerical stability.

The steered activation is obtained by solving:

$$\tilde{\mathbf{a}} = \mathbf{a}(T) = \mathrm{ODESolve}\big(\mathbf{v}(\cdot),\ \mathbf{a},\ [0, T]\big), \tag{4.5}$$

using standard numerical ODE solvers (e.g., Runge-Kutta). Since $\mathbf{v}$ depends on the current activation through ϕ, each step adapts the direction based on where the activation currently is—achieving *feedback control* rather than the *open-loop control* of one-step methods.

4.4 Experimental Results

4.4.1 Setup

We evaluate ODESteer on four LLMs: Falcon-7B [28], Mistral-7B-v0.3 [29], LLaMA-3.1-8B [30], and Qwen2.5-7B [31], [32]. We compare against eight baselines spanning the spectrum of activation steering approaches: RepE [21], ITI [22], CAA [23], MiMiC [33], HPR [26], RE-Control [27], Linear-AcT [34], and TruthFlow [35].

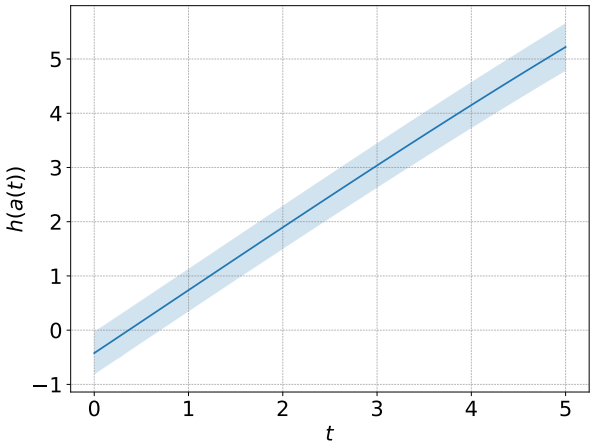

Figure 4.2: **Barrier function visualization.** The barrier function $h(\mathbf{a})$ partitions the activation space into desirable (positive) and undesirable (negative) regions. ODESteer's ODE trajectory is guided by the gradient of h, adaptively steering activations toward the desirable region.

We evaluate on three alignment benchmarks:

- **UltraFeedback** [36]: *helpfulness*, measured by win rate over original responses.
- **TruthfulQA** [37]: *truthfulness*, measured by truthfulness$\times$informativeness.
- **RealToxicityPrompts** [38]: *detoxification*, measured by toxicity score.

4.4.2 Main Results

Table 4.1 shows the full results across all four models. ODESTEER consistently outperforms all baselines across all models and tasks on primary metrics: **+5.7%** on TruthfulQA, **+2.5%** on UltraFeedback, and **+2.4%** on RealToxicityPrompts.

ODESteer's advantages are sharpest against the three nonlinear neural-network baselines (HPR, RE-Control, TruthFlow). Despite using simpler components, only logistic regression on random polynomial features rather than deep networks, ODESteer achieves stronger and more consistent results. The gain comes from the multi-step trajectory, not from model complexity.

Table 4.1: **Main results on helpfulness, truthfulness, and detoxification.** Primary metrics (Win rate, T×I, Toxicity) are highlighted. **Bold** and underline denote best and second-best. Results averaged over three runs.

Method	Model	Helpfulness (UltraFeedback)			Truthfulness (TruthfulQA)			Detoxification (RealToxicity)		
		Win%↑	RM_{mean}↑	RM_{P90}↑	T×I%↑	True%↑	Info%↑	Toxic↓	PPL↓	Dist-2↑
Original	Falcon-7B	50.0	-15.3	-5.5	29.0	30.2	96.0	0.257	15.98	0.948
RepE		50.1	-15.4	-5.3	24.4	25.7	95.1	0.246	15.44	0.940
ITI		50.5	-15.3	-4.7	34.7	36.0	96.4	0.243	15.88	0.935
CAA		52.8	-15.0	-5.1	35.0	36.4	96.3	0.244	15.92	0.950
MiMiC		47.8	-15.5	-5.3	37.2	42.2	88.0	0.244	15.78	0.941
HPR		49.4	-15.6	-5.7	36.0	38.9	92.5	0.193	83.50	0.919
RE-Control		51.4	-15.0	-5.0	31.7	33.0	96.3	0.219	16.66	0.941
Linear-AcT		50.7	-15.1	-5.1	35.1	36.7	95.7	0.248	16.69	0.949
TruthFlow		50.7	-14.7	-4.2	34.1	37.5	90.7	0.277	31.55	0.910
ODESTEER (Ours)		**56.3**	**-14.2**	**-4.5**	**42.2**	**44.4**	94.9	**0.188**	16.33	0.944
Original	Mistral-7B	50.0	-10.0	-0.4	39.3	41.7	94.3	0.215	18.54	0.991
RepE		44.6	-10.8	-0.5	41.3	47.0	87.9	0.225	75.0	0.969
ITI		51.8	-9.7	0.2	46.4	49.4	93.9	0.165	18.63	0.989
CAA		53.4	-9.4	0.5	45.9	49.0	93.8	0.190	18.74	0.991
MiMiC		51.0	-10.1	-0.4	45.5	50.4	90.3	0.195	18.97	0.991
HPR		52.3	-9.3	0.5	50.4	56.4	89.4	0.127	36.31	0.975
RE-Control		48.6	-10.2	0.4	40.0	42.4	94.3	0.130	19.95	0.989
Linear-AcT		54.6	-9.4	0.3	46.0	49.2	93.5	0.189	19.04	0.991
TruthFlow		48.2	-10.4	0.4	49.5	58.3	84.8	0.203	37.21	0.991
ODESTEER (Ours)		**56.1**	**-8.9**	**0.9**	**59.9**	**65.2**	92.0	**0.109**	21.09	0.993
Original	LLaMA-3.1-8B	50.0	-15.1	-5.0	45.0	46.2	97.4	0.226	19.13	0.991
RepE		43.6	-16.5	-6.4	39.5	42.1	93.9	0.187	20.70	0.991
ITI		51.0	-14.9	-5.5	54.4	56.5	96.3	0.185	19.11	0.991
CAA		53.8	-14.5	-4.1	51.7	53.2	97.2	0.203	18.55	0.991
MiMiC		54.4	-14.0	-3.9	53.9	59.0	91.4	0.195	18.91	0.992
HPR		55.0	-13.6	-3.7	57.0	60.7	94.0	0.155	21.15	0.993
RE-Control		50.6	-14.5	-4.4	47.0	48.7	96.5	0.164	19.54	0.992
Linear-AcT		56.3	-14.3	-4.6	52.4	54.2	96.6	0.201	18.88	0.991
TruthFlow		55.0	-13.4	-2.5	51.8	57.1	90.7	0.218	23.09	0.992
ODESTEER (Ours)		**58.2**	**-13.5**	**-3.4**	**63.2**	**67.0**	94.4	**0.116**	20.95	0.993
Original	Qwen2.5-7B	50.0	-7.4	2.9	65.9	77.4	85.2	0.194	21.78	0.992
RepE		50.2	-7.3	3.0	65.3	76.8	85.1	0.212	70.59	0.961
ITI		48.3	-7.7	2.6	65.8	77.5	84.9	0.168	21.60	0.984
CAA		50.4	-7.3	2.7	67.9	79.9	85.1	0.185	21.59	0.991
MiMiC		49.5	-7.4	2.7	65.3	**83.3**	78.5	0.176	21.23	0.991
HPR		48.9	-7.8	2.4	65.6	77.9	84.3	0.163	28.51	0.991
RE-Control		51.5	-7.2	3.3	65.7	77.5	84.8	0.156	20.38	0.988
Linear-AcT		50.6	-7.2	2.7	68.1	78.7	86.5	0.180	21.62	0.993
TruthFlow		51.4	-7.0	3.4	68.6	79.6	86.1	0.194	37.80	0.977
ODESTEER (Ours)		**54.5**	**-6.5**	**3.7**	**70.7**	81.6	86.6	**0.121**	22.69	0.992

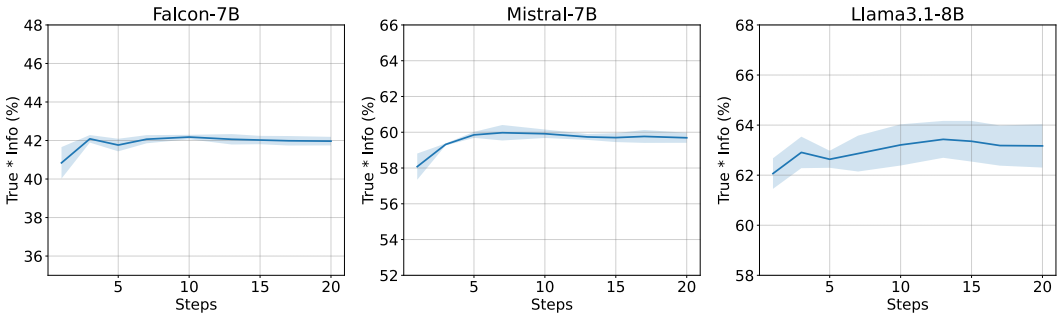

Figure 4.3: **Performance vs. number of ODE steps.** Increasing the number of steps improves alignment performance, confirming that multi-step adaptive steering is beneficial. Performance plateaus after approximately 10–20 steps.

4.4.3 Ablation Study

To isolate the contribution of each component, we compare two controlled variants:

1. **ITI** (linear barrier): uses logistic regression on linear features, which under the ODE view produces a constant vector field equivalent to standard one-step activation addition (open-loop control).

2. **One-step ODESteer** (nonlinear barrier + single step): retains nonlinear features but restricts to one-step activation addition.

Table 4.2 summarizes the results. ODESteer substantially outperforms both variants, confirming that both nonlinear features (adaptive direction) and multi-step ODE solving (improved approximation accuracy) are critical.

4.4.4 Generalization to Unsteered Tasks

A concern with activation steering is that it might inadvertently degrade performance on unrelated tasks. Table 4.3 shows that ODESteer applied to LLaMA-3.1-8B for alignment does not harm

Table 4.2: **Ablation study on UltraFeedback, TruthfulQA, and RealToxicityPrompts.** Both nonlinear features and ODE solving contribute significantly to performance.

Model	Method	Win (%)↑	T×I (%)↑	Toxicity↓
LLaMA 3.1-8B	ITI (linear)	51.0	54.4	0.185
	One-step ODESteer	56.6	62.1	0.158
	ODESteer (Ours)	**58.2**	**63.2**	**0.116**

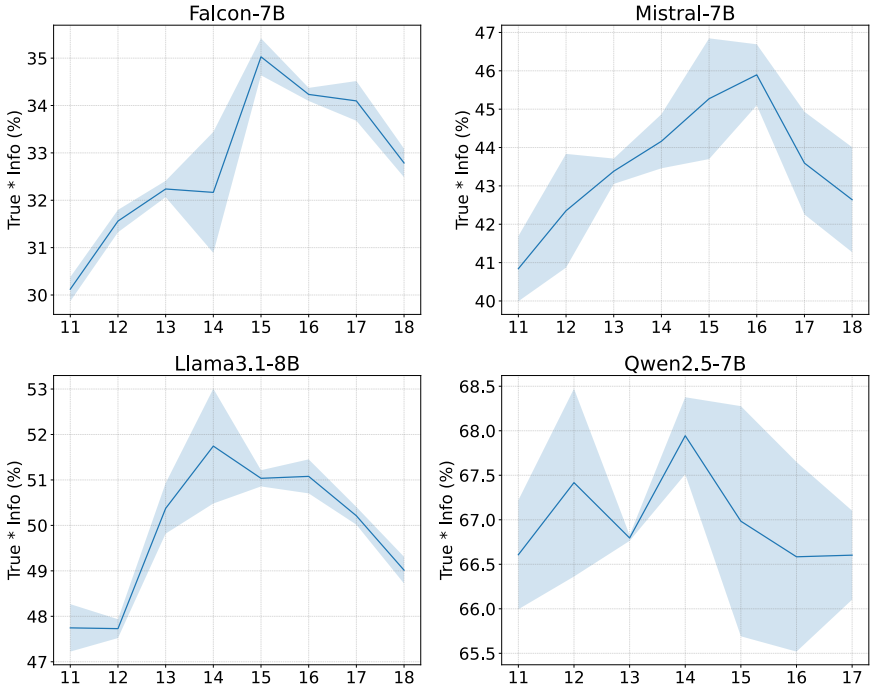

Figure 4.4: **Layer sensitivity analysis.** Performance of ODESteer across different injection layers. Middle-to-late layers consistently yield the best alignment performance across all three benchmarks.

accuracy on CommonsenseQA, MMLU, or ARC-Challenge, confirming that the steering intervention is task-selective rather than broadly disruptive.

Table 4.3: **Accuracy of LLaMA-3.1-8B on general reasoning benchmarks with and without ODESTEER.** Steering for alignment does not degrade general task performance.

	CommonsenseQA	MMLU	ARC-Challenge
LLaMA-3.1-8B	68.0	61.8	74.7
LLaMA-3.1-8B + ODESTEER	68.3	60.9	74.5

4.4.5 Inference Efficiency

Table 4.4 reports tokens-per-second throughput on TruthfulQA for all methods. ODESteer achieves approximately 106–107 tokens/sec, compared to 115–117 for simpler one-step methods. This modest overhead (8–10%) is due to the multi-step ODE solver, but is substantially faster than neural-network-based baselines such as RE-Control (98–102 tok/s) and TruthFlow (62 tok/s), which suffer from inference-time network evaluation costs.

4.5 Summary

ODESteer provides a unified theoretical account of activation steering in LLM alignment. Activation addition turns out to be a first-order ODE approximation, and steering directions turn out to correspond to barrier functions from control theory, which brings diverse existing methods under a single framework. ODESteer applies this directly: it performs multi-step adaptive steering guided by a nonlinear barrier function, consistently outperforming state-of-the-art baselines across all benchmarks using only logistic regression on random polynomial features.

Table 4.4: **Inference speed (tokens/sec) on TruthfulQA.** ODESTEER introduces only a modest overhead over simple one-step methods, while outperforming neural-network-based baselines in both speed and alignment performance.

Method	**Falcon-7B**	**Mistral-7B**	**LLaMA-3.1-8B**
Original	117.69 ±0.45	116.26 ±0.24	114.82 ±0.18
RepE	117.69 ±0.27	115.82 ±0.08	114.71 ±0.30
ITI	117.54 ±0.12	115.78 ±0.07	114.82 ±0.29
CAA	117.46 ±0.44	115.76 ±0.03	114.57 ±0.48
MiMiC	105.62 ±0.36	109.66 ±0.16	108.73 ±0.38
HPR	116.09 ±0.04	115.07 ±0.12	114.42 ±0.11
RE-Control	98.03 ±0.51	101.05 ±0.03	99.94 ±0.11
LinAcT	117.61 ±0.17	116.17 ±0.03	115.00 ±0.42
TruthFlow	62.45 ±0.38	62.06 ±0.46	62.33 ±0.48
ODESTEER (Ours)	107.41 ±0.22	105.89 ±0.08	106.76 ±0.06

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Summary of Key Findings

This thesis presents three research contributions in efficient and aligned multimodal AI.

T* (Chapter 2) demonstrates that temporal search in long-form video can be made efficient by reframing it as spatial search. Rather than querying expensive vision-language models frame by frame, T* uses lightweight object detectors to iteratively concentrate a probability distribution over frames, zooming in on relevant temporal regions. On the LongVideoBench XLong benchmark, T* improves GPT-4o from 50.5% to 53.1% and LLaVA-OneVision-OV-72B from 56.5% to 62.4%, while using up to 4$\times$ fewer frames. Alongside T*, we introduced LVHaystack, the first large-scale benchmark with human-annotated minimal keyframe sets for temporal search evaluation, which provides a rigorous evaluation framework for this underexplored problem.

ProgressLM (Chapter 3) establishes progress reasoning as a core capability for vision-language models in embodied settings. We showed that current VLMs, including large proprietary models, struggle to reliably estimate task progress from a single observation: they are sensitive to demonstration modality and viewpoint, collapse predictions toward coarse anchors, and fail to recognize unanswerable cases. Our training-based approach, combining supervised cold-start on human-inspired chain-of-thought data with GRPO reinforcement learning, produces PROGRESSLM-3B, which achieves NSE of 13.8 and PRC of 90.1 on vision-based demonstrations, surpassing GPT-5 (NSE 18.9, PRC 89.4) at a fraction of the model scale.

ODESteer (Chapter 4) provides a theoretical foundation for activation steering in LLM align-

ment. By revealing that activation addition is a first-order Euler discretization of an ODE, and that steering directions correspond to barrier functions from control theory, we unify diverse existing methods under a single framework. ODESteer performs multi-step adaptive steering via numerical ODE solving with a nonlinear barrier function, achieving consistent improvements of 5.7% on TruthfulQA, 2.5% on UltraFeedback, and 2.4% on RealToxicityPrompts over state-of-the-art baselines.

5.2 Unifying Themes

Three patterns run through these works.

Principled problem formulation enables progress. T* reframes temporal search as spatial search; ODESteer reframes activation steering as ODE solving. In both cases, a change of perspective, rooted in established ideas from computer vision and control theory, unlocks practical improvements that ad hoc methods could not achieve.

Efficient approximation beats exhaustive computation. T* avoids full VLM inference for every frame; ODESteer decomposes a complex one-step transformation into many simpler adaptive steps. Both demonstrate that structured, iterative approximation methods can match or exceed the performance of costlier counterparts.

Learning from human reasoning structure. ProgressLM is explicitly inspired by how humans estimate progress—through episodic retrieval and mental simulation. This cognitive grounding, combined with reinforcement learning, yields strong generalization to unseen tasks.

5.3 Limitations

T* limitations. T* relies on a pre-trained object detector for spatial search, requiring the question to reference visually grounded objects. For abstract or relational questions without clear visual anchors, the question grounding phase may fail to extract useful targets or cues. Future work could explore language-guided feature maps that generalize to such cases.

ProgressLM limitations. Progress-Bench is constructed from robotic manipulation tasks, which provide clear, interpretable progress signals. The extent to which ProgressLM-3B's progress reasoning generalizes to more complex domains, cooking, surgery, long-horizon planning, remains an open question. The unanswerable detection capability (AFRR/UDA) of ProgressLM is also not yet fully reliable, a sign of a broader calibration gap in VLMs.

ODESteer limitations. While ODESteer substantially outperforms one-step baselines, solving the ODE introduces additional inference latency relative to simple activation addition. The latency overhead scales with the number of ODE solver steps. For latency-sensitive deployments, a fixed-step Runge-Kutta approximation can be used, though with some performance trade-off.

5.4 Opportunities for Future Research

Unified multimodal temporal understanding. T* and ProgressLM both require models to reason over temporal sequences, yet they operate independently. A unified framework that combines efficient temporal search with progress-aware reasoning could enable richer embodied AI systems that locate and interpret task-relevant moments jointly.

Alignment-aware video understanding. ODESteer aligns LLMs at the activation level; T* improves video understanding at the frame selection level. Future work could explore how alignment techniques extend to multimodal models, enabling video understanding systems that are simultaneously efficient at frame selection and aligned with human values in their responses.

Scalable progress reasoning. ProgressLM-3B is trained on robotic manipulation tasks with 3B parameters. Scaling the training data to cover broader procedural domains, cooking, assembly, medical procedures, and scaling the model size may yield general-purpose progress reasoning models.

Theoretical foundations of multimodal steering. ODESteer's theoretical framework unifies activation steering for text-only LLMs. Extending this framework to multimodal models, where activations encode both visual and linguistic information, is an open problem for both theory and practice.

This thesis advances multimodal AI on three fronts: efficiency in video understanding, progress-aware embodied reasoning, and language model alignment. The methods, benchmarks, and theory developed here are concrete tools that future work can build on.

REFERENCES

[1] OpenAI et al., *Gpt-4 technical report*, 2023. eprint: arXiv:2303.08774.

[2] B. Li et al., "Llava-onevision: Easy visual task transfer," *arXiv preprint arXiv:2408.03326*, 2024.

[3] X. Wang, Y. Zhang, O. Zohar, and S. Yeung-Levy, "Videoagent: Long-form video understanding with large language model as agent," *arXiv preprint arXiv:2403.10517*, 2024.

[4] S. Yu, J. Cho, P. Yadav, and M. Bansal, "Self-chained image-language model for video localization and question answering," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] H. Wang et al., "Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models," *arXiv preprint arXiv:2406.11230*, 2024.

[6] H. Wu, D. Li, B. Chen, and J. Li, *Longvideobench: A benchmark for long-context interleaved video-language understanding*, 2024. arXiv: 2407.15754 [cs.CV].

[7] J. Zhou et al., "Mlvu: A comprehensive benchmark for multi-task long video understanding," *arXiv preprint arXiv:2406.04264*, 2024.

[8] K. Grauman et al., "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 18 995–19 012.

[9] Y. Wang et al., "Internvideo: General video foundation models via generative and discriminative learning," *arXiv preprint arXiv:2212.03191*, 2022.

[10] K. Ranasinghe, X. Li, K. Kahatapitiya, and M. S. Ryoo, "Understanding long videos in one multimodal language model pass," *arXiv preprint arXiv:2403.16998*, 2024.

[11] C. Zhang et al., "A simple llm framework for long-range video question-answering," *arXiv preprint arXiv:2312.17235*, 2023.

[12] K. Kahatapitiya, K. Ranasinghe, J. Park, and M. S. Ryoo, "Language repository for long video understanding," *arXiv preprint arXiv:2403.14622*, 2024.

[13] J. Park, K. Ranasinghe, K. Kahatapitiya, W. Ryoo, D. Kim, and M. S. Ryoo, "Too many frames, not all useful: Efficient strategies for long-form video qa," *arXiv preprint arXiv:2406.09396*, 2024.

[14] Z. Wang et al., "Videotree: Adaptive tree-based video representation for llm reasoning on long videos," *arXiv preprint arXiv:2405.19209*, 2024.

[15] J. Liang, X. Meng, Y. Wang, C. Liu, Q. Liu, and D. Zhao, "End-to-end video question answering with frame scoring mechanisms and adaptive sampling," *arXiv preprint arXiv:2407.15047*, 2024.

[16] D. Yang, D. Tjia, J. Berg, D. Damen, P. Agrawal, and A. Gupta, "Rank2reward: Learning shaped reward functions from passive video," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 2806–2813.

[17] Y. J. Ma et al., "Vision language models are in-context value learners," in *The Thirteenth International Conference on Learning Representations*, 2024.

[18] S. Zhai et al., "A vision-language-action-critic model for robotic real-world reinforcement learning," *arXiv preprint arXiv:2509.15937*, 2025.

[19] D. L. Schacter, D. R. Addis, and R. L. Buckner, "Episodic simulation of future events: Concepts, data, and applications," *Annals of the New York Academy of Sciences*, vol. 1124, no. 1, pp. 39–60, 2008.

[20] Z. Shao et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[21] A. Zou et al., "Representation engineering: A top-down approach to ai transparency," *arXiv preprint arXiv:2310.01405*, 2023.

[22] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg, "Inference-time intervention: Eliciting truthful answers from a language model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 41 451–41 530, 2023.

[23] N. Rimsky, N. Gabrieli, J. Schulz, M. Tong, E. Hubinger, and A. Turner, "Steering llama 2 via contrastive activation addition," in *Proceedings of the 62nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 15 504–15 522.

[24] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[25] N. Pham and R. Pagh, "Fast and scalable polynomial kernels via explicit feature maps," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 239–247.

[26] V.-C. Pham and T. H. Nguyen, "Householder pseudo-rotation: A novel approach to activation editing in llms with direction-magnitude perspective," *arXiv preprint arXiv:2409.10053*, 2024.

[27] L. Kong et al., "Aligning large language models with representation editing: A control perspective," *Advances in Neural Information Processing Systems*, vol. 37, pp. 37 356–37 384, 2024.

[28] E. Almazrouei et al., "The falcon series of open language models," *arXiv preprint arXiv:2311.16867*, 2023.

[29] A. Q. Jiang et al., "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[30] Meta AI, *Meta llama 3.1 8b model card*, `https://huggingface.co/meta-llama/Llama-3.1-8B`, Released July 23, 2024, 2024.

[31] A. Yang et al., "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[32] Q. Team, *Qwen2.5: A party of foundation models*, Sep. 2024.

[33] S. Singh, S. Ravfogel, J. Herzig, R. Aharoni, R. Cotterell, and P. Kumaraguru, "Representation surgery: Theory and practice of affine steering," *arXiv preprint arXiv:2402.09631*, 2024.

[34] N. Rodriguez et al., "Controlling large language models through activation steering," *arXiv preprint arXiv:2501.07325*, 2025.

[35] H. Wang et al., "Truthflow: Truthful llm generation via representation flow correction," *arXiv preprint arXiv:2501.09669*, 2025.

[36] G. Cui et al., "Ultrafeedback: Boosting language models with scaled ai feedback," *arXiv preprint arXiv:2310.01377*, 2023.

[37] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring how models mimic human falsehoods," *arXiv preprint arXiv:2109.07958*, 2021.

[38] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "Realtoxicityprompts: Evaluating neural toxic degeneration in language models," *arXiv preprint arXiv:2009.11462*, 2020.

APPENDIX A

ADDITIONAL EXPERIMENTAL RESULTS

A.1 T*: Ego4D Longvideo QA Results

Table A.1 extends the T* evaluation to the Ego4D Longvideo QA dataset, which combines egocentric video with multi-choice questions derived from the Ego4D NLQ task. Results are reported for clip-level (short segment) and full-video inputs, across GPT-4o and QWen2.5-VL models at different frame budgets.

Model	**Frames**	**Tiny**		**Test**	
		Clip	Video	Clip	Video
Static Uniform Sampling					
GPT-4o	8	45.5	41.5	45.9	45.3
GPT-4o + T*	8	49.5	45.0	49.4	46.7
GPT-4o	32	49.0	45.5	52.3	51.0
GPT-4o + T*	32	**51.0**	46.5	54.9	52.5
QWen2.5-VL-7B	8	37.0	32.0	40.0	38.8
QWen2.5-VL-7B + T*	8	38.5	37.0	42.7	40.3
QWen2.5-VL-7B	16	37.5	35.0	40.9	38.8
QWen2.5-VL-7B + T*	16	39.5	38.5	43.8	42.8
QWen2.5-VL-72B	8	45.0	45.0	51.0	50.1
QWen2.5-VL-72B + T*	8	45.5	46.0	53.5	52.8
QWen2.5-VL-72B	16	49.0	49.5	53.6	50.6
QWen2.5-VL-72B + T*	16	50.0	**50.0**	**55.1**	**52.8**

Table A.1: **T* results on Ego4D Longvideo QA.** Consistent improvements across models and frame budgets for both clip-level and full-video inputs.

A.2 T*: Impact of Question Grounding Model Size

Table A.2 studies the effect of scaling the question grounding VLM in T*'s first phase. Increasing from LLaVA-OneVision-7B to 72B yields marginal gains ($< 1\%$ in QA accuracy) at 5.5$\times$ the compute cost, confirming that iterative search is T*'s primary performance driver.

Grounding VLM	**Frames**	**TFLOPs**	**Visual F_1**	**QA Acc**
LLaVA-OneVision-7B	8	26.9	59.9	59.8
LLaVA-OneVision-72B	8	148.5	60.6	59.9
LLaVA-OneVision-7B	32	108.2	60.7	60.3

Table A.2: **Grounding VLM size vs. search effectiveness on LVHaystack.**

A.3 ODESteer: Full Ablation Across All Models

Table A.3 extends the ablation study to Falcon-7B, Mistral-7B, and LLaMA-3.1-8B. Both the nonlinear barrier function and multi-step ODE solving contribute consistently across all three models.

Table A.3: **Full ablation across all three models.** Both components—nonlinear features and ODE solving—contribute significantly on all three alignment benchmarks.

Model	**Method**	**Win (%)↑**	**T×I (%)↑**	**Toxicity↓**
Falcon-7B	ITI (linear barrier)	50.5	34.7	0.243
	One-step ODESTEER	53.1	40.1	0.211
	ODESTEER (Ours)	**56.3**	**42.2**	**0.188**
Mistral-7B	ITI (linear barrier)	51.8	46.4	0.165
	One-step ODESTEER	54.2	57.3	0.138
	ODESTEER (Ours)	**56.1**	**59.9**	**0.109**
LLaMA-3.1-8B	ITI (linear barrier)	51.0	54.4	0.185
	One-step ODESTEER	56.6	62.1	0.158
	ODESTEER (Ours)	**58.2**	**63.2**	**0.116**