Which Way Did It Move? Diagnosing and Overcoming Directional Motion Blindness in Video-LLMs

Jongseo Lee [1] **Hyuntak Lee** [1] **Sunghun Kim** [1] **Sooa Kim** [1] **Jihoon Chung** [2] **Jinwoo Choi** [1]

Abstract

Video Large Language Models (Video-LLMs) have made rapid progress on temporal video understanding, yet many fail at a basic perceptual primitive: signed image-plane motion direction. On simple videos of a single object moving left, right, up, or down, most Video-LLMs perform near chance, with above-chance cases largely attributable to prediction biases rather than genuine direction understanding. We call this failure *directional motion blindness*. We localize the failure by tracing motion direction information through the Video-LLM pipeline. Motion direction remains linearly accessible from the vision encoder, projector, and LLM hidden states, but the readout fails to bind this signal to the correct verbal answer option, revealing a *direction binding gap*. Although synthetic motion direction instruction tuning reduces this gap on the source domain, motion direction concept vector analysis shows that visual complexity weakens the signal magnitude and limits out-of-domain generalization. We introduce MoDirect, a dataset family for motion direction instruction tuning and evaluation, and `DeltaDirect`, a diagnosis-driven, projector-level objective that predicts normalized 2-D motion vectors from adjacent-frame feature deltas. On MoDirect-SynBench, instruction tuning with `DeltaDirect` improves motion direction accuracy from 25.9% to 85.4%. On MoDirect-RealBench, `DeltaDirect` improves real-world motion direction accuracy by 21.9 points over the vanilla baseline without real-world tuning data, while preserving standard video-understanding performance.

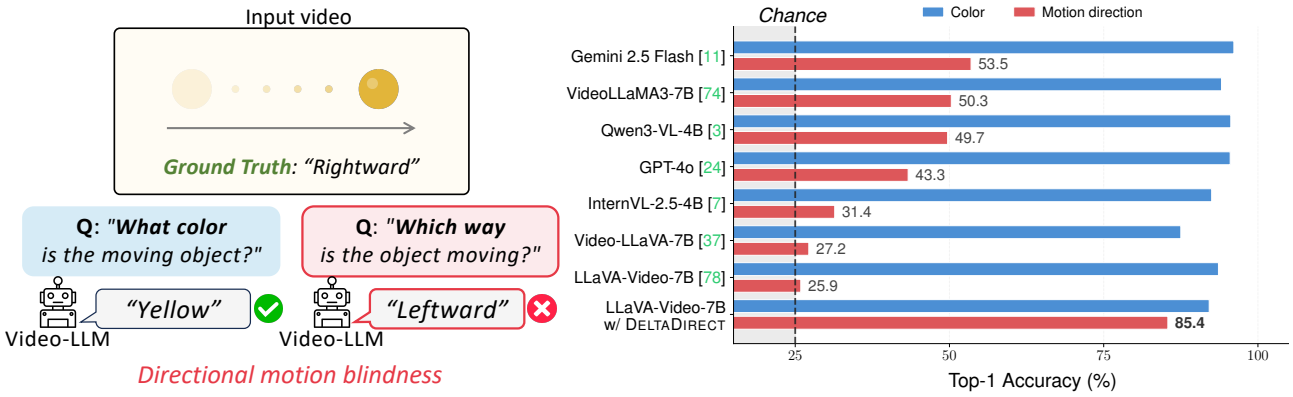
[1]Kyung Hee University, Yongin, South Korea [2]Princeton University, Princeton, NJ, USA. Correspondence to: Jinwoo Choi <jinwoochoi@khu.ac.kr>.

Accepted to the 2nd Workshop on Compositional Learning at ICML 2026, Seoul, South Korea. Copyright 2026 by the author(s).

1. Introduction

Recent work on Video-LLMs has made rapid progress in temporal understanding (Shangguan et al., 2025; Liu et al., 2024b; Hong et al., 2025; Li et al., 2025b; Qi et al., 2025) with different benchmarks analyzing a model's temporal capability by testing its memory (Patraucean et al., 2023), ability to identify complex human action or object movements (Salehi et al., 2024; Liu et al., 2024b; Shangguan et al., 2025), identifying the order of stitched videos (Liu et al., 2024b; Zhao et al., 2025b), or understanding long-form videos (Zhou et al., 2025a; Fu et al., 2025a; Mangalam et al., 2023). Despite this progress, we find that many Video-LLMs (Cheng et al., 2024b; Li et al., 2024c; 2025a; Zhang et al., 2024b; Lin et al., 2024; Wang et al., 2024; Bae et al., 2025; Li et al., 2024a) fail at a much simpler temporal primitive: motion direction understanding. Consider a seemingly trivial synthetic video in which a yellow circle moves from left to right, as shown in Figure 1. Can a Video-LLM correctly tell which way it moves: left→right, right→left, bottom→top, or top→bottom? To our surprise, most Video-LLMs show accuracy close to random chance (25%), with Gemini2.5-Flash (Comanici et al., 2025) only reaching 53.5% accuracy. Humans identify such motion directions instantly and effortlessly, reflecting the fundamental role of motion direction in visual perception and physical navigation (Born and Bradley, 2005; Gibson, 2014; Nakayama, 1985). We refer to this systematic failure to distinguish signed motion directions as *directional motion blindness*.

To understand where this failure arises, we trace motion direction information across the Video-LLM pipeline. We find that the signal remains linearly accessible throughout the pipeline: motion direction is decodable from the visual encoder, the projector output, and intermediate LLM states. The breakdown emerges at the readout position, i.e., the final token representation used to predict the answer: although a motion direction signal is still linearly accessible, the model does not reliably connect it to the correct verbal response (Orgad et al., 2025; Marks and Tegmark, 2024). We call this mismatch a *direction binding gap*, and observe the same pattern across multiple Video-LLMs (Figure 3). This suggests that directional motion blindness is not simply a failure of visual perception, but a failure to make an available motion signal usable for language-level readout.



(a) (b)

Figure 1. **Directional motion blindness in Video-LLMs.** (a) Given a simple synthetic video of a yellow circle moving from left to right, recent Video-LLMs correctly identify the object's color but answer the wrong motion direction. (b) Across Video-LLMs, appearance recognition is high, yet signed motion direction accuracy remains much lower, often near chance.

We therefore introduce MoDirect, a dataset family for signed image-plane motion direction, with three subsets: MoDirect-Inst, MoDirect-SynBench, and MoDirect-RealBench. Its synthetic part contains four domains under a controlled 2×2 design over foreground type and background type: Primitive-on-Syn, Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real. Primitive videos contain rendered geometric primitives such as circles, triangles, and squares; Cutout videos contain segmented real-world object images (Lin et al., 2014). Synthetic videos use uniform-color backgrounds, whereas Real videos use natural scene backgrounds (Zhou et al., 2017). From the simplest synthetic domain, Primitive-on-Syn, we construct MoDirect-Inst for instruction tuning. We use all four synthetic domains as MoDirect-SynBench for controlled synthetic evaluation, and use MoDirect-RealBench for real-video evaluation.

Instruction tuning on MoDirect-Inst substantially improves on the source domain motion direction recognition. However, models that perform well on Primitive-on-Syn still degrade on more complex domains, especially Cutout-on-Real. This indicates that next-token prediction on motion direction instructions can improve answer binding on the source domain, Primitive-on-Syn, but does not necessarily produce a strong domain-invariant motion direction signal.

We next ask why this generalization gap persists. Using difference-in-means (Arditi et al., 2024; Marks and Tegmark, 2024; Li et al., 2023; Rimsky et al., 2024; Tigges et al., 2024), we extract motion direction concept vectors from the readout position of each LLM layer for each domain and motion direction. After instruction tuning, motion direction concept vectors become well aligned across domains, despite being trained only on Primitive-on-Syn; for example, at certain layers, their cross-domain cosine similarity is much higher than in the baseline without motion direction instruction tuning. Yet their magnitude drops sharply on complex domains, mirroring the observed accuracy drop. Moreover, restoring only the magnitude of the motion direction concept vector recovers much of the lost accuracy. These results suggest that the out-of-domain failure is not due to a missing motion direction geometry, but to a *magnitude deficit*: the model learns the motion direction structure, but the signal is too weak to be reliably read out across domains. Please refer to Section 3 for more details.

This diagnosis motivates `DeltaDirect`, a training-only projector-level objective that strengthens signed displacement cues at the visual-language interface. Rather than adding motion tokens or a motion-specific encoder at inference time, `DeltaDirect` uses analytically available 2-D motion vectors from MoDirect-Inst as supervision during training. Specifically, `DeltaDirect` predicts normalized 2-D motion vectors from adjacent-frame projector-feature deltas, encouraging the projector output to carry a stronger signed displacement signal before it enters the LLM. The auxiliary branch is discarded after training, leaving the test-time input format, token sequence, model architecture, and decoding procedure unchanged.

We evaluate `DeltaDirect` across synthetic, real-world, and general video understanding benchmarks. On MoDirect-SynBench, instruction tuning on MoDirect-Inst improves over the zero-shot model, and adding `DeltaDirect` further improves average motion direction accuracy by 6.5 points, with the largest gain of 11.2 points on Cutout-on-Real. To test real-world transfer, we construct MoDirect-RealBench, a motion direction benchmark curated from Something-Something-V2 (SSv2) (Goyal et al., 2017), TOMATO (Shangguan et al., 2025), and KTH (Schuldt et al., 2004). Without seeing real-world videos during tuning, `DeltaDirect` improves accuracy by 21.9 points over the vanilla baseline. Beyond motion direction-specific benchmarks, `DeltaDirect` preserves or improves performance on standard video understanding benchmarks (Li et al., 2024a; Liu et al., 2024b; Patraucean et al., 2023; Hong et al., 2025; Tu

et al., 2025; Mangalam et al., 2023; Zhang et al., 2024a), suggesting positive transfer rather than overfitting to synthetic motion. Finally, when applied during full fine-tuning, `DeltaDirect` alone yields strong improvements on both motion direction and general video benchmarks, indicating that motion-vector supervision can serve as a useful training signal for future Video-LLMs.

In this work, our major contributions are threefold:

- We identify *directional motion blindness* in Video-LLMs: despite strong appearance recognition, recent models fail to distinguish basic signed motion directions such as left, right, up, and down. We diagnose this failure as a *direction binding gap*, where motion direction information is linearly decodable but not reliably bound to the correct verbal response.

- We show that instruction tuning closes the direction binding gap on the source domain but leaves an out-of-domain generalization gap. Through concept vector analysis, we trace this gap to a magnitude deficit in shared motion direction representations across domains.

- We introduce MoDirect, a controlled dataset family for motion direction instruction tuning and evaluation, and propose `DeltaDirect`, a training-only auxiliary objective that predicts motion vectors from adjacent-frame feature deltas. `DeltaDirect` improves motion direction understanding while preserving or improving general video understanding.

2. Related Work

Video-LLMs and temporal-motion benchmarks. Video-LLMs extend image-based MLLMs to temporal understanding through video instruction tuning, unified image-video architectures, and long-video modeling (Maaz et al., 2024; Lin et al., 2024; Li et al., 2024c; Wang et al., 2024; Liu et al., 2024a; Bae et al., 2025; Cheng et al., 2024b; Zhang et al., 2025a; Chung et al., 2025; Li et al., 2025a; Bai et al., 2025b;a; Liu et al., 2025b; Hurst et al., 2024; Comanici et al., 2025). Existing benchmarks evaluate temporal memory, action and event understanding, temporal ordering, and long-form video reasoning (Li et al., 2024a; Liu et al., 2024b; Shangguan et al., 2025; Cores et al., 2024; Mangalam et al., 2023). Motion direction appears in several temporal or motion-centric benchmarks, often alongside speed, order, rotation, and broader motion reasoning (Li et al., 2024b; Zhang et al., 2024a; Hong et al., 2025; Tu et al., 2025; Zhou et al., 2025b; Xue et al., 2025; Du et al., 2026). These benchmarks are valuable for real-world evaluation, but real videos entangle apparent direction with camera motion, occlusion, object identity, scene layout, scale change, and event semantics. We instead separate diagnosis from transfer: controlled synthetic videos *isolate signed image-plane direction*, and real-world benchmarks test whether the intervention transfers.

Mechanistic analysis of LLMs. A growing line of work analyzes internal representations rather than final outputs alone. Probing and difference-in-means analyses show that model states can encode information that is not reliably verbalized (Orgad et al., 2025; Marks and Tegmark, 2024). For Video-LLMs, recent work begins to trace layer-wise information flow and temporal bottlenecks inside the model (Kim et al., 2026; Li et al., 2025b). Our analysis is complementary: we isolate motion direction as a single perceptual primitive and ask *whether the signal is present, where it weakens, and how it couples to verbal readout*.

Motion representations and supervision. Video models encode motion explicitly through frame differences, optical flow, or trajectories for action recognition (Wang et al., 2021; Li et al., 2020; Kwon et al., 2020), flow distillation (Crasto et al., 2019; Zhu et al., 2018; Stroud et al., 2020), and self-supervised learning (Yang et al., 2022; Huang et al., 2021). Recent Video-LLMs use motion-vector tokenization (Jin et al., 2024), compressed-video cues (Zhao et al., 2025a), optical flow (Liu et al., 2025a; Nguyen et al., 2025), or motion-appearance disentanglement (Zhan et al., 2025). `DeltaDirect` instead uses motion vectors only as training-time supervision on projector-feature deltas, without changing the test-time input or decoding pipeline.

3. Where Does Directional Motion Blindness Arise in Video-LLMs?

We investigate a Video-LLM, LLaVA-Video (Zhang et al., 2025c), from visual representations to the answer token, and test whether the failure is due to missing supervision, insufficient prompting, visual encoding, projection, or the final binding between a perceived direction and the corresponding answer option. We also investigate several Video-LLMs and draw the same conclusions. For more results and details, refer to Appendix E.

We organize the analysis around three research questions. **(i)** Can the failure be explained by missing direction supervision or insufficient input-side scaffolding? (Section 3.2) **(ii)** Is the signed direction erased by the vision encoder or the vision-language projector, or the intermediate LLM layers? (Section 3.3) **(iii)** If direction remains accessible, can the final readout token bind it to the prompt-specific answer option? (Section 3.4) We find that motion direction remains linearly accessible across the visual encoder, projector, and LLM hidden states, but the final readout token fails to encode the mapping from direction to answer option. We call this mismatch a direction binding gap.

3.1. Experimental Setup

Datasets. We introduce MoDirect-SynBench, a controlled synthetic benchmark for signed image-plane motion direction. MoDirect-SynBench follows a 2×2 design over foreground type and background type: Primitive-on-Syn, Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real.

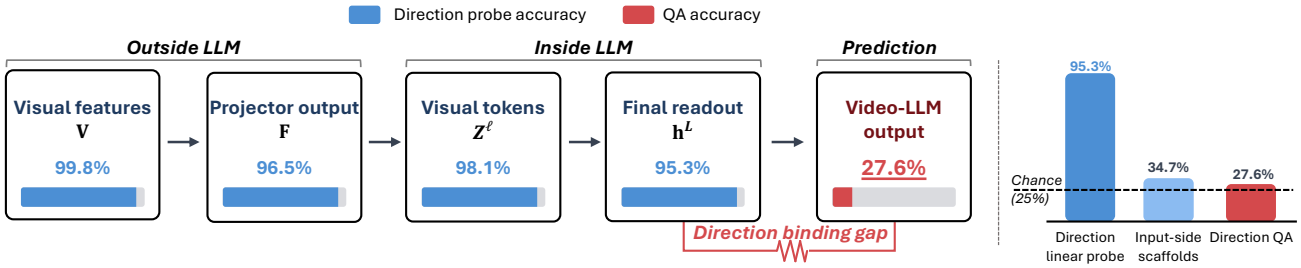


Figure 2. **Direction is decodable, but not converted into the answer.** On Primitive-on-Syn in MoDirect-SynBench, motion direction remains linearly decodable throughout LLaVA-Video-7B (Zhang et al., 2025c), yet QA accuracy stays near chance, exposing the *direction binding gap*.

Each video contains a single foreground object moving in one of four signed directions: left, right, up, down. We randomize start and end positions within the valid frame region and use held-out start–end trajectories for evaluation, reducing static-position shortcuts. In every analysis on MoDirect-SynBench, we train a linear probe on the training split and evaluate it on held-out videos.

Task. We use a four-way multiple-choice question (MCQ) format as the primary evaluation protocol. The four options correspond to left→right, right→left, bottom→top, and top→bottom. Crucially, we randomize the order of answer options for each example. Therefore, the correct answer letter is not a fixed alias of the motion direction; the model must bind the perceived direction to the prompt-specific option text. This design enables single-token, logit-based evaluation and reduces free-form generation confounds, while directly testing whether the model can bind the perceived direction to the correct prompt-specific option (Li et al., 2024a; Robinson et al., 2023; Kim et al., 2026).

Video-LLM pipeline. A Video-LLM processes a video through three stages. Given T sampled frames, a vision encoder produces visual features $\mathbf{V} \in \mathbb{R}^{T\times M\times D_v}$, where M denotes the number of patch tokens per frame and D_v denotes the embedding dimension. A projector maps these features into the LLM embedding space, producing visual tokens $\mathbf{F} \in \mathbb{R}^{T\times N\times D}$, where $N \leq M$ reflects spatial pooling and D is the LLM hidden dimension. An L-layer LLM consumes the visual tokens with text tokens and produces an answer. Inside the LLM, we track the hidden states at the visual-token positions across layers. Let $\mathbf{z}_t^\ell \in \mathbb{R}^D$ denote the spatially pooled hidden state of the visual-token positions at frame t and layer $\ell \in \{1,\ldots,L\}$.

We refer to the hidden state at the last token position used for next-token prediction as the *readout token*, and denote $\mathbf{h}^\ell \in \mathbb{R}^D$ for the readout token at layer ℓ. We call $\mathbf{h}^L$ the *final readout token*; this is the representation the Language Model (LM) head maps to output vocabulary logits.

3.2. Can Data or Prompting Explain the Failure?

Before inspecting model internals, we rule out two external explanations. First, motion-direction supervision is scarce in existing video instruction data. Using keyword pre-filtering followed by semantic classification, we estimate that only 0.91% of LLaVA-Video-178K (Zhang et al., 2025c) examples are direction-dependent; human verification further indicates that these estimates are upper bounds (Appendix E.1). This scarcity motivates controlled motion-direction supervision, but it does not localize where the failure occurs. Second, simple input-side scaffolds do not solve the task. We test visual boundary cues, step-by-step location reasoning (Wei et al., 2022; Kojima et al., 2022), and coordinate-grid prompts (Yin et al., 2026; Chen et al., 2024a; Cheng et al., 2024a); the best combination reaches only 34.7% on Primitive-on-Syn as shown in Figure 2. Thus, external factors alone are insufficient, and we turn to representation-level diagnostics.

3.3. Is Motion Direction Erased by the Vision Encoder or Projector?

We next ask whether the failure is caused by visual processing. We use linear probing as a diagnostic tool (Orgad et al., 2025; Li et al., 2023; Gurnee and Tegmark, 2024): if a linear classifier can recover the signed direction from a frozen representation, then that representation contains direction information in a linearly accessible form. This does not imply that the model can directly use the signal for generation; rather, it tests whether the visual-temporal evidence is present before the final readout.

Vision encoder preserves direction. We train a four-way linear probe on the frozen vision-encoder output $\mathbf{V}$. The probe recovers motion direction with 99.8% accuracy (Figure 2), indicating that the visual encoder preserves a near-perfect direction signal.

Projector also preserves direction. We repeat the same probe on the projector output $\mathbf{F}$. Accuracy remains high at 96.5%, indicating that the vision-language projection step does not erase the signed direction signal. Thus, directional motion blindness is not explained by a lack of linearly accessible direction information in either the vision encoder or the projector. The failure must arise from *how this signal is used or read out by the LLM*.

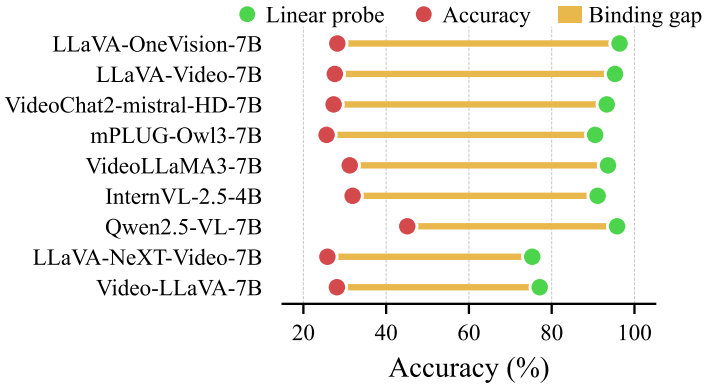


Figure 3. **The direction binding gap is universal across Video-LLMs.**

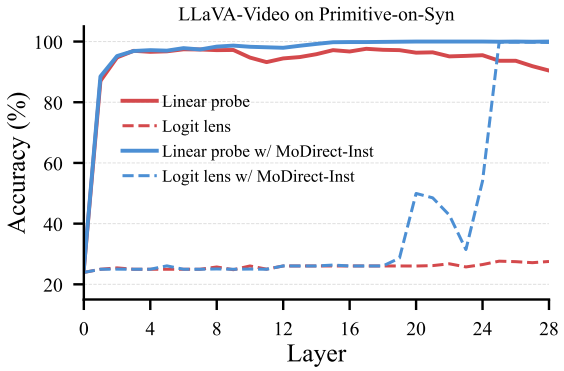


Figure 4. **Instruction tuning closes the direction binding gap (source domain).**

3.4. Can the Readout Bind Motion Direction to the Answer Option?

Direction information reaches the LLM input in a linearly accessible form. We now ask whether it remains accessible inside the LLM and whether the readout token binds this direction signal to the correct prompt-specific answer option.

Direction remains decodable inside the LLM. We probe two internal representations across LLM layers: the visual tokens $\mathbf{z}_t^\ell$ and the readout token $\mathbf{h}^\ell$. As shown in Figure 2, visual-token probes achieve $98.1\% \pm 1.3$, and readout token probes achieve $95.3\% \pm 2.4$. These results show that the LLM does not simply discard the direction signal.

The bottleneck is answer-option binding. The key question is whether the internally decodable direction signal reaches the correct *answer option*. Because the MCQ option order is randomized, the answer letter is not a fixed alias of the motion direction; the model must bind the perceived direction to the option text in the current prompt. Yet the model achieves only 27.6% MCQ accuracy, near the 25% chance level (Figure 2). This shows that the model carries the direction signal internally but fails to bind it to the prompt-specific answer option.

Direction binding gap. Taken together, these results show that the bottleneck lies not in representing motion direction, but in converting that representation into the correct answer option. The direction signal remains accessible across the vision encoder, the projector, and the readout state, yet it is not reliably bound to the prompt-specific answer option (Orgad et al., 2025; Sun et al., 2025; Park et al., 2025). We call this mismatch the *direction binding gap*. As shown in Figure 3, this gap persists across architectures and scales, indicating a shared structural limitation rather than a model-specific artifact. We further verify in Appendix E.4 that the same failure appears in open-ended direction-answer generation, confirming that the phenomenon is not an artifact of the MCQ format. This diagnosis motivates Section 4, where we test whether controlled motion direction instruction tuning can close the gap.

4. What Instruction Tuning Solves and What It Does Not

Section 3 localizes directional motion blindness to answer-option binding: direction is linearly accessible inside the model, but the readout does not reliably link it to the correct answer option. We now test the most direct remedy: *motion direction instruction tuning*. We fine-tune on the simplest synthetic domain and ask whether the learned binding transfers to more complex visual conditions.

4.1. Setup: Training on MoDirect-Inst, Testing on MoDirect-SynBench

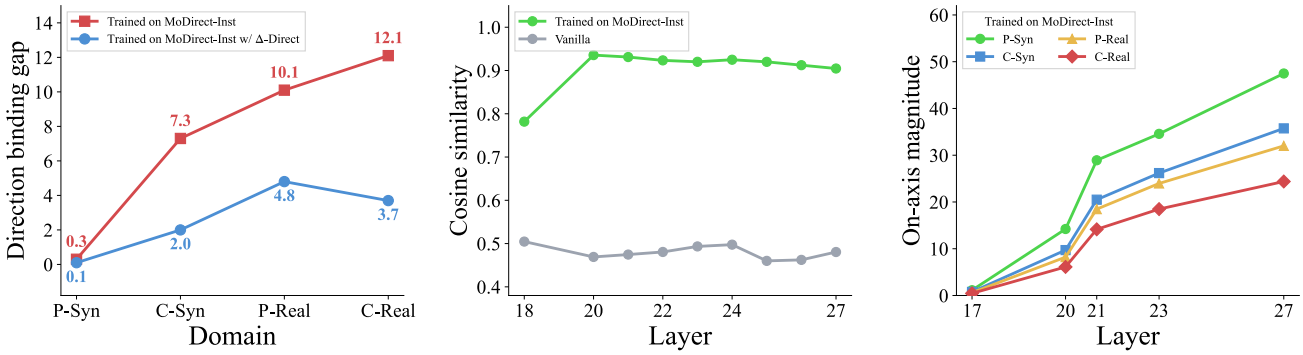
Training data. We fine-tune a Video-LLM on MoDirect-Inst, a synthetic instruction dataset built from the Primitive-on-Syn domain. MoDirect-Inst contains 100K training videos with diverse motion direction-related QA formats, including direction MCQ, open-ended direction, and static appearance questions (e.g., object color and shape, to preserve general visual recognition) .

Evaluation data. We use MoDirect-SynBench as our testbed. MoDirect-SynBench includes Primitive-on-Syn, the source domain used to construct MoDirect-Inst, and three OOD domains: Primitive-on-Real, Cutout-on-Syn, and Cutout-on-Real. These OOD domains introduce natural scene backgrounds (Zhou et al., 2017), realistic object cutouts (Lin et al., 2014), or both. This split allows us to distinguish source domain binding from domain-invariant direction understanding.

Training. We use LLaVA-Video-7B (Zhang et al., 2025c) as the backbone, keep the vision encoder and original LLM weights frozen, and update only the projector and LoRA (Hu et al., 2022) adapters in the LLM.
We provide the comprehensive details of the analysis in Appendix B.3.

4.2. Source Domain: Answer-Option Binding Emerges

We evaluate whether instruction tuning closes the direction binding gap on the held-out Primitive-on-Syn test set. We repeat the layer-wise probing analysis from Section 3. To



(a) OOD binding gap (b) Cross-domain geometry (c) Direction signal magnitude

Figure 5. **Shared orientation, weak magnitude. (a)** Instruction tuning closes the binding gap on Primitive-on-Syn, but the gap reopens on OOD domains; `DeltaDirect` narrows it across domains. **(b)** Direction concept vector orientations align across domains after instruction tuning, with late-layer cosine similarity exceeding 0.9. **(c)** Despite this alignment, concept-vector magnitude decreases with visual complexity, revealing the *magnitude deficit* behind the OOD binding gap.

diagnose answer-option binding, we use the *logit lens* (nostalgebraist, 2020; Kim et al., 2026), which projects each intermediate readout token $\mathbf{h}^\ell$ through the LM head without training an external classifier. This measures whether the LM head can recover the correct answer letter at each layer. As shown in Figure 4, in the vanilla model, direction is highly decodable from the readout token (—), but its logit lens accuracy stays near chance (- -), matching its low MCQ accuracy of 27.6%. After instruction tuning, logit lens accuracy (- -) rises in the late layers and converges to direction probing accuracy (—) in the final layers. MCQ accuracy correspondingly increases to 99.5%. Thus, direction-focused instruction tuning establishes a source domain binding pathway from the internal direction signal to the correct answer option.

4.3. Out-of-Domain: The Binding Gap Reopens

We next ask whether this learned binding transfers to more complex visual conditions. As shown in Figure 5 (a), instruction tuning nearly closes the gap on the source domain Primitive-on-Syn test set (0.3 points), but the gap reopens on OOD domains, e.g., 12.1 points on Cutout-on-Real. Although direction remains decodable from the final readout token, MCQ accuracy drops as object and background complexity increase. This suggests that next-token supervision learns source domain answer binding, but does not produce a direction signal strong enough for reliable OOD readout.

4.4. Why Does the Binding Gap Reopen? Shared Orientation but Insufficient Magnitude

To understand the OOD failure, we analyze *motion direction concept vectors* using difference-in-means (Arditi et al., 2024; Marks and Tegmark, 2024; Li et al., 2023; Rimsky et al., 2024; Tigges et al., 2024) at the readout state after instruction tuning. For each domain A, direction class d, and layer ℓ, we define $\mathbf{v}^\ell_{d,A} = \mathbb{E}[\mathbf{h}^\ell \mid y = d, A] - \mathbb{E}[\mathbf{h}^\ell \mid A]$. Here we compute the expectation over the MoDirect-SynBench samples. We decompose each vector into its unit orientation $\hat{\mathbf{v}}^\ell_{d,A} = \mathbf{v}^\ell_{d,A} / \|\mathbf{v}^\ell_{d,A}\|$ and magnitude $\|\mathbf{v}^\ell_{d,A}\|$. We use "orientation" here to distinguish the representation-space axis from the physical motion direction.

Orientation aligns across domains. To test whether the representation space orientation of each motion direction is shared across domains, we compare unit motion direction concept vectors across domains. For each layer ℓ, direction $d \in \{\text{left}, \text{right}, \text{up}, \text{down}\}$, and unordered pair of domains (A, B), we compute $\cos\left(\hat{\mathbf{v}}^\ell_{d,A}, \hat{\mathbf{v}}^\ell_{d,B}\right)$. Since MoDirect-SynBench has four domains, this gives $\binom{4}{2} = 6$ domain pairs and therefore $4 \times 6 = 24$ cosine similarity values per layer. The cosine similarity reported in Figure 5 (b) is the average over these values. After instruction tuning, this average rises sharply in the late layers, indicating that the model learns a shared motion direction concept vector orientation for each motion direction across domains, despite being trained on Primitive-on-Syn.

Magnitude drops under visual complexity. The aligned orientation alone is insufficient. As shown in Figure 5 (c), the same late layers exhibit a magnitude drop under increasing visual complexity; in particular, Cutout-on-Real shows a clear magnitude deficit relative to Primitive-on-Syn. This mirrors the OOD MCQ accuracy drop and suggests a *magnitude deficit*: although domains share a motion direction concept vector orientation for each motion direction, the corresponding signal magnitude is too weak in complex domains to support reliable answer-option binding.

Restoring magnitude recovers accuracy. As a diagnostic intervention, for each OOD sample with ground-truth motion direction d, we rescale $\mathbf{h}^\ell$ along the corresponding unit concept-vector orientation $\hat{\mathbf{v}}^\ell_{d,A}$ to match the average Primitive-on-Syn magnitude, leaving the orthogonal component unchanged. We observe that this intervention improves OOD MCQ accuracy, with the largest gain of 15.5 points on Cutout-on-Real. These results suggest that the OOD binding gap is not caused by a missing shared orientation across do-

mains, but by *insufficient signal magnitude*. For more results and details, refer to Appendix E.8.

This motivates a training objective (Section 5) that does not merely teach the answer token, but makes the projector output carry a stronger signed displacement signal before it enters the LLM.

5. DeltaDirect: Strengthening Motion Signals at the VL Interface

The preceding sections suggest a specific intervention point. Directional motion blindness is not caused by the complete absence of direction evidence: direction is linearly accessible from the vision encoder, projector output, and LLM hidden states (Section 3). However, this signal is not reliably bound to the correct answer option. Moreover, motion direction instruction tuning closes the binding gap on the source domain, but visual complexity weakens the magnitude of the shared concept-vector orientation, reopening the gap in out-of-domains (Section 4).

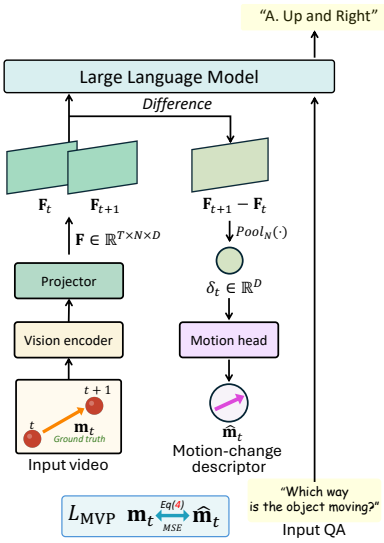


Figure 6. **DeltaDirect.**

These observations motivate a simple design principle: make *the projector output carry a stronger signed displacement signal* before it enters the LLM. We therefore introduce DeltaDirect, a training-only auxiliary objective applied to the projector output. Instead of adding learned motion tokens, or a motion-specific encoder at inference time, DeltaDirect uses synthetic 2-D motion vectors as supervision during training. The auxiliary branch is discarded after training, so the test-time input format, token sequence, model architecture, and decoding procedure remain unchanged.

These observations lead to three design requirements. First, the supervision should act at the projector output, where visual tokens enter the LLM. Second, it should target motion-induced change rather than static appearance, so we supervise adjacent-frame feature deltas. Third, it should supervise signed displacement directly, rather than only the answer word, so we predict normalized 2-D motion vectors. DeltaDirect instantiates these requirements with a training-only branch.

5.1. Motion-Induced Feature Deltas

DeltaDirect operates on the projector output $\mathbf{F} \in \mathbb{R}^{T\times N\times D}$. For each adjacent frame pair, we spatially pool the temporal feature difference to obtain a motion-change descriptor:

$$\delta_t = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{F}_{t+1}[n] - \mathbf{F}_t[n]) \in \mathbb{R}^D, \qquad t = 1,\dots,T-1. \tag{1}$$

Feature differencing suppresses content that remains stable across adjacent frames and focuses the auxiliary branch on motion-induced change. We compute deltas at the projector output because these are the visual tokens directly passed to the LLM. This lets the auxiliary branch emphasize motion-induced changes in the same representation space used for language readout, without modifying the vision encoder or adding inference-time motion inputs.

5.2. Training-Only Motion Vector Prediction

Motion vector target. Because MODIRECT-INST is synthetic, the object center $\mathbf{c}_t \in \mathbb{R}^2$ is known at each frame. For each moving frame pair, we define the normalized 2-D motion target

$$\mathbf{m}_t = \frac{\mathbf{c}_{t+1} - \mathbf{c}_t}{\|\mathbf{c}_{t+1} - \mathbf{c}_t\|_2 + \varepsilon} \in \mathbb{R}^2. \tag{2}$$

This target captures the signed image-plane displacement direction while removing speed as a confounding factor. Rather than treating directions as discrete labels for an auxiliary classifier, the normalized vector target preserves the geometry of signed displacement: opposite motions have opposite targets, and different directions remain geometrically related in the same 2-D space.

Prediction head. A lightweight linear head predicts the motion vector from each pooled delta:

$$\hat{\mathbf{m}}_t = \mathbf{W}\,\delta_t + \mathbf{b}, \qquad \mathbf{W} \in \mathbb{R}^{2\times D},\ \mathbf{b} \in \mathbb{R}^2. \tag{3}$$

The head introduces only $2D+2$ trainable parameters and is used only during training.

MVP loss. We define the Motion Vector Prediction loss as

$$L_{\text{MVP}} = \frac{1}{T-1}\sum_{t=1}^{T-1}\|\hat{\mathbf{m}}_t - \mathbf{m}_t\|_2^2. \tag{4}$$

This objective does not ask the model to memorize a direction word. Instead, it directly shapes the projector output so that adjacent-frame changes preserve signed displacement cues before entering the LLM. We apply MVP supervision at the projector output by default, and compare it with intermediate-layer and readout-position supervision in Appendix F.

5.3. Training Objective and Inference

The final training objective combines standard next-token prediction with the MVP loss:

Table 1. **`DeltaDirect` achieves state-of-the-art motion direction understanding.** We report Top-1 accuracy (%) on MoDirect-SynBench and MoDirect-RealBench.

Method	MoDirect-SynBench					MoDirect-RealBench				Overall Avg.
	P-Syn	C-Syn	P-Real	C-Real	Avg.	SSv2 (Goyal et al., 2017)	KTH (Schuldt et al., 2004)	TOMATO (Shangguan et al., 2025)	Avg.	
Random Chance	25.0	25.0	25.0	25.0	25.0	50.0	50.0	20.0	40.0	31.4
GPT-4o (Hurst et al., 2024)	46.6	37.8	48.0	40.7	43.3	51.0	52.3	34.5	45.9	44.4
Gemini 2.5 Flash (Comanici et al., 2025)	58.1	43.4	61.7	50.6	53.5	20.0	69.4	25.6	38.3	47.0
Video-LLaVA-7B (Lin et al., 2024)	28.1	27.8	25.5	27.2	27.2	49.2	49.5	14.6	37.8	31.7
VideoChat2-HD-7B (Li et al., 2024a)	26.1	23.5	24.6	24.1	24.6	50.0	54.2	20.1	41.4	31.8
LLaMA-VID-7B (Li et al., 2024c)	25.3	25.6	24.7	25.0	25.2	51.4	54.2	16.9	40.8	31.9
LLaVA-NeXT-Video-7B	25.8	24.8	25.0	25.2	25.2	50.8	52.4	21.1	41.4	32.2
LLaVA-OneVision-7B (Li et al., 2025a)	28.3	23.4	29.9	29.3	27.7	52.2	50.3	26.8	43.1	34.3
Qwen2.5-VL-7B (Bai et al., 2025b)	45.1	32.5	30.8	30.2	34.7	52.6	44.2	25.3	40.7	37.2
Qwen3-VL-4B (Bai et al., 2025a)	66.6	50.2	40.8	41.1	49.7	60.9	62.1	35.0	52.7	51.0
InternVL-2.5-4B (Chen et al., 2024b)	31.9	31.8	30.4	31.4	31.4	66.1	50.7	26.1	47.6	38.3
VideoLLaMA3-7B (Zhang et al., 2025a)	56.4	46.6	50.0	48.3	50.3	56.5	52.2	19.9	42.9	47.1
LLaVA-Video-7B (Zhang et al., 2025c)	27.6	23.4	26.9	25.8	25.9	52.2	50.3	26.8	43.1	33.3
LLaVA-Video-7B w/ FlashVID (Fan et al., 2026)	25.2	25.1	24.3	24.8	24.9	52.5	50.3	21.8	41.5	32.0
LLaVA-Video-7B w/ MoDirect-Inst	99.5	80.7	74.7	60.5	78.9	72.4	66.6	35.2	58.1	69.9
LLaVA-Video-7B w/ `DeltaDirect`	**99.7**	**84.9**	**85.2**	**71.7**	**85.4**	**81.5**	**74.8**	**38.8**	**65.0**	**76.7**
Qwen2-0.5B Full-FT w/ MoDirect-Inst	99.5	97.3	62.5	51.5	77.7	59.0	55.2	20.6	44.9	63.7
Qwen2-0.5B Full-FT w/ `DeltaDirect`	**99.7**	**98.7**	**91.0**	**80.1**	**92.4**	**77.8**	**69.4**	**21.7**	**56.3**	**76.9**

Table 2. **Where to apply supervision.**

Stage	MoDirect		MVBench
	SynBench	RealBench	
None	78.9	58.1	**60.9**
Vision encoder	80.4	59.3	**60.9**
Pre-projector	**86.1**	62.2	60.2
Post-projector	85.4	**65.0**	60.7
LLM visual tokens	70.0	58.3	57.4
Final readout	27.7	42.5	59.9

Table 3. **What motion signal to supervise.**

Supervision target	MoDirect		MVBench
	SynBench	RealBench	
MoDirect-Inst tuning only	78.9	58.1	**60.9**
Frame order	24.7	40.3	54.5
Concat. feature deltas	80.1	58.0	60.7
Delta equivariance	78.4	58.0	60.6
`DeltaDirect`	**85.4**	**65.0**	60.7

$$L_{\text{total}} = L_{\text{LM}} + \lambda L_{\text{MVP}}, \tag{5}$$

where L_{LM} is the next-token cross-entropy on answer tokens and λ controls the auxiliary loss weight. The MVP branch updates only the projector and prediction head; the LLM decoder is trained solely by L_{LM}. During inference, the MVP branch is removed, leaving the original input format, token sequence, and decoding procedure unchanged.

6. Experimental Results

We evaluate whether `DeltaDirect` turns our diagnosis into practical gains. We focus on three questions: **(i)** Does it reduce the diagnosed binding gap? **(ii)** Does `DeltaDirect` improve synthetic and real-world motion-direction understanding without real-world direction supervision? **(iii)** Does it preserve general video understanding? Implementation details, benchmark protocols, and additional analyses and ablations are provided in Appendix B, F.

6.1. Improving Motion Direction without Sacrificing General Video Understanding

The direction binding gap narrows. In Figure 5 (a), instruction tuning on MoDirect-Inst leaves a widening binding gap (12.1 on Cutout-on-Real) under visual complexity. `DeltaDirect` reduces this gap to 3.7 points on Cutout-on-Real and narrows it on every OOD domain.

Table 4. **General video understanding results.**

Method	Standard	Fine-grained
LLaVA-Video-7B (Zhang et al., 2025c)	69.4	47.3
LLaVA-Video-7B w/ `DeltaDirect`	**70.1**	**48.7**
Qwen2-0.5B Full-FT w/ MoDirect-Inst	58.7	32.8
Qwen2-0.5B Full-FT w/ `DeltaDirect`	**59.4**	**33.3**

`DeltaDirect` substantially improves motion direction. Table 1 shows that MoDirect-Inst tuning nearly solves Primitive-on-Syn (99.5%) but transfers poorly to the hardest synthetic OOD domain, Cutout-on-Real, (60.5%). `DeltaDirect` targets this gap, improving Cutout-on-Real to 71.7% and raising MoDirect-SynBench average accuracy from 78.9% to 85.4% over MoDirect-Inst alone. It also transfers to real videos without real-world tuning, improving MoDirect-RealBench Avg. from 43.1% to 65.0% over vanilla LLaVA-Video-7B. In full fine-tuning, the same trend holds: `DeltaDirect` raises MoDirect-SynBench performance from 77.7% to 92.4% and MoDirect-RealBench Avg. from 44.9% to 56.3%.

`DeltaDirect` improves general video understanding. Although `DeltaDirect` targets motion direction, it improves aggregate general video performance in Standard (Li et al., 2024a; Xiao et al., 2021; Patraucean et al., 2023; Man-

galam et al., 2023; Jang et al., 2017) and Fine-grained (Liu et al., 2024b; Zhang et al., 2024a; Tu et al., 2025; Hong et al., 2025) video benchmarks. For LLaVA-Video-7B, Standard and Fine-grained averages increase from 69.4 to 70.1 and from 47.3 to 48.7, respectively. In the full fine-tuning setup, `DeltaDirect` further improves over MoDirect-Inst on both averages (58.7 → 59.4, 32.8 → 33.3). Thus, direction supervision does not trade off broader video capability.

6.2. Ablation Study

Where to apply supervision. In Table 2, we apply MVP supervision at various locations in a Video-LLM. Post-projector supervision performs best on average, while readout supervision performs poorly despite the binding gap being observed there. This supports strengthening motion signals at the vision-language interface, before the LLM transforms them for answer generation.

What motion signal to supervise. In Table 3, we compare `DeltaDirect` with various supervision targets including a direction-aware delta equivariance objective. `DeltaDirect` performs best, indicating that signed 2-D motion-vector supervision is most effective.

7. Conclusions

In this work, we identify *directional motion blindness*, a systematic failure of Video-LLMs to resolve signed image-plane motion direction. Through controlled diagnosis on MoDirect, we localize the failure to a *direction binding gap*: motion direction remains linearly accessible inside the model, but is not reliably connected to the correct verbal response. Instruction tuning closes this gap on the source domain, but visual complexity reopens it through a magnitude deficit in the shared direction representation. We address this with `DeltaDirect`, a training-only objective that predicts motion vectors from adjacent-frame projector-feature deltas, strengthening signed displacement cues without changing inference. Trained only with synthetic motion supervision, `DeltaDirect` improves synthetic and real-world direction understanding while preserving general video understanding. These results suggest that diagnosis-driven supervision can help close perception–language gaps in Video-LLMs.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *NeurIPS*, 2024. 2, 6, 26, 29

Kyungho Bae, Jinhyung Kim, Sihaeng Lee, Soonyoung Lee, Gunhee Lee, and Jinwoo Choi. Mash-vlm: Mitigating action-scene hallucination in video-llms through disentangled spatial-temporal representations. In *CVPR*, 2025. 1, 3

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a. 3, 8, 15, 25, 30, 38, 39

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b. 3, 8, 25, 39

Richard T Born and David C Bradley. Structure and function of visual area mt. *Annu. Rev. Neurosci.*, 28(1):157–189, 2005. 1

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, 2024a. 4

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024b. 8, 15, 21, 25, 39

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. In *NeurIPS*, 2024a. 4

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024b. 1, 3

Jihoon Chung, Tyler Zhu, Max Gonzalez Saez-Diez, Juan Carlos Niebles, Honglu Zhou, and Olga Russakovsky. Unifying specialized visual encoders for video language models. In *ICML*, 2025. 3

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 1, 3, 8, 39

Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees G. M. Snoek, and Yuki M. Asano. Tvbench: Redesigning video-language evaluation. *arXiv:2410.07752*, 2024. 3

Nieves Crasto, Philippe Weinzaepfel, Karteek Alahari, and Cordelia Schmid. MARS: Motion-augmented RGB stream for action recognition. In *CVPR*, 2019. 3

Yipeng Du, Tiehan Fan, Kepan Nan, Rui Xie, Penghao Zhou, Xiang Li, Jian Yang, Zhenheng Yang, and Ying Tai. Motion-sight: Boosting fine-grained motion understanding in multi-modal LLMs. In *ICLR*, 2026. 3

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging. In *ICLR*, 2026. 8, 15, 39

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *CVPR*, 2025a. 1

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, Jingqun Tang, Wei Chen, Lianwen Jin, Yuliang Liu, and Xiang Bai. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. In *NeurIPS*, 2025b. 21

James J Gibson. *The ecological approach to visual perception: classic edition*. Psychology press, 2014. 1

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 2, 8, 22, 39

Wes Gurnee and Max Tegmark. Language models represent space and time. In *ICLR*, 2024. 4, 27, 28

Wenyi Hong, Yean Cheng, Zhuoyi Yang, Weihan Wang, Lefan Wang, Xiaotao Gu, Shiyu Huang, Yuxiao Dong, and Jie Tang. Motionbench: Benchmarking and improving fine-grained video motion understanding for vision language models. In *CVPR*, 2025. 1, 2, 3, 9, 14, 19, 38, 39

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 5

Ziyuan Huang, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Rong Jin, and Marcelo H. Ang. Self-supervised motion learning from static images. In *CVPR*, 2021. 3

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 3, 8, 24, 39

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 9, 14, 18, 38, 39

Yang Jin, Zhicheng Sun, Kun Xu, Kun Xu, Liwei Chen, Hao Jiang, Quzhe Huang, Chengru Song, Yuliang Liu, Di Zhang, Yang Song, Kun Gai, and Yadong Mu. Video-LaVIT: Unified video-language pre-training with decoupled visual-motional tokenization. In *ICML*, 2024. 3

Minji Kim, Taekyung Kim, and Bohyung Han. Map the flow: Revealing hidden pathways of information in videoLLMs. In *ICLR*, 2026. 3, 4, 6

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022. 4

Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. MotionSqueeze: Neural motion feature learning for video understanding. In *ECCV*, 2020. 3

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *TMLR*, 2025a. 1, 3, 8, 15, 20, 21, 25, 30, 38, 39

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *NeurIPS*, 2023. 2, 4, 6, 26, 29

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024a. 1, 2, 3, 4, 8, 14, 15, 16, 17, 20, 35, 38, 39

Lei Li, Yuanxin Liu, Linli Yao, Peiyuan Zhang, Chenxin An, Lean Wang, Xu Sun, Lingpeng Kong, and Qi Liu. Temporal reasoning transfer from text to video. In *ICLR*, 2025b. 1, 3

Shicheng Li, Lei Li, Yi Liu, Shuhuai Ren, Yuanxin Liu, Rundong Gao, Xu Sun, and Lu Hou. Vitatecs: A diagnostic dataset for temporal concept understanding of video-language models. In *ECCV*, 2024b. 3

Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: Temporal excitation and aggregation for action recognition. In *CVPR*, 2020. 3

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2024c. 1, 3, 8, 39

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *EMNLP*, 2024. 1, 3, 8, 25, 39

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2, 5

Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *ECCV*, 2024a. 3

Ruyang Liu, Shangkun Sun, Haoran Tang, Ge Li, and Wei Gao. Flow4Agent: Long-form video understanding via motion prior from optical flow. In *ICCV*, 2025a. 3

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? In *ACL*, 2024b. 1, 2, 3, 9, 14, 18, 38, 39

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, Ranjay Krishna, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. NVILA: Efficient frontier visual language models. In *CVPR*, 2025b. 3

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 3, 20

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *NeurIPS*, 2023. 1, 3, 8, 14, 18, 38, 39

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. 1, 2, 3, 6, 26, 27, 28

Ken Nakayama. Biological image motion processing: a review. *Vision Research*, 25(5):625–660, 1985. doi: 10.1016/0042-6989(85)90171-3. 1

Hong Nguyen, Dung Tran, Hieu Hoang, Phong Nguyen, and Shrikanth Narayanan. MOOSE: Pay attention to temporal dynamics for video understanding via optical flows. *arXiv preprint arXiv:2506.01119*, 2025. 3

nostalgebraist. Interpreting GPT: The logit lens. LessWrong, August 2020. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2026-02-22. 6, 20, 27

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations. In *ICLR*, 2025. 1, 3, 4, 5, 27

Yoonah Park, Haesung Pyun, and Yohan Jo. Bridging the knowledge-prediction gap in llms on multiple-choice questions. *arXiv preprint arXiv:2509.23782*, 2025. 5, 27

Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2023. 1, 2, 8, 14, 18, 38, 39

Daiqing Qi, Dongliang Guo, Hanzhang Yuan, Handong Zhao, Mengxuan Hu, Lehan Yang, and Sheng Li. Improve temporal reasoning in multimodal large language models via video contrastive decoding. In *NeurIPS*, 2025. 1

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander 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)*, 2024. 2, 6, 26, 29

Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering. arxiv (2022). In *ICLR*, 2023. 4

Mohammadreza Salehi, Jae S Park, Tanush Yadav, Aditya Kusupati, Ranjay Krishna, Yejin Choi, Hannaneh Hajishirzi, and Ali Farhadi. Actionatlas: A videoqa benchmark for domain-specialized action recognition. In *NeurIPS*, 2024. 1

Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 2, 8, 39

Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models. In *ICLR*, 2025. 1, 2, 3, 8, 17, 39

Jonathan C. Stroud, David A. Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3D: Distilled 3D networks for video action recognition. In *WACV*, 2020. 3

Yucheng Sun, Alessandro Stolfo, and Mrinmaya Sachan. Probing for arithmetic errors in language models. In *EMNLP*, 2025. 5, 27

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. In *ICML*, 2024. 2, 6, 26, 29

Chongjun Tu, Lin Zhang, Pengtao Chen, Peng Ye, Xianfang Zeng, Wei Cheng, Gang YU, and Tao Chen. Favor-bench: A comprehensive benchmark for fine-grained video motion understanding. In *NeurIPS*, 2025. 2, 3, 9, 14, 19, 38, 39

Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. TDN: Temporal difference networks for efficient action recognition. In *CVPR*, 2021. 3

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024. 1, 3

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. 4

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019. 14

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 8, 14, 18, 38, 39

Zihui Xue, Mi Luo, and Kristen Grauman. Seeing the arrow of time in large multimodal models. In *NeurIPS*, 2025. 3

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. 15, 34, 35

Haosen Yang, Deng Huang, Bin Wen, Jiannan Wu, Hongxun Yao, Yi Jiang, Xiatian Zhu, and Zehuan Yuan. Self-supervised video representation learning with motion-aware masked autoencoders. *arXiv preprint arXiv:2210.04154*, 2022. 3

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mPLUG-Owl3: Towards long image-sequence understanding in multi-modal large language models. In *ICLR*, 2025. 15, 25, 38, 39

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, et al. Spatial mental modeling from limited views. In *ICLR*, 2026. 4

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *CVPR*, 2023. 15, 22, 34, 35

Yu-Wei Zhan, Xin Wang, Hong Chen, Tongtong Feng, Wei Feng, Ren Wang, Guangyao Li, Qing Li, and Wenwu Zhu. PhyVLLM: Physics-guided video language model with motion-appearance disentanglement. *arXiv preprint arXiv:2512.04532*, 2025. 3

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a. 3, 8, 25, 39

Jianrui Zhang, Mu Cai, and Yong Jae Lee. Vinoground: Scrutinizing lmms over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*, 2024a. 3, 9, 14, 19, 38, 39

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *NAACL 2025*, 2025b. 14

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024b. URL https://llava-vl.github.io/blog/2024-04-30-llava-next-video/. 1, 25

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. LLaVA-video: Video instruction tuning with synthetic data. *TMLR*, 2025c. 3, 4, 5, 8, 15, 20, 22, 24, 25, 26, 38, 39

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. Efficient motion-aware video mllm. In *CVPR*, 2025a. 3

Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, Jing Liu, et al. Needle in a video haystack: A scalable synthetic evaluator for video mllms. In *ICLR*, 2025b. 1

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 40(6):1452–1464, 2017. 2, 5

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *CVPR*, 2025a. 1

Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *ICCV*, 2025b. 3

Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G. Hauptmann. Hidden two-stream convolutional networks for action recognition. In *ACCV*, 2018. 3

A. Appendix

Table of Contents

A Implementation Details 14

A.1 Evaluation 14

A.2 Model Card 14

A.3 Training Details: LoRA Fine-tuning 15

A.4 Training Details: Full Fine-tuning with `DeltaDirect` 15

B Dataset and Benchmarks 16

B.1 MoDirect 16

B.2 General Video Benchmarks 17

C Analysis Details 19

C.1 Linear Probing Details 19

C.2 Logit Lens Details 20

D Additional Analysis 20

D.1 Video LLMs Instruction-tuning Dataset Analysis 20

D.2 Input-side Scaffolds: Design and Full Results 21

D.3 Visual Perception: Additional Results and Controls 22

D.4 Motion Direction Binding Gap Beyond MCQ Format 24

D.5 Motion Direction Binding Gap Across Video-LLMs 24

D.6 Out-of-Domain: The Binding Gap Reopens 25

D.7 Direction Concept Vector Analysis 26

D.8 Diagnostic Intervention: Additional Results and Controls 30

D.9 Delta Feature Validation 31

E Additional Experimental Results 34

E.1 Ablation Study of `DeltaDirect` 34

E.2 `DeltaDirect` Across Video-LLM Backbones 37

E.3 Quantitative Results 37

E.4 Additional Analysis of `DeltaDirect` 38

F Case Study 40

G Limitations 41

H Broader Impacts 41

B. Implementation Details

To support full reproducibility, we will publicly release the complete training and evaluation code, along with model checkpoints and the full MoDirect in the future.

This section reports the implementation details for all experiments in the paper. Unless otherwise stated, all reported experiments follow three global conventions. We use 8× NVIDIA A6000 48GB GPUs for training and 8× NVIDIA RTX 4090 GPUs for evaluation. We use `lmms-eval` (Zhang et al., 2025b) wherever the benchmark is supported; for the remaining benchmarks we follow each benchmark's official evaluation script, or implement the protocol directly when no official harness is available. We uniformly sample $T=8$ frames per video at both training and inference time. We will release our full training and evaluation code upon publication.

B.1. Evaluation

In this section, we provide the full evaluation details. Unless otherwise specified, all inference runs are conducted using LMMS-Eval (Zhang et al., 2025b) and Hugging Face (Wolf et al., 2019) implementations. We provide the sources and model cards of all evaluated models in Appendix B.2. We will publicly release MoDirect-SynBench, MoDirect-RealBench, and all trained model weights to support reproducibility.

B.1.1. MoDirect

We evaluate both MoDirect-SynBench and MoDirect-RealBench using the LMMS-Eval framework. For each video, we uniformly sample 8 frames and ask a four-way multiple-choice question about the signed image-plane motion direction. The model is allowed to generate up to 16 new tokens. We parse the generated response by first extracting an answer option, and then compare it with the ground-truth option to compute accuracy. All evaluations are run on eight NVIDIA RTX 4090 GPUs with a per-device batch size of 1.

B.1.2. Standard Video Benchmarks

MVBench. MVBench (Li et al., 2024a) evaluates general video understanding across diverse temporal reasoning tasks.

NExT-QA. NExT-QA (Xiao et al., 2021) evaluates causal and temporal reasoning in video question answering. We sample 16 frames per video and report the resulting accuracy.

Perception Test. Perception Test (Patraucean et al., 2023) evaluates multimodal video understanding across perception-oriented reasoning tasks.

EgoSchema. EgoSchema (Mangalam et al., 2023) evaluates long-form egocentric video understanding through multiple-choice question answering. We use the publicly released subset, sample 32 frames per video, and report accuracy.

TGIF-QA. TGIF-QA (Jang et al., 2017) evaluates short-video question answering with emphasis on actions, transitions, and temporal dynamics.

B.1.3. Fine-grained Video Benchmarks

TempCompass. TempCompass (Liu et al., 2024b) evaluates fine-grained temporal reasoning in videos, including event order, duration, and temporal relations. We exclude the captioning subtask.

VinoGround. VinoGround (Zhang et al., 2024a) evaluates compositional video-language understanding by testing whether models can distinguish subtle visual and temporal differences. We sample 16 frames per video and report only the group score, the strictest of the three official metrics (text, video, and group score).

FAVOR-Bench. FAVOR-Bench (Tu et al., 2025) evaluates fine-grained video understanding under temporally sensitive question-answering settings.

MotionBench. MotionBench (Hong et al., 2025) evaluates motion-centric video understanding with emphasis on fine-grained motion perception and reasoning.

B.2. Model Card

In this section, we provide the Hugging Face checkpoints for all evaluated models. All open-source models use greedy decoding with a maximum of 1024 new tokens. Following prior multiple-choice direction evaluation settings, we use a maximum of 16 new tokens because the answer is determined from the generated option token.

Table 5. Hugging Face checkpoints and GitHub repositories for all evaluated models.

Model	HF Checkpoint / GitHub
GPT	https://platform.openai.com/docs/models
Gemini	https://ai.google.dev/
VideoChat2-HD-7B	OpenGVLab/VideoChat2_HD_stage4_Mistral_7B
Qwen2.5-VL-7B	Qwen/Qwen2.5-VL-7B-Instruct
Qwen3-VL-4B	Qwen/Qwen3-VL-4B-Instruct
VideoLLaMA3-2B	DAMO-NLP-SG/VideoLLaMA3-2B
VideoLLaMA3-7B	DAMO-NLP-SG/VideoLLaMA3-7B
Video-LLaVA-7B	LanguageBind/Video-LLaVA-7B
LLaVA-OneVision-SI-7B	lmms-lab/llava-onevision-qwen2-7b-si
LLaVA-OneVision-7B	lmms-lab/llava-onevision-qwen2-7b-ov
LLaVA-Video-7B	lmms-lab/LLaVA-Video-7B-Qwen2
LLaVA-NeXT-Video-7B	lmms-lab/LLaVA-NeXT-Video-7B
LLaMA-VID-7B	YanweiLi/llama-vid-7b-full-224-video-fps-1
InternVL-2.5-2B	OpenGVLab/InternVL2_5-2B
InternVL-2.5-4B	OpenGVLab/InternVL2_5-4B
mPLUG-Owl3-7B	https://github.com/x-plug/mplug-owl
FlashVID	https://github.com/Fanziyang-v/FlashVID
`DeltaDirect` (LLaVA-Video-7B)	TBD

B.2.1. FlashVID

FlashVID (Fan et al., 2026) is a training-free inference acceleration method for Video-LLMs that reduces the number of visual tokens by selecting representative tokens and merging spatiotemporally redundant ones. We evaluate FlashVID as a plug-and-play inference wrapper on three open-source Video-LLMs: LLaVA-OneVision-7B-OV, LLaVA-Video-7B-Qwen2, and Qwen2.5-VL-7B-Instruct. For all models, we uniformly sample T=8 frames and keep each model's default video preprocessing pipeline. We use the default hyperparameters from the official FlashVID GitHub implementation and perform no per-task tuning.

B.2.2. mPLUG-Owl3

mPLUG-Owl3 (Ye et al., 2025) is a Video-LLM that integrates hyper-attention blocks into each language model layer for efficient long image-sequence and video understanding. We evaluate the official 7B checkpoint released on Hugging Face, uniformly sampling T=8 frames per video and following the model's default video preprocessing pipeline and chat template. We use the default inference hyperparameters from the official repository and perform no per-task tuning.

B.3. Training Details: LoRA Fine-tuning

We perform LoRA fine-tuning with two codebases. We adopt the official LLaVA-NeXT repository[1] for LLaVA-Video-7B (Zhang et al., 2025c) and LLaVA-OneVision-7B (Li et al., 2025a), and LLaMA-Factory[2] for Qwen3-VL-4B (Bai et al., 2025a) and InternVL2.5-2B (Chen et al., 2024b). LoRA fine-tuning takes approximately 7 hours.

B.3.1. DeltaDirect

We provide detailed training configurations for the `DeltaDirect` variant and the associated LoRA setup. We use LLaVA-Video-Qwen2 as the backbone model. This backbone consists of a Qwen2-7B (Yang et al., 2024) large language model, a SigLIP (Zhai et al., 2023) vision encoder with patch size 14 and input resolution 384×384, and a two-layer GELU-based multimodal projector. The vision encoder is kept frozen throughout training.

Auxiliary Motion Direction Objective. `DeltaDirect` introduces an auxiliary supervision signal for motion direction. Specifically, we compute temporal differences between consecutive frame features, apply mean pooling, and feed the resulting representation into a linear prediction head. The auxiliary direction loss is jointly optimized with the main objective using a loss weight of $\lambda_{\text{direct}} = 1.0$.

Optimization Details. Training is performed for 1 epoch using AdamW with a learning rate of 1×10^{-5}, cosine learning-rate decay, a warmup ratio of 0.03, and zero weight decay. We enable mixed precision training with bf16 and tf32, as well as gradient checkpointing. The effective batch size is 144, obtained from 6 GPUs, a per-device batch size of 12, and gradient accumulation over 2 steps. Training uses DeepSpeed ZeRO-2.

Fine-tuning and LoRA Setup. We fine-tune the multimodal projector and the large language model while keeping the vision encoder frozen. The projector and large language model use separate learning rates of 2×10^{-5} and 1×10^{-5}, respectively. LoRA is applied only to the large language model, with rank $r = 64$, scaling factor $\alpha = 128$, and dropout 0.05. The multimodal projector is fully fine-tuned.

Video Processing. Videos are processed using 8 frames with forced sampling. We apply bilinear spatial pooling with stride 2 to reduce the spatial resolution of visual features.

B.3.2. Other models

Qwen3-VL-4B. We fine-tune Qwen3-VL-4B (Bai et al., 2025a) with LoRA at rank 16 on all linear layers, using the LLaMA-Factory. We train for 2 epochs and follow the default configuration for all other hyperparameters.

InternVL2_5-2B. We fine-tune InternVL2_5-2B (Chen et al., 2024b) with LoRA at rank 16 (alpha 32) applied to the LLM attention and MLP modules, using the LLaMA-Factory. We train for 1 epoch and follow the default configuration for all other hyperparameters.

B.4. Training Details: Full Fine-tuning with `DeltaDirect`

Setup. We jointly optimize the vision encoder, projector, and language model of a Qwen2-0.5B (Yang et al., 2024) backbone paired with a SigLIP (Zhai et al., 2023) vision encoder and a two-layer GELU projector, initialized from the Stage-0 LLaVA-OneVision-0.5B (Li et al., 2025a) checkpoint ($\sim$893M trainable parameters). Full fine-tuning takes approximately 52 hours. For the training data, we subsample VideoChat2-Instruction (VideoChat2-IT) (Li et al., 2024a) using the dataset configuration [3], resulting in approximately 354K training samples.

Objective. We combine the standard language-modeling loss with the auxiliary motion-direction loss of `DeltaDirect`. Since not all training samples contain motion-direction vector annotations, we apply the auxiliary loss only to annotated samples and mask it out

[1] https://github.com/LLaVA-VL/LLaVA-NeXT
[2] https://github.com/hiyouga/LLaMA-Factory
[3] https://github.com/byminji/map-the-flow/blob/main/docs/TRAIN.md

otherwise. The final objective is

$$\mathscr{L} = \mathscr{L}_{\text{LM}} + \lambda_{\text{direct}}\, m\, \mathscr{L}_{\text{direct}},$$

where $m = 1$ for samples with motion-direction vector annotations and $m = 0$ otherwise.

Hyperparameters. We train for 2 epochs with AdamW at learning rate 2×10^{-5} for the language model and projector, and 2×10^{-6} for the vision encoder. We use a cosine schedule with warmup ratio 0.03, no weight decay, and an effective batch size of 96 (batch size 2 $\times$ gradient accumulation 6 across 8 GPUs). Each video is force-sampled to T=8 frames with spatial average pooling at stride 2. For more details, please refer to Table 6.

Table 6. Hyperparameters for full fine-tuning.

Hyperparameter	**Value**
Epochs	2
Total steps	9,438
LR (LLM, projector)	2×10^{-5}
LR (vision tower)	2×10^{-6}
LR schedule / warmup ratio	cosine / 0.03
Weight decay	0
Optimizer	AdamW (fused)
Per-device batch size	2
Gradient accumulation	6
GPUs	8
Effective batch size	96
Precision	bf16 + TF32
Gradient checkpointing	on
DeepSpeed	ZeRO Stage 2
Max sequence length	32,768
Frames T	16
Training samples	453,054
Synthetic Shape-Plain	98,399
VideoChat2-IT (Li et al., 2024a)	354,655

C. Dataset and Benchmarks

C.1. MoDirect

In the main text, we introduced MoDirect as a dataset family for motion direction instruction tuning and evaluation. In this section, we provide detailed descriptions of its three components: MoDirect-Inst for instruction tuning, MoDirect-SynBench as a controlled synthetic benchmark for analysis and evaluation, and MoDirect-RealBench for real-world evaluation.

C.1.1. MoDirect-Inst

MoDirect-Inst is a synthetic instruction-tuning dataset constructed from the Primitive-on-Syn (P-Syn) domain, which serves as the source domain for training. The dataset contains 100,000 video–question–answer (QA) pairs with diverse motion-related question formats. Each video consists of 8 frames at a resolution of 384×384, depicting a single object with diverse appearance, motion patterns, and procedural background textures. Examples are shown in Figure 12.

QA composition and answer formats. The dataset contains both multiple-choice and open-ended QA formats, encouraging the model to learn both discrete decision boundaries and natural language grounding of motion concepts. The QA formats include direction prediction, motion description, trajectory understanding, object appearance recognition, location grounding, motion existence prediction, and rotation reasoning. A detailed breakdown of QA categories and their frequencies is provided in Table 7 (a). Direction-related QA pairs consist of both open-ended responses and multiple-choice questions. Specifically, we include both 5-way MCQ settings (4 cardinal directions + stationary) and 9-way MCQ settings (8 directions + stationary) to support both coarse and fine-grained motion understanding. Trajectory MCQs additionally require the model to reason about compound motion patterns, such as sequential direction changes in zigzag or roundtrip trajectories. The dataset also includes appearance-related questions about object and background properties, location grounding questions based on 3×3 spatial regions, and motion existence questions that determine whether an object is moving or stationary. Rotation-related QA pairs are generated only for circular-motion videos, where the model predicts clockwise (CW) or counterclockwise (CCW) motion directions using either multiple-choice or open-ended formats. Some QA categories naturally appear less frequently because they are only valid for specific motion types. For example, rotation QA pairs are generated only for circular-motion videos, while trajectory MCQs are constructed only from multi-stage motion patterns such as zigzag and roundtrip motions.

Dataset statistics. Table 7 further summarizes the distributions of motion patterns, motion directions, and pro-

cedural background textures. The dataset includes diverse motion patterns, including perturbed linear motion, zigzag trajectories, circular motion, roundtrip motion, and static scenes. Motion directions are approximately uniformly distributed across all 8 directions, reducing directional bias during instruction tuning. To increase visual diversity and reduce low-level shortcut learning, the dataset incorporates diverse procedural background textures, including gradients, checkerboards, stripes, noise-based textures, paper-like patterns, and vignette-style backgrounds. The final background distribution is not perfectly uniform because all generated samples are filtered using minimum foreground–background contrast constraints to ensure object visibility.

C.1.2. MoDirect-SynBench

MoDirect-SynBench is a controlled synthetic benchmark designed to isolate motion direction understanding. It follows a 2×2 factorial design over foreground type and background type: Primitive-on-Syn (P-Syn), Cutout-on-Syn (C-Syn), Primitive-on-Real (P-Real), and Cutout-on-Real (C-Real).

Dataset size and balance. Each domain contains 6,000 video samples, with 1,500 samples per direction (left, right, up, down). This balanced design ensures that performance differences cannot be attributed to class imbalance.

Video generation. Each video contains a single object moving with constant velocity along a linear trajectory. The starting position is sampled uniformly within a valid region, and the velocity is chosen such that (i) the object remains within the frame for the entire sequence and (ii) the total displacement exceeds a minimum threshold. This avoids degenerate cases such as negligible motion or early exit from the frame. The example is shown in Figure 13

Task formulation. We adopt a fourway MCQ format, where the answer options correspond to four signed directions. The order of answer options is randomized for each sample, so that the correct answer letter is not a fixed alias of the motion direction. This design directly tests whether the model can bind the perceived direction to the prompt-specific answer option.

Question format. Each video is paired with a direction question: *"From the viewer's perspective, in which direction is the object moving in this video?"* We use a 4-way multiple-choice setting with candidates {Left, Right, Up, Down}. Candidate order is randomly shuffled to eliminate answer-position bias.

C.1.3. MoDirect-RealBench

MoDirect-RealBench is a real-world motion direction benchmark constructed by curating samples from existing video datasets, including Something-Something-V2 (SSv2), KTH, and TOMATO. For SSv2 and KTH, we reformulate each video into a motion direction multiple-choice question that asks: *"From the viewer's perspective, in which direction is the person moving in this video?"*

Something-Something V2. We select all samples from four action categories that explicitly encode left–right motion: *"Pulling something from left to right," "Pulling something from right to left," "Pushing something from left to right,"* and *"Pushing something from right to left."*. Each sample is reformulated into a binary multiple-choice question, resulting in a chance performance of 50%. This filtering results in 722 samples. We include all available samples from the selected categories, without additional curation or filtering. The example is shown in Figure 14 (a).

KTH. We use samples from the *walking*, *jogging*, and *running* classes. Motion direction (left vs. right) is determined from human skeleton trajectories by measuring the dominant horizontal displacement of key joints. This results in 899 samples. We include all videos from the selected action classes, without manual selection. Each sample is cast as a binary MCQ (left vs. right), yielding a chance accuracy of 50%. The example is shown in Figure 14 (b).

TOMATO. We include 403 samples from the TOMATO (Shangguan et al., 2025) benchmark, which evaluates visual temporal reasoning across three demonstration categories: *human-centric*, *real-world object-centric*, and *simulated* scenarios. Human-centric videos involve human body movements or interactions, such as hand gestures or full-body actions. Object-centric videos focus on the temporal dynamics of real-world objects in natural scenes. Simulated videos depict simplified synthetic environments with abstract moving entities designed to isolate temporal reasoning from appearance complexity. We use the original annotations and convert them into a unified multiple-choice format. Each sample is cast as a 5-way MCQ, yielding a chance accuracy of 20%. Examples are shown in Figure 14 (c) (d).

Evaluation format. All datasets are converted into a multiple-choice format to ensure consistent evaluation. This allows direct comparison with synthetic benchmarks while testing real-world generalization.

C.2. General Video Benchmarks

C.2.1. Standard Video Benchmarks

MVBench. MVBench (Li et al., 2024a) is a multi-modal video understanding benchmark for evaluating temporal comprehension in Multi-modal Large Language Models (MLLMs). It consists of 20 video tasks constructed via a static-to-dynamic transformation and evaluates models using automatically generated multiple-choice questions based on video annotations.

Table 7. MoDirect-Inst statistics.

(a) QA types

QA type	Description	Cnt
direction MCQ (9-way)	8 directions + stationary, 9 choices	17,973
direction MCQ (5-way)	4 directions + stationary, 5 choices	17,903
direction open	open-ended direction answer	13,953
description	motion description	9,885
appearance	object/background appearance description	10,393
location MCQ	3×3 region, 9 choices	6,461
location open	open-ended location answer	6,386
move or not MCQ	yes/no multiple choice	5,128
move or not open	yes/no open-ended answer	5,168
trajectory MCQ	compound direction, 4 choices	4,039
rotation MCQ	CW/CCW multiple choice	1,294
rotation open	CW/CCW open-ended answer	1,417
Total	–	100,000

(b) Motion

Type	Cnt
perturbed linear	29,942
zigzag	20,126
circular	19,830
roundtrip	14,972
static	15,130

(c) Direction

Dir	Cnt
right	8,268
left	8,087
up	8,122
down	8,176
top-right	8,068
top-left	8,067
bottom-right	8,083
bottom-left	8,169

(d) Background

Texture	Cnt
gradient	23,528
stripes	9,840
solid	9,701
checker	9,690
noise	9,407
gaussian noise	9,407
blobs	9,096
speckle	8,953
paper	5,917
vignette	4,406

NExT-QA. NExT-QA (Xiao et al., 2021) is a video question answering benchmark designed to evaluate deeper video understanding beyond description, focusing on causal reasoning, temporal action reasoning, and scene comprehension. It provides both multiple-choice and open-ended QA tasks and reveals that existing models remain weak in causal and temporal reasoning despite strong performance on shallow scene description.

Perception Test. Perception Test (Patraucean et al., 2023) is a multimodal video benchmark designed to evaluate perception and reasoning abilities of pre-trained multimodal models across video, audio, and text modalities. It focuses on skills such as memory, abstraction, physics, and semantics, and provides densely annotated real-world videos with multiple-choice and grounded video question answering for comprehensive evaluation.

EgoSchema. EgoSchema (Mangalam et al., 2023) is a long-form video question answering benchmark designed to evaluate long-term video understanding in vision–language models. It contains over 5,000 human-curated multiple-choice questions derived from Ego4D videos, requiring reasoning over approximately three-minute-long clips and demonstrating the difficulty of long temporal reasoning for current models. In this work, we use the subset of EgoSchema included in our evaluation suite and report multiple-choice accuracy following the standard evaluation protocol.

TGIF-QA. TGIF-QA (Jang et al., 2017) is a video question answering benchmark that extends visual question answering from images to videos, focusing on spatio-temporal reasoning. It introduces three video-specific QA tasks and provides a large-scale dataset designed to evaluate models' ability to reason over temporal dynamics in videos.

C.2.2. Fine-grained video benchmarks

TempCompass. TempCompass (Liu et al., 2024b) is a benchmark designed to evaluate temporal perception abilities of Video LLMs across diverse temporal aspects and task formats. It introduces videos that differ only in specific temporal properties and provides multiple evaluation tasks to measure nuanced temporal understanding beyond single-frame bias.

VinoGround. VinoGround (Zhang et al., 2024a) is a benchmark for evaluating temporal reasoning in short videos using temporal counterfactual video–caption pairs. It consists of 1,000 natural video-caption examples designed to test models' ability to distinguish temporal differences between actions and object transformations.

FAVOR-Bench. FAVOR-Bench (Tu et al., 2025) is a benchmark for evaluating fine-grained motion understanding in Multimodal Large Language Models (MLLMs). It provides manually annotated videos with both multiple-choice and open-ended tasks to assess detailed temporal motion comprehension.

MotionBench. MotionBench (Hong et al., 2025) is a benchmark designed to evaluate fine-grained motion comprehension in video understanding models. It assesses motion-level perception through diverse motion-oriented question types collected from varied real-world video sources.

D. Analysis details

D.1. Linear Probing Details

This appendix specifies the linear probing protocol used in Section 3, Section 4, the cross-backbone direction binding gap analysis in Appendix E.5, and the direction concept vector analysis in Appendix E.7. The goal of this protocol is to measure whether motion-direction information is linearly accessible at different stages of the Video-LLM pipeline, rather than to introduce an additional high-capacity classifier. We therefore use the same lightweight probe design and optimization setting across all stages and backbones.

Probe protocol. For each stage, we freeze the underlying Video-LLM and train a single linear classifier on top of the extracted features. The probe is optimized with cross-entropy loss using standardized inputs, where the mean and standard deviation are computed only from the training split to avoid test-set leakage. Because the probe contains no hidden layers, its performance reflects the linear decodability of the underlying representation rather than the expressive power of the probe itself. Unless otherwise specified, we use this probing protocol for all linear-probing results reported in the main paper and appendix. We use the same hyperparameters for all stages and all backbones, summarized in Table 8.

Table 8. Linear probing hyperparameters used throughout the paper.

Hyperparameter	Value
Probe architecture	Single linear layer + softmax
Loss	Cross-entropy
Optimizer	AdamW
Learning rate	10^{-3}
Weight decay	10^{-2}
Batch size	64
Epochs	50
Train–test split	70/30 stratified

Pipeline stages. We probe four stages along the Video-LLM pipeline. First, we probe the visual encoder output $\mathbf{V} \in \mathbb{R}^{T\times M\times D_v}$. Second, we probe the projector output $\mathbf{F} \in \mathbb{R}^{T\times N\times D}$, which maps visual features into the language-model embedding space. Third, we probe the LLM visual-token hidden state $\mathbf{z}_t^\ell \in \mathbb{R}^D$ at frame t and layer ℓ. Following Section 3.1, $\mathbf{z}_t^\ell$ denotes the spatially pooled hidden state over the visual-token positions corresponding to frame t. Finally, we probe the readout token hidden state $\mathbf{h}^\ell \in \mathbb{R}^D$ at layer ℓ. The readout token is the last token in the prefill and is directly used by the LM head for next-token prediction. In our MCQ template, this position immediately precedes the answer character, making it the relevant representation for analyzing answer-option binding.

D.2. Logit Lens Details

While linear probing verifies whether motion direction is linearly decodable from an internal representation, it does not test whether that representation is already aligned with the LM head for answer selection. To measure this alignment, we use the logit lens at the readout token position.

Formulation. The logit lens (nostalgebraist, 2020) feeds an intermediate hidden state directly to the frozen LM head, without any additional training, to inspect what the model would predict if decoding stopped at that layer. For layer $\ell \in \{1, \ldots, L\}$, we compute the logit-lens letter prediction at the readout token position as

$$\hat{y}^{\ell}_{\text{lens}} = \arg\max_{c \in \{A,B,C,D\}} \left(\mathbf{W}_u \cdot \text{Norm}(\mathbf{h}^{\ell}) \right)_{t_c}, \tag{6}$$

where $\mathbf{h}^{\ell} \in \mathbb{R}^D$ is the readout token hidden state at layer ℓ, $\text{Norm}(\cdot)$ is the final pre-unembedding normalization layer of the LLM, $\mathbf{W}_u \in \mathbb{R}^{V \times D}$ is the frozen LM head, and t_c is the vocabulary index of answer-letter token c.

Protocol. Unless otherwise specified, we use the same prompt format, video preprocessing, answer-token candidates, and held-out evaluation split as in the corresponding MCQ evaluation. We apply the frozen final normalization layer and LM head to each intermediate readout-token hidden state and evaluate whether the highest-scoring answer-letter token matches the ground-truth option. No additional parameters are trained for this analysis.

Interpretation. Unlike a learned linear probe, the logit lens evaluates whether the model's own LM head can already read out the correct answer from a given layer. Thus, a gap between linear-probe accuracy and logit-lens accuracy indicates that motion-direction information is present in the representation but is not yet aligned with the answer-selection space.

E. Additional Analysis

E.1. Video LLMs Instruction-tuning Dataset Analysis

This section provides details of the instruction-data analysis summarized in Section 3.2. We describe the direction supervision estimation pipeline, its validation via human annotation, and the resulting dataset statistics.

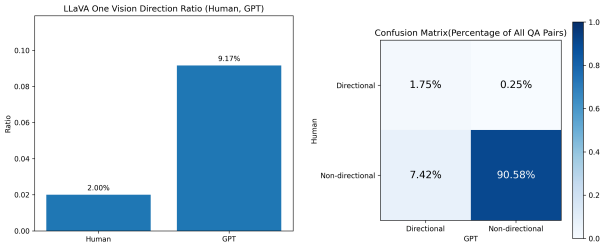
Table 9. **Motion direction supervision is scarce in video instruction-tuning data.** Proportion of QA pairs identified by two stages of the Direction Supervision Estimation pipeline. **Keyword Matched** reports the number of QA pairs that contain explicit directional keywords and the corresponding total number of QA pairs in each dataset. **Semantically Direction-dependent QA** reports the subset of keyword-matched QA pairs that require motion direction understanding after semantic filtering, where the total corresponds to the size of the keyword-matched subset. **Final Ratio** denotes the proportion of semantically direction-dependent QA pairs relative to the total number of QA pairs in the dataset.

Dataset	Keyword Matched		Semantically Direction-dependent QA		Final Ratio
	Matched	Total	Directional QA	Matched	
VideoChat2-IT (Li et al., 2024a)	143,759	957,999	18,134	143,759	1.89%
VideoChatGPT100K (Maaz et al., 2024)	23,156	99,814	989	23,156	0.99%
LLaVA-OneVision (video) (Li et al., 2025a)	47,240	487,632	3,766	47,240	0.77%
LLaVA-Video-178K (Zhang et al., 2025c)	802,700	5,699,294	51,776	802,700	0.91%

E.1.1. Motion Direction Supervision Estimation Pipeline

Keyword pre-filtering for motion direction. The instruction data contains a diverse set of QA pairs. To efficiently identify candidate samples, we first apply a lightweight pre-filtering step based on keyword matching. We define a keyword set $\mathscr{K} = \{\text{left, right, up, down}\}$, which captures the primary horizontal and vertical directions. A QA pair is retained if any keyword in $\mathscr{K}$ appears as a case-insensitive whole-word match in either the question or the ground-truth answer. This step intentionally retains a broad set of candidate QA pairs, including cases where directional terms appear without requiring motion direction reasoning. For example, words such as "right" may refer to correctness rather than spatial direction, and phrases like "close up" may include directional terms without implying object motion. These cases cannot be reliably disambiguated by keyword matching alone and are further filtered in the subsequent semantic classification stage.

Semantic filtering for motion direction. We apply a semantic filtering step to determine whether a QA pair truly requires motion direction understanding. We formulate this as a binary classification problem, where each QA pair is labeled as *directional* or *non-directional*. The classifier takes both the question and the ground-truth answer as input, and predicts whether motion direction reasoning is required to answer the question correctly. We implement this classifier using a GPT-4o model via API. This step filters out semantically irrelevant cases that pass the keyword pre-filtering, yielding a refined subset of direction-dependent QA pairs.



(a) *(b)*

Figure 7. **LLM-based semantic classification tends to over-predict direction.** (a) Comparison of overall direction ratios. While human annotations show that only a small fraction of QA pairs require direction understanding, the LLM predicts direction at a substantially higher rate, further supporting the over-prediction tendency. (b) Row-normalized confusion matrix comparing LLM predictions with human annotations. The model more often labels non-direction QA pairs as direction than it misses true direction cases, indicating a tendency to over-predict direction.

E.1.2. Reliability of Semantic Filtering

Human verification. Since GPT-based semantic filtering may cause classification errors, we validate its reliability against human annotations. We randomly sample 5,000 QA pairs from the LLaVA-OneVision instruction dataset(Li et al., 2025a) and label each pair as *directional* or *non-directional*. Annotators follow the same labeling criteria used for the GPT-based classifier and are not exposed to its predictions. The resulting annotations are used to analyze agreement and error patterns between GPT and human judgments.

Error analysis. Using these annotations, we analyze the agreement between GPT predictions and human labels, as shown in Figure 7. Figure 7 (a) shows that GPT overestimates the proportion of direction-dependent QA pairs. Figure 7 (b) further shows that false positives primarily drive this discrepancy is primarily driven by false positives. Specifically, the classifier incorrectly labels non-directional QA pairs as directional, while missing very few true directional cases. This results in high recall (87.5%) but low precision (19.1%), indicating that the classifier rarely misses truly directional QA pairs while substantially over-predicting directional ones. Since the analysis is based on sampled data, we account for sampling variability by computing a 95% confidence interval ($9.17\% \pm 2.8\%$). Even under this uncertainty, the estimated proportion remains substantially higher than the human-annotated rate, indicating consistent overestimation. Because the classifier tends to include non-directional cases rather than exclude true directional ones, we interpret this estimate as an upper bound on the true amount of motion direction supervision.

E.1.3. Results

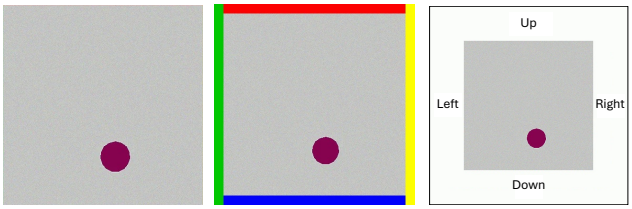
We quantify the presence of motion direction supervision in video instruction-tuning datasets using the proposed two-stage pipeline. As shown in Table 9, keyword-based pre-filtering retains a substantial portion of QA pairs (23–29%), but this proportion drops sharply after semantic filtering. Across datasets, only 0.77–1.89% of QA pairs require motion direction understanding, remaining consistently around 1% despite differences in dataset size and composition. Given that GPT-based semantic filtering tends to overestimate due to false positives (Appendix E.1.2), these values should be interpreted as upper bounds. Even under this conservative interpretation, motion direction supervision remains limited across all datasets.

E.2. Input-side Scaffolds: Design and Full Results

In this section, we provide a detailed description of of the prompting design and full results underlying this finding. We study input-side scaffolds from two perspectives. On the visual side, we augment video frames with explicit directional cues. On the text side, we modify the prompt to guide step-by-step reasoning about object motion. We consider multiple visual and text scaffolds and evaluate all visual–text combinations across synthetic and real-world benchmarks. We describe the visual and text prompting strategies and then report the full results for all scaffold configurations.

E.2.1. Visual Prompting

Visual prompting introduces explicit directional cues into the video frames, providing additional spatial signals that can guide direction inference (Chen et al., 2024b; Fu et al., 2025b). We consider three visual conditions: *(i) Plain* uses the original video without any additional visual cues. *(ii) Color Edges* assign distinct colors to the four image borders, allowing the model to relate the object's position to the corresponding boundary cues. *(iii) Text Edges* explicitly annotate each border with directional words (e.g., "Up", "Down", "Left", "Right"), providing a more explicit association between position and direction. Examples of all conditions are shown in Figure 8



(a) Plain *(b)* Colored Edge *(c)* Text Edge

Figure 8. **Visual prompting examples.** We compare the original input with two visual cue variants designed to make directional information more explicit. (a) *Plain* shows the unmodified video frame without any additional visual cue. (b) *Colored Edge* marks the four image borders with distinct colors. (c) *Text Edge* annotates the borders with directional words.

E.2.2. Text Prompt

Text prompting modifies the input question to elicit structured reasoning about motion direction. Rather than determining direction from a single query, these prompts guide the model to explicitly reason over object positions and their temporal changes.

We consider three variants: *(i) Default prompting* asks the direction question directly without additional guidance. *(ii) Temporal prompting* decomposes the task into intermediate steps, asking the model to describe the object and its positions at the beginning and end of the video before determining direction. *(iii) Grid-based prompting* follows the same step-by-step structure as temporal prompting, but additionally introduces an explicit coordinate system over the image, requiring the model to estimate start and end positions within the grid and determine direction from coordinate change. Examples of all conditions are shown in Figure 15, Figure 16, Figure 17

E.2.3. Full Results.

Table 10. **Prompting does not reliably recover motion direction reasoning.** Accuracy under combinations of visual and text prompting.

Visual prompt	Text prompt	P-Syn	SSv2(Goyal et al., 2017)
Random chance		25.0	50.0
Plain	Default	31.5	54.2
	Temporal	29.0	**57.1**
	Grid	28.2	53.6
Colored edge	Default	25.5	48.6
	Temporal	29.4	52.9
	Grid	23.0	49.2
Text edge	Default	32.0	49.9
	Temporal	**34.7**	54.2
	Grid	29.1	50.1

Table 10 reports accuracy under all scaffold combinations on Primitive-on-Syn and SSv2 (Goyal et al., 2017). On Primitive-on-Syn, the best-performing configuration, *Temporal + Text edge*, reaches 34.7%, only a modest improvement over the raw baseline (31.5%).

On SSv2, the best result is achieved by *Temporal prompting* at 57.1%, but this gain is not consistently preserved across other combinations. Visual prompting alone (e.g., Colored edge: 48.6%, Text edge: 49.9%) generally underperforms the raw baseline (54.2%), and combining visual and text prompts does not yield reliable improvements.

Overall, the results show that neither visual cues nor structured text prompting consistently recovers motion direction reasoning, and in many cases, additional prompting introduces noise that degrades performance.

E.3. Visual Perception: Additional Results and Controls

E.3.1. Setup

We provide control experiments supporting the main claim of Section 3.3: the vision encoder retains strong motion-direction information that survives projection into the LLM embedding space. The frozen SigLIP (Zhai et al., 2023) encoder of LLaVA-Video (Zhang et al., 2025c) produces patch-token features $\mathbf{V} \in \mathbb{R}^{T\times M\times D_v}$ (T=8, M=729, D_v=1152); we mean-pool over the spatial axis to obtain per-frame descriptors $\bar{\mathbf{V}}_t \in \mathbb{R}^{D_v}$ and train a linear probe on top, following the protocol in Appendix D.1. Unless otherwise stated, we anchor the analysis on Primitive-on-Syn, the cleanest synthetic domain, and Cutout-on-Real, the most visually complex synthetic domain, which span the two endpoints of the MoDirect-SynBench complexity axis.

Table 11. **Mean-pooled visual features preserve object position.** Coefficient of determination R^2 for closed-form ridge regression predicting per-frame object center coordinates (x, y) from features at two stages: pre-projector (SigLip output, $\mathbf{V}$) and post-projector (after mm_projector, $\mathbf{F}$). Position is recovered with R^2 between 0.80 and 0.99 across both domains and stages, demonstrating that mean-pooling and projection do not destroy spatial localization, and validating the linear probing protocol used throughout the paper.

Stage	Domain	$R^2(x)$	$R^2(y)$
Pre-projector ($\mathbf{V}$)	Primitive-on-Syn	0.985	0.994
	Cutout-on-Real	0.806	0.871
Post-projector ($\mathbf{F}$)	Primitive-on-Syn	0.968	0.987
	Cutout-on-Real	0.839	0.885

E.3.2. Mean-Pooling Preserves Spatial Position

Our main analysis operates on per-frame descriptors $\bar{\mathbf{V}}_t$ rather than the original patch-level features $\mathbf{V}$. We therefore first verify that averaging the M=729 patch tokens does not discard spatial information. We probe the per-frame object center $(x, y) \in \mathbb{R}^2$ from $\bar{\mathbf{V}}_t$ via closed-form ridge regression (α=1.0). As shown in Table 11, position is recovered with $R^2 \in [0.80, 0.99]$ across Primitive-on-Syn and Cutout-on-Real at both pre- and post-projector stages. The per-frame descriptor therefore retains the spatial informa-

tion that downstream probes read, and the 99.78% accuracy reported in Section 3.3 reflects a property of the SigLIP encoder rather than an artifact of patch pooling or probe overfitting.

E.3.3. The Direction Probe Reads Inter-Frame Change

Table 12. **Direction probing requires the temporal axis.** Four-way direction probe accuracy (%) on the frozen vision-encoder output across five feature constructions, evaluated on Primitive-on-Syn (the cleanest synthetic domain) and Cutout-on-Real (the most visually complex synthetic domain). Constructions that destroy temporal order (*single*, *tmean*) collapse near chance, while every construction that retains it (*stack*, *delta*, *delta7*) recovers direction at high accuracy, validating that the (T, D_v)-shape probe in Section 3.3 relies on temporal evolution rather than per-frame appearance. Chance accuracy is 25.0%.

Construction	Primitive-on-Syn	Cutout-on-Real
Random chance	25.0	25.0
Single ($\bar{\mathbf{V}}_{T/2}$)	41.6	37.3
T-mean ($\frac{1}{T}\sum_t \bar{\mathbf{V}}_t$)	30.1	30.7
Stack ($[\bar{\mathbf{V}}_0;\ldots;\bar{\mathbf{V}}_{T-1}]$)	99.1	81.2
Delta ($\bar{\mathbf{V}}_{T-1}-\bar{\mathbf{V}}_0$)	**99.8**	**88.8**
Concat. Delta (consecutive differences)	98.9	72.7

The probe in Section 3.3 achieves near-perfect accuracy from the temporally-stacked per-frame descriptors $[\bar{\mathbf{V}}_0;\ldots;\bar{\mathbf{V}}_{T-1}]$. However, this result alone does not show whether the probe uses true motion dynamics or static appearance cues correlated with direction. To distinguish these possibilities, we remove temporal order while preserving the same visual content. Specifically, we compare the ordered *stack* with two order-free variants: *single*, which uses the middle-frame descriptor $\bar{\mathbf{V}}_{T/2}$, and *T-mean*, which uses the temporal average $\frac{1}{T}\sum_t \bar{\mathbf{V}}_t$. As reported in Table 12, the order-free variants remain near chance (30–42%), whereas the ordered stack reaches 99.1% on Primitive-on-Syn and 81.2% on Cutout-on-Real. Direction decoding therefore relies on inter-frame change rather than per-frame appearance. We additionally report two temporal-difference features, *delta* and *delta7*, as temporal controls; their representational properties and role in the MVP loss are analyzed separately in Appendix E.9.

E.3.4. The Direction Probe Is Not an Appearance Shortcut

A remaining concern is whether the encoder-side 99.78% direction signal in Section 3.3 reflects an appearance shortcut, where specific objects co-occur with specific motion directions. Table 13 repeats the construction sweep with object identity as the target (30-way on Primitive-on-Syn, 26-way on Cutout-on-Real); identity recovery varies steeply across constructions and does not track direction-probe accuracy. The encoder therefore encodes object identity and motion direction in separable subspaces, and the 99.78% direction signal cannot be a re-reading of the identity signal. The full direction–identity dissociation at the projector output, its extension to all four MoDirect-SynBench domains, and its design implications for the MVP supervision target of `DeltaDirect` are analyzed in Appendix E.9.

Table 13. **The *delta* construction destroys appearance content.** Object-identity probe accuracy (%) on the same five feature constructions, with the same probe and protocol as Table 12. The identity probe classifies among 30 shape-color combinations on Primitive-on-Syn and among 26 COCO classes on Cutout-on-Real. The *Delta* construction collapses identity recovery while preserving direction (Table 12), showing that the direction probe is not exploiting an appearance shortcut.

Construction	Primitive-on-Syn	Cutout-on-Real
Random chance	3.3	3.8
Single ($\bar{\mathbf{V}}_{T/2}$)	89.8	78.2
T-mean ($\frac{1}{T}\sum_t \bar{\mathbf{V}}_t$)	93.2	79.2
Stack ($[\bar{\mathbf{V}}_0;\ldots;\bar{\mathbf{V}}_{T-1}]$)	93.4	79.4
Delta ($\bar{\mathbf{V}}_{T-1}-\bar{\mathbf{V}}_0$)	**30.2**	**11.4**
Delta7 (consecutive differences)	38.8	14.9

Table 14. **Direction survives projection; identity remains suppressed.** Direction-probe and identity-probe accuracy (%) on the *delta* construction at two stages of the Video-LLM pipeline: the pre-projector encoder output (**V**) and the post-projector LLM input (**F**). Direction information is retained almost entirely after projection, while identity information is further suppressed. This rules out the projector as the source of directional motion blindness.

	Direction probe		Identity probe	
Stage	Primitive-on-Syn	Cutout-on-Real	Primitive-on-Syn (30-way)	Cutout-on-Real (26-way)
Pre-projector (**V**)	99.8	88.8	30.2	11.4
Post-projector (**F**)	99.1	86.7	21.8	9.9
Δ (pre → post)	−0.7	−2.1	−8.4	−1.5

E.3.5. Direction Survives Projection into the LLM Embedding Space

The main analysis in Section 3.3 probes the frozen encoder output **V**. We additionally probe the projector output **F** to test whether projection into the LLM embedding space erases the signal. As shown in Table 14, the direction probe drops by only 0.7 and 2.1 points on Primitive-on-Syn and Cutout-on-Real after projection, while identity drops further on both domains. The projector therefore passes direction to the LLM without measurable degradation, so the failure observed in Section 3.4 cannot be attributed to representation loss at this stage.

E.3.6. Direction Degrades Smoothly with Visual Complexity

We extend the direction probe from the Primitive-on-Syn/Cutout-on-Real anchors to the intermediate domains Cutout-on-Syn and Primitive-on-Real, separating foreground complexity (Primitive vs. Cutout) from background complexity (Solid vs. Scene). Table 15 shows that direction accuracy degrades smoothly with visual complexity from Primitive-on-Syn to Cutout-on-Syn, Primitive-on-Real,

Table 15. **Direction-probe accuracy and direction-discriminative dimension counts across the four MoDirect-SynBench domains.** On the encoder output **V**, four-way direction probe accuracy under the *delta* construction degrades smoothly with visual complexity, remaining far above the 25% chance baseline on every domain. The number of direction-discriminative dimensions ($F>5$ from per-dimension ANOVA) follows the same trend. Background complexity (Solid→Scene) is the dominant degradation axis (−12.7 points from Primitive-on-Syn to Primitive-on-Real) compared with foreground complexity (Primitive→Cutout, −3.1 points from Primitive-on-Syn to Cutout-on-Syn).

Domain	Direction linear probing acc.	R_x^2	R_y^2	F_{top50}	$\#(F>5)$
Primitive-on-Syn	99.8	0.99	0.99	309.4	1084
Cutout-on-Syn	94.3	0.85	0.95	109.6	981
Primitive-on-Real	97.1	0.88	0.93	77.6	724
Cutout-on-Real	88.8	0.85	0.90	69.4	610

Cutout-on-Real, remaining far above the 25% chance baseline even on the most complex domain. Decomposing by factor, switching from solid to scene backgrounds (Primitive-on-Syn→Primitive-on-Real) costs 2.7 points, while switching from primitive to cutout foregrounds (Primitive-on-Syn→Cutout-on-Syn) costs only 5.5, identifying background complexity as the dominant degradation axis.

E.3.7. Position Information Migrates to Low-Variance Dimensions Under Visual Complexity

While direction remains decodable across all four domains, the *geometry* of preservation differs. Table 16 reports position decodability from the top-k principal components of **V** alongside the variance they explain. On Primitive-on-Syn, the top-50 PCs (92.6% of variance) recover position with R^2=0.86; on Cutout-on-Real, the same top-50 PCs explain a comparable 91.7% of variance but yield R^2=0.03, with recovery requiring the top-500 PCs. On the most complex domain, the motion signal is therefore not erased but *redistributed* into the low-variance tail—directions that downstream layers, biased toward high-variance features, are unlikely to prioritize. This upstream redistribution at the visual interface is the encoder-side counterpart of the magnitude deficit observed at the LLM readout in Section 4.4.

E.3.8. Dimension-Removal Ablations

To localize where position is encoded in **V**, we ask whether removing high-variance components destroys position information—once in the raw-dimension basis and once in the PCA basis. *(i) Pos-corr removal.* Dropping the top-10 position-correlated dimensions—only 0.9% of total variance—reduces R_x^2 from 0.81 to 0.63, confirming that position is concentrated in a small set of low-variance dimensions. *(ii) Obj-disc removal* (reverse control). Removing the top-500 object-discriminative dimensions (41.6% of variance) leaves R_x^2 nearly intact (0.81→0.78), showing that position and object information occupy largely disjoint dimension sets. *(iii) High-variance PC removal.* The same question in the PCA basis yields the same answer: keeping the top-50 PCs (91.7% of variance) still yields R_x^2=0.77, yet additionally removing PCs 51–100 (a further 4%) collapses the probe to $R_x^2 \approx 0$, directly localizing position to the low-variance band of PCs 51–100. All three ablations converge: position is encoded in a small, low-variance subspace that is largely orthogonal to the dominant semantic axes.

E.4. Motion Direction Binding Gap Beyond MCQ Format

In this section, we verify that the direction binding gap is not an artifact of the letter-MCQ format. We evaluate vanilla LLaVA-Video (Zhang et al., 2025c) on Primitive-on-Syn under two additional answer formats that preserve the direction question but change the required output: *direction-word MCQ*, where the four options are the direction words {left, right, up, down} rather than the letters A/B/C/D, and *open-ended generation*, where the model generates a free-text answer that we score with GPT-4o (Hurst et al., 2024). For each format, we measure the final-readout direction probe accuracy and the corresponding task accuracy.

The binding gap persists across answer formats. Table 18 reports the results. The final-readout direction probe stays above 90% in all three formats, indicating that motion direction is linearly accessible regardless of the required output format. Yet the task accuracy remains low across the board, ranging from 19.7% for open-ended generation to 27.8% for direction-word MCQ. The direction binding gap therefore exceeds 60 pp in every format we tested.

The binding gap is not an MCQ artifact. The persistence of the gap under both direction-word MCQ and open-ended generation rules out the letter-MCQ format as the source of directional motion blindness. The bottleneck lies in converting the linearly accessible direction signal into the correct verbal answer, regardless of whether that answer is a letter, a word, or a free-text response.

E.5. Motion Direction Binding Gap Across Video-LLMs

Setup. We test whether the direction binding gap observed in LLaVA-Video (Section 3.4) generalizes to other Video-LLMs. For each model, we measure two quantities on the Primitive-on-Syn domain of MoDirect-SynBench: (i) direction probing accuracy at the final readout token, fol-

Table 16. **Position decodability shifts to low-variance directions under visual complexity.** We decode the horizontal position from the top-k principal components of the encoder output $\mathbf{V}$ across the four MODIRECT-SYNBENCH domains. On Primitive-on-Syn, the top-50 PCs already recover position well (R_x^2=0.86), while on Cutout-on-Real the same top-50 PCs yield near-zero decodability (R_x^2=0.03) despite explaining comparable variance. Thus, visual complexity pushes position information from dominant variance directions to the low-variance tail.

Domain	Position Decodability R_x^2			Explained Variance
	Top-50	Top-100	Top-500	Top-50 PCs
Primitive-on-Syn	0.86	0.93	0.98	0.93
Cutout-on-Syn	0.08	0.27	0.69	0.65
Primitive-on-Real	0.01	0.73	0.82	0.98
Cutout-on-Real	0.03	0.70	0.78	0.92

Table 17. **Dimension-removal ablations on the encoder output V of Cutout-on-Real.** Pos-corr removal *(i)* ranks raw dimensions by position correlation and removes the top-k; Obj-disc removal *(ii)* does the same by object-class discriminability as a reverse control; High-variance PC removal *(iii)* removes the top-k principal components ranked by variance explained. Removing a small fraction of position-correlated dimensions (k=10, 0.9% of variance) sharply reduces R_x^2, while removing far more object-discriminative dimensions (k=500, 41.6%) leaves R_x^2 nearly intact, and the dominant PCs (k=50, 91.7%) carry little position information.

Ablation	k	Var. removed	R_x^2
Full (no ablation)	0	0.000	0.806
(i) Pos-corr removal	10	0.009	0.630
	100	0.104	0.598
	500	0.449	0.412
(ii) Obj-disc removal	100	0.078	0.800
	500	0.416	0.777
(iii) Top-PC removal	50	0.917	0.767
	100	0.958	−0.004

lowing Appendix D.1, and (ii) MCQ accuracy from greedy generation, parsed as the first emitted letter among `{A, B, C, D}`. We define the binding gap as the difference between the two, following Section 3.4. We evaluate every model on 6,000 balanced samples (1,500 per direction, four directions).

Models. We evaluate eight Video-LLMs spanning different vision encoders, projectors, LLM families (Qwen2 / Qwen2.5 / Qwen3 / Vicuna), and parameter scales (2B-7B). We list the full model set with citations in Table 19.

The binding gap is consistent across Video-LLMs. Table 19 reports direction probing and MCQ accuracy on the eight evaluated Video-LLMs. Direction probing is high in every model (75.3 %-97.8 %), confirming that motion direction is linearly decodable from the readout token across architectures. MCQ accuracy, however, stays near the 25 % chance baseline in six of the eight models; only Qwen2.5-VL (45.1 %) and Qwen3-VL (66.6 %) achieve meaningful direction MCQ out of the box. The resulting binding gap exceeds 30 percentage points in every Video-LLM we evaluate. Even Qwen3-VL — the most MCQ-aligned vanilla model — exhibits a 31.2 pp gap (probe 97.8 % vs. MCQ 66.6 %): the binding gap persists even when downstream MCQ capability is high. The gap is therefore not specific to a particular vision encoder, projector, or LLM family, but reflects a shared failure mode across the Video-LLMs we evaluate.

Table 18. **The direction binding gap persists beyond the MCQ format.** Final-readout direction probe accuracy and task accuracy on vanilla LLaVA-Video, evaluated on Primitive-on-Syn under three answer formats: letter MCQ (default), direction-word MCQ, and open-ended generation. The final-readout direction probe stays above 90% in all three formats, while task accuracy remains low, leaving a binding gap of more than 60 pp in every format. Open-ended generation is scored with GPT-4o.

Answer format	Probe (%)	Acc. (%)	Gap (pp)
Letter MCQ	95.3	27.6	67.7
Direction-word MCQ	92.3	27.8	64.5
Open-ended generation	93.3	19.7	73.6

Table 19. **The direction binding gap is universal across Video-LLMs.** Direction probing accuracy and MCQ accuracy at the final readout state, evaluated on Primitive-on-Syn in MODIRECT-SYNBENCH (6,000 balanced samples). Direction is linearly decodable in every backbone, yet MCQ accuracy stays near the 25 % chance baseline in seven of the nine models. Even Qwen3-VL, the strongest MCQ baseline, retains a 31.2 pp gap.

Method	Direction Probe (%)	MCQ Acc. (%)	Gap (pp)
LLaVA-Video-7B (Qwen2) (Zhang et al., 2025c)	95.3	27.6	67.8
mPLUG-Owl-Video (Ye et al., 2025)	90.5	25.6	64.9
VideoLLaMA3-2B (Qwen2) (Zhang et al., 2025a)	93.6	30.9	62.7
LLaVA-OneVision-7B (Qwen2) (Li et al., 2025a)	90.8	28.3	62.5
InternVL2.5-4B (Qwen2.5) (Chen et al., 2024b)	91.1	31.9	59.2
Qwen2.5-VL-7B (Bai et al., 2025b)	95.8	45.1	50.7
LLaVA-NeXT-Video-7B (Vicuna) (Zhang et al., 2024b)	75.3	25.8	49.5
Video-LLaVA-7B (Vicuna) (Lin et al., 2024)	77.1	28.1	49.0
Qwen3-VL-4B (Bai et al., 2025a)	**97.8**	**66.6**	**31.2**

E.6. Out-of-Domain: The Binding Gap Reopens

We train on Primitive-on-Syn with MODIRECT-INST and probe the final readout token at the last layer with a direction linear probe. The probe and the LM head share the same token, so any divergence between the two reflects a binding failure rather than a representation failure. We evaluate on Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real as OOD conditions (Table 20).

Table 20. **`DeltaDirect` narrows the direction binding gap across visual complexity.** Last-layer measurements on the four MoDirect-SynBench conditions (%). The *direction binding gap* (direction linear probe − MCQ accuracy) shrinks on every OOD condition under `DeltaDirect`.

Condition	Direction	MCQ	Binding gap
Trained on MoDirect-Inst			
Primitive-on-Syn (Source domain)	99.8	99.5	0.3
Cutout-on-Syn	88.8	80.7	7.3
Primitive-on-Real	84.8	74.7	10.1
Cutout-on-Real	**72.6**	**60.5**	**12.1**
`DeltaDirect`			
Primitive-on-Syn (Source domain)	99.8	99.7	0.1
Cutout-on-Syn	87.8	84.9	2.9
Primitive-on-Real	90.0	85.2	4.8
Cutout-on-Real	**75.4**	**71.7**	**3.7**

Under MoDirect-Inst alone, the direction binding gap grows with visual complexity. The gap stays near zero on the source domain (0.3%), then widens to 7.3% on Cutout-on-Syn, 10.1% on Primitive-on-Real, and 12.1% on Cutout-on-Real. The direction probe still recovers the motion direction on every OOD condition, so the model encodes the answer but fails to route it to the MCQ letter.

`DeltaDirect` narrows the gap on every OOD condition. The gap drops to 2.9% on Cutout-on-Syn, 4.8% on Primitive-on-Real, and 3.7% on Cutout-on-Real, while the direction probe accuracy stays comparable or improves. The MCQ accuracy rises by 4.2%, 10.5%, and 11.2% on Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real, respectively. `DeltaDirect` closes the binding gap rather than enhancing the underlying direction representation.

E.7. Direction Concept Vector Analysis

In Section 4.4, we apply difference-in-means (Marks and Tegmark, 2024) to extract motion direction concept vectors and analyze their orientation and magnitude across domains. This appendix examines whether the extracted vectors capture motion direction rather than letter, identity, or other confounded factors, through six complementary measurements: existence (Appendix E.7.1), specificity (Appendix E.7.2), antipodal mirror structure (Appendix E.7.3), decodability (Appendix E.7.4), cross-domain consistency (Appendix E.7.5), and causality (Appendix E.7.6).

Background. Difference-in-means is a standard technique for extracting concept directions from LM hidden states. Marks and Tegmark (2024) use it to recover truth directions in factual statements, and Arditi et al. (2024) apply it to isolate and ablate refusal directions in instruction-tuned LMs. We adopt the same construction to obtain motion direction concept vectors from Video-LLM hidden states.

Setup recap. We analyze three models on MoDirect-SynBench: the vanilla LLaVA-Video (Zhang et al., 2025c), the instruction-tuned model from Section 4.2, and `DeltaDirect` from Section 5. For each domain $A \in \{\text{P-Syn, C-Syn, P-Real, C-Real}\}$, direction $d \in \{\text{left, right, up, down}\}$, and decoder layer $\ell \in \{1,\dots,28\}$, we collect the readout token state $\mathbf{h}_A^\ell \in \mathbb{R}^D$. Following the difference-in-means construction widely used to isolate concept-specific directions in language model representations (Marks and Tegmark, 2024; Arditi et al., 2024; Li et al., 2023; Tigges et al., 2024; Rimsky et al., 2024), the direction concept vector at layer ℓ is

$$\mathbf{v}_{d,A}^\ell = \mathbb{E}\Big[\mathbf{h}^\ell \mid y = d, A\Big] - \mathbb{E}\Big[\mathbf{h}^\ell \mid A\Big],$$

where the expectation is over the held-out MoDirect-SynBench samples (1.5K per direction per domain, 6K per domain in total). Because MoDirect-SynBench is class-balanced, all visual factors orthogonal to direction cancel out in the subtraction. We decompose each vector into its unit orientation $\hat{\mathbf{v}}_{d,A}^\ell = \mathbf{v}_{d,A}^\ell / \|\mathbf{v}_{d,A}^\ell\|$ and magnitude $\|\mathbf{v}_{d,A}^\ell\|$. We similarly define *letter* concept vectors for $l \in \{A,B,C,D\}$ (gold MCQ answer letter, depends on per-sample candidate shuffle) and *identity* concept vectors for the foreground/background identity classes (shape/color for Primitive-on-Syn, object class for Cutout-on-Syn, shape/place for Primitive-on-Real, object class/place for Cutout-on-Real). Letter and identity vectors serve as control concepts to test direction-specificity.

E.7.1. Existence: Direction Magnitude Dominates Control Concepts

We first check whether the vectors (i) exist in the readout state with non-trivial magnitude and (ii) exhibit larger magnitudes than competing concept axes. Table 21 reports $\|\mathbf{v}_{d,A}^\ell\|$ averaged over the four directions, alongside the cor-

Table 21. **Direction concept vector magnitude dominates control concepts.** We report $\|\mathbf{v}^\ell\|$ averaged over the four directions for the direction axis at six representative layers, and at $\ell = 21$ for the letter and identity control axes (averaged over their respective classes). After instruction tuning, the direction concept vector is amplified by roughly an order of magnitude relative to the vanilla model and dominates both control concepts at $\ell = 21$. `DeltaDirect` preserves this amplification while increasing OOD magnitudes (e.g., Primitive-on-Real: $18.47 \rightarrow 21.37$).

	Direction $\|\mathbf{v}_d^\ell\|$						**Controls at** ℓ=21	
Domain	L10	L17	L20	L21	L23	L27	Letter	Identity
Vanilla								
Primitive-on-Syn	0.22	0.36	1.59	2.57	3.56	5.24	0.38	3.36
Cutout-on-Syn	0.24	0.31	1.07	1.62	2.22	3.43	0.41	5.38
Primitive-on-Real	0.10	0.17	0.84	1.49	1.96	2.89	0.50	7.49
Cutout-on-Real	0.12	0.24	0.87	1.26	1.72	2.67	0.43	6.36
Instruction-tuned								
Primitive-on-Syn	0.21	1.13	14.22	28.92	34.57	47.48	10.23	3.01
Cutout-on-Syn	0.22	0.77	9.71	20.49	26.18	35.74	7.25	5.23
Primitive-on-Real	0.12	0.61	8.21	18.47	23.96	32.00	5.72	7.59
Cutout-on-Real	0.12	0.46	6.11	14.15	18.48	24.38	4.34	6.45
`DeltaDirect`								
Primitive-on-Syn	0.25	1.18	13.92	28.29	34.44	46.13	10.02	3.14
Cutout-on-Syn	0.24	0.80	9.68	20.85	26.88	35.19	7.29	5.42
Primitive-on-Real	0.12	0.66	9.21	21.37	27.81	35.91	6.94	7.11
Cutout-on-Real	0.13	0.51	7.19	17.03	22.48	28.63	5.44	6.36

responding letter and identity magnitudes at $\ell = 21$. After instruction tuning, direction magnitude grows over $100\times$ from $\ell = 10$ to $\ell = 21$ on Primitive-on-Syn ($0.21 \rightarrow 28.92$), reaching values 2-10$\times$ larger than the letter and identity controls in the same layer. This late-layer amplification is consistent with the readout-consolidation pattern observed via the logit lens in language models (nostalgebraist, 2020). The vanilla model exhibits the same monotone growth pattern but at a much smaller scale (peak 2.57 at $\ell = 21$ on Primitive-on-Syn), indicating that fine-tuning amplifies the direction signal at the readout state by roughly an order of magnitude. `DeltaDirect` preserves a comparable amplification pattern while noticeably lifting OOD magnitudes (e.g., Primitive-on-Real at $\ell = 21$: $18.47 \rightarrow 21.37$), consistent with the magnitude-deficit closure reported in Section 5.

E.7.2. Specificity: Direction Is Orthogonal to Letter and Identity

Large magnitude alone does not imply direction-specificity. The vector could still entangle with the answer letter or with foreground/background identity. For each layer, we compute the average absolute cosine $|\cos(\hat{\mathbf{v}}^\ell_{d,A}, \hat{\mathbf{v}}^\ell_{c,A})|$ between the direction axis and each control concept axis c, averaged over all class pairs. Table 22 shows that, in the readout-consolidation layers $\ell \in [20, 23]$, the direction-letter cosine stays at or below $\sim$0.17 across all (model, task) settings on the fine-tuned models, and the direction-identity cosine stays at or below $\sim$0.21. The vanilla model exhibits slightly higher entanglement, particularly with identity in late layers (up to 0.40 on Primitive-on-Syn at $\ell = 28$), reflecting the absence of a clean answer-binding pathway (Orgad et al., 2025; Park et al., 2025; Sun et al., 2025). After fine-tuning, direction is distinct from both control axes in the layers where it is largest, suggesting that the amplified direction signal is not a simple leakage of the letter or identity axes.

E.7.3. Geometric Structure: Opposite Directions Form Antipodal Axes

If motion direction is encoded as a coherent geometric concept, we would expect opposite physical directions to lie on opposite sides of a common axis: $\hat{\mathbf{v}}_{\text{left}} \approx -\hat{\mathbf{v}}_{\text{right}}$ and $\hat{\mathbf{v}}_{\text{up}} \approx -\hat{\mathbf{v}}_{\text{down}}$. Such linear, antipodal structure has been observed for a range of high-level concepts in language model representations (Marks and Tegmark, 2024; Gurnee and Tegmark, 2024). Table 23 reports the signed cosine for both pairs. After instruction tuning, both pairs reach -0.84 to -0.92 at $\ell = 21$ across all four domains, while the vanilla model stays near 0 on the OOD domains for left-right (e.g., -0.04 on Cutout-on-Syn, -0.03 on Cutout-on-Real), suggesting that the antipodal structure is a learned property of fine-tuning rather than an intrinsic property of the backbone. Notably, vanilla up-down already exhibits a moderate mirror (-0.54 to -0.81 at $\ell = 21$), likely reflecting language-prior asymmetries between vertical and horizontal direction tokens (Gurnee and Tegmark, 2024). `DeltaDirect` further sharpens the mirror, reaching -0.91 on Cutout-on-Real for both pairs.

E.7.4. Decodability: The Direction Axis Classifies Direction

We evaluate decodability via two complementary tests. First, we use the concept axes as a parameter-free classifier: predict $\hat{d} = \arg\max_d(\mathbf{h}^\ell \cdot \hat{\mathbf{v}}^\ell_d)$. Second, we train a 4-class linear

Table 22. **Direction axis is orthogonal to letter and identity controls.** Average absolute cosine $|\cos(\hat{\mathbf{v}}_{\text{dir}}, \hat{\mathbf{v}}_c)|$ between the direction axis and each control axis $c \in \{\text{letter, identity}\}$, averaged over class pairs. In the readout consolidation range ($\ell \in [20, 23]$), the fine-tuned models keep the direction-letter cosine at or below $\sim$0.17 and the direction-identity cosine at or below $\sim$0.21 across all domains, ruling out leakage of the answer-letter or identity axes into the amplified direction signal.

	$\lvert\cos(\hat{\mathbf{v}}_{\text{dir}}, \hat{\mathbf{v}}_{\text{letter}})\rvert$				$\lvert\cos(\hat{\mathbf{v}}_{\text{dir}}, \hat{\mathbf{v}}_{\text{identity}})\rvert$			
Domain	L17	L20	L21	L23	L17	L20	L21	L23
Vanilla								
Primitive-on-Syn	0.150	0.153	0.184	0.201	0.102	0.160	0.221	0.275
Cutout-on-Syn	0.113	0.113	0.179	0.240	0.138	0.215	0.217	0.217
Primitive-on-Real	0.109	0.107	0.104	0.125	0.097	0.160	0.173	0.180
Cutout-on-Real	0.102	0.113	0.101	0.114	0.112	0.129	0.125	0.146
Instruction-tuned								
Primitive-on-Syn	0.167	0.098	0.109	0.147	0.066	0.101	0.108	0.102
Cutout-on-Syn	0.106	0.102	0.135	0.150	0.085	0.101	0.136	0.114
Primitive-on-Real	0.113	0.103	0.122	0.162	0.080	0.092	0.131	0.150
Cutout-on-Real	0.096	0.114	0.136	0.161	0.088	0.108	0.187	0.201
`DeltaDirect`								
Primitive-on-Syn	0.157	0.100	0.107	0.142	0.075	0.116	0.133	0.128
Cutout-on-Syn	0.131	0.104	0.139	0.151	0.101	0.102	0.128	0.104
Primitive-on-Real	0.074	0.111	0.123	0.161	0.072	0.097	0.125	0.140
Cutout-on-Real	0.095	0.128	0.151	0.177	0.084	0.107	0.167	0.172

probe (Gurnee and Tegmark, 2024; Marks and Tegmark, 2024) $(\mathbf{W}, \mathbf{b})$ on cached readout states and (i) evaluate its standard sample-test accuracy and (ii) ask whether each concept vector $\mathbf{v}_d^\ell$, when standardized and fed to the probe, is classified as direction d (we report the diagonal count out of 4). Table 24 shows that on the fine-tuned models the parameter-free axis classifier reaches 99.7%/99.9% on Primitive-on-Syn at $\ell = 21$ and stays above 79% even on the hardest OOD Cutout-on-Real, far above the 25% chance baseline. The trained probe additionally classifies all four concept vectors as their corresponding direction in 100% of the 108 (model $\times$ task $\times$ layer) settings examined. The two tests are mathematically related (Marks and Tegmark, 2024) but use different normalizations and bias terms; their agreement strengthens the conclusion that $\hat{\mathbf{v}}_d^\ell$ is the same direction axis the probe relies on.

E.7.5. Orientation and Magnitude of Direction Concept Vectors

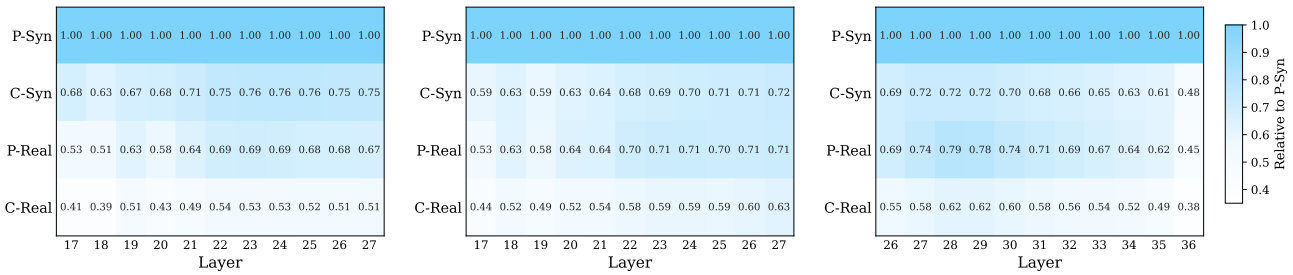
Orientation aligns uniformly across all domain pairs. Figure 5 (a) in the main text reports the cross-domain cosine averaged over the six unordered domain pairs and four directions per layer. Table 25 provides the per-pair breakdown at the consolidation layer $\ell = 21$. After instruction tuning, every pair exceeds 0.86, with the highest alignment between Primitive-on-Real-Cutout-on-Real (which share a real scene background) and the lowest between Primitive-on-Syn-Cutout-on-Real (which share no visual factor). `DeltaDirect` uniformly lifts alignment over the instruction-tuned baseline. In particular, the Primitive-on-Syn-Cutout-on-Real pair rises from 0.87 to 0.89, confirming that `DeltaDirect` produces a more domain-invariant orientation even on the most distant pair.

Orientation alignment generalizes to other backbones. To verify that the cross-domain alignment at $\ell = 21$ is not specific to LLaVA-Video-7B, we instruction-tune two additional Video-LLM backbones on MoDirect-Inst and measure the same metric. Table 26 reports the result. All three backbones reach cross-domain cosine above 0.84, with the LLaVA-family backbones converging on 0.93 and Qwen3-VL-4B at 0.85.

Magnitude does not generalize to OOD domains. While orientation aligns well across domains, magnitude does not transfer in the same way. Figure 9 reports the ratio of the concept-vector magnitude on each domain to the magnitude on the source domain Primitive-on-Syn, so Primitive-on-Syn equals 1.00 by construction and the other domains take values in $[0, 1]$. After instruction tuning on MoDirect-Inst, all three backbones exhibit a clear magnitude collapse on OOD domains, with the most distant domain Cutout-on-Real consistently retaining the smallest fraction of the source magnitude. For LLaVA-Video-7B at $\ell = 21$, Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real retain only 71%, 64%, and 49% of the Primitive-on-Syn magnitude, respectively. LLaVA-OV-7B at the same layer and Qwen3-VL-4B at $\ell = 28$ show the same pattern, with Cutout-on-Real retaining 54% and 62%, respectively. The same deficit appears across three backbones drawn from two different model families (LLaVA and Qwen3-VL), indicating that the magnitude collapse is a general property of instruction-tuned Video-LLMs rather than a model-specific artifact, motivating `DeltaDirect`.

Table 23. **Opposite directions form antipodal axes.** Signed cosine between opposite-direction concept vectors. After instruction tuning, both pairs reach -0.84 to -0.92 at $\ell = 21$ across all domains, while the vanilla model fails to establish left-right antipodality on OOD domains (e.g., -0.04 on Cutout-on-Syn, -0.03 on Cutout-on-Real). `DeltaDirect` sharpens the mirror further, reaching -0.91 on Cutout-on-Real for both pairs.

	$\cos(\hat{\mathbf{v}}_{\text{left}}, \hat{\mathbf{v}}_{\text{right}})$				$\cos(\hat{\mathbf{v}}_{\text{up}}, \hat{\mathbf{v}}_{\text{down}})$			
Domain	L17	L20	L21	L23	L17	L20	L21	L23
Vanilla								
Primitive-on-Syn	−0.349	−0.377	−0.349	−0.308	−0.722	−0.675	−0.719	−0.733
Cutout-on-Syn	+0.280	+0.136	−0.039	−0.138	−0.356	−0.411	−0.544	−0.618
Primitive-on-Real	−0.362	−0.259	−0.029	−0.058	−0.821	−0.816	−0.815	−0.857
Cutout-on-Real	+0.021	−0.185	−0.029	−0.026	−0.708	−0.839	−0.811	−0.856
Instruction-tuned								
Primitive-on-Syn	−0.932	−0.916	−0.902	−0.879	−0.901	−0.855	−0.844	−0.814
Cutout-on-Syn	−0.724	−0.923	−0.922	−0.909	−0.666	−0.850	−0.863	−0.828
Primitive-on-Real	−0.809	−0.902	−0.898	−0.881	−0.858	−0.881	−0.889	−0.861
Cutout-on-Real	−0.614	−0.902	−0.874	−0.857	−0.784	−0.892	−0.896	−0.872
`DeltaDirect`								
Primitive-on-Syn	−0.929	−0.925	−0.904	−0.886	−0.898	−0.869	−0.845	−0.821
Cutout-on-Syn	−0.668	−0.931	−0.934	−0.926	−0.647	−0.883	−0.899	−0.875
Primitive-on-Real	−0.816	−0.919	−0.924	−0.908	−0.868	−0.889	−0.911	−0.883
Cutout-on-Real	−0.644	−0.919	−0.908	−0.894	−0.792	−0.896	−0.918	−0.896



(a) LLaVA-Video-7B (MoDIRECT) (b) LLaVA-OV-7B (MoDIRECT) (c) Qwen3-VL-4B (MoDIRECT)

Figure 9. **Magnitude collapses on OOD domains across all three backbones.** Each cell shows the per-layer ratio of the concept-vector magnitude on the row's domain to that on the source domain P-Syn. The P-Syn row is therefore 1.00 by construction, while OOD rows below 1.00 indicate that OOD magnitudes are smaller than the source magnitude. P / C denotes the subject type (primitive shape or cutout image) and Syn / Real denotes the background type (synthetic solid or real scene). All three backbones, (a) LLaVA-Video-7B, (b) LLaVA-OV-7B, and (c) Qwen3-VL-4B, are instruction-tuned on MoDirect-Inst, yet OOD rows consistently fall below 1.00, with the largest deficit on C-Real.

E.7.6. Causality: Direction-Axis Injection Controls Direction Prediction

Specificity, decodability, and consistency leave open the possibility that $\hat{\mathbf{v}}_d^\ell$ is correlated rather than causal. A coincident axis with high decodability could still be a confounder. To probe causality, we intervene on the readout state along the concept axis, following the activation-steering protocol commonly used to causally validate concept directions in language models (Arditi et al., 2024; Rimsky et al., 2024; Li et al., 2023; Tigges et al., 2024). We intervene at $\ell = 21$ on the instruction-tuned model. For each OOD sample, we replace the on-canonical-axis projection of the readout state with $k \cdot m_{Primitive-on-Syn} \cdot \hat{\mathbf{v}}^\ell$, with $k = 1$ matching the average Primitive-on-Syn magnitude $m_{Primitive-on-Syn}$, and leave the orthogonal component unchanged. We compare four axis types: **canonical** ($\hat{\mathbf{v}}_{d_{\text{gold}}}$, the gold direction's axis); **random** (a fresh random unit vector in $\mathbb{R}^D$); **wrong-anti** (the opposite direction's axis—up↔down, left↔right); and **magnitude-only** (rescale $\|\mathbf{h}^\ell\|$ to the Primitive-on-Syn mean without axis selection). Table 27 reports OOD MCQ accuracy after each intervention. Canonical injection improves Cutout-on-Real by +14.6 pp and Cutout-on-Syn by +8.2 pp; random and magnitude-only baselines are essentially unchanged (±1.4 pp); and wrong-anti collapses Cutout-on-Real by −38.6 pp—well below chance (25%). This pattern is direction-axis-specific. Injecting the canonical direction axis at the canonical magnitude shifts the prediction toward the gold direction, while injecting the same magnitude on a non-direction axis or on the opposite-direction axis has either no effect or the opposite effect.

Table 24. **The direction axis classifies direction. Left:** parameter-free axis classifier accuracy $\arg\max_d(\mathbf{h}^\ell \cdot \hat{\mathbf{v}}_d^\ell)$ in %. **Right:** probe-classifies-axis: out of 4 concept vectors $\{\mathbf{v}_d^\ell\}_d$ standardized and fed to a trained 4-class probe, the count classified as their corresponding direction (parenthesized: probe's standard sample-test accuracy in %). The fine-tuned models reach near-perfect accuracy on Primitive-on-Syn and stay above 79% on the hardest OOD Cutout-on-Real (chance: 25%); the trained probe additionally classifies all 4 concept vectors correctly in 100% of the 108 (model × task × layer) settings.

	Axis-only acc. (%)				**Probe diag. count** (4) [probe acc. %]			
Domain	L17	L20	L21	L23	L17	L20	L21	L23
Vanilla								
Primitive-on-Syn	45.3	53.0	48.3	44.9	4 [99.0]	4 [98.1]	4 [96.3]	4 [92.1]
Cutout-on-Syn	38.7	37.3	37.3	36.5	4 [90.4]	4 [88.3]	4 [86.8]	4 [85.8]
Primitive-on-Real	33.8	36.5	39.3	38.3	4 [86.2]	4 [82.1]	4 [79.9]	4 [77.6]
Cutout-on-Real	34.6	35.5	36.4	36.3	4 [77.4]	4 [76.0]	4 [72.6]	4 [69.2]
Instruction-tuned								
Primitive-on-Syn	91.1	99.7	99.9	99.8	4 [99.9]	4 [100.0]	4 [100.0]	4 [100.0]
Cutout-on-Syn	72.1	93.0	92.2	90.8	4 [98.5]	4 [97.7]	4 [97.6]	4 [97.4]
Primitive-on-Real	55.3	88.1	88.9	87.1	4 [96.6]	4 [95.1]	4 [95.7]	4 [95.8]
Cutout-on-Real	51.0	78.9	79.1	75.5	4 [93.8]	4 [93.8]	4 [93.8]	4 [92.2]
`DeltaDirect`								
Primitive-on-Syn	91.5	99.3	99.7	99.6	4 [100.0]	4 [100.0]	4 [100.0]	4 [100.0]
Cutout-on-Syn	70.5	93.0	92.8	91.3	4 [98.5]	4 [97.5]	4 [97.6]	4 [97.4]
Primitive-on-Real	60.9	91.5	93.9	92.7	4 [97.6]	4 [97.3]	4 [97.4]	4 [97.2]
Cutout-on-Real	56.1	84.0	85.6	83.1	4 [95.6]	4 [94.6]	4 [94.8]	4 [93.7]

Table 25. **Cross-domain cosine similarity of direction concept vectors at LLM layer $\ell = 21$.** Each cell reports the mean cosine similarity $\cos(\hat{\mathbf{v}}_{d,A}^{21}, \hat{\mathbf{v}}_{d,B}^{21})$ between concept vectors of the same direction d on two different MODIRECT-SYNBENCH domains A and B, averaged over the four motion directions (left, right, up, down). Higher values indicate that the same direction is encoded in the same hidden subspace across domains. Fine-tuning lifts every pair above 0.86, and `DeltaDirect` further improves over the instruction-tuned baseline on five of six pairs.

Domain pair (A,B)	Cosine similarity $\cos(\hat{\mathbf{v}}_{d,A}^{21}, \hat{\mathbf{v}}_{d,B}^{21}) \uparrow$		
	Vanilla	Instruction-tuned	`DeltaDirect`
P-Syn – C-Syn	0.585	**0.964**	0.962
P-Syn – P-Real	0.406	0.916	**0.933**
P-Syn – C-Real	0.316	0.868	**0.894**
C-Syn – P-Real	0.325	0.940	**0.951**
C-Syn – C-Real	0.454	0.920	**0.944**
P-Real – C-Real	0.761	0.979	**0.981**
Mean	0.475	0.931	**0.944**

E.7.7. SYNTHESIS

The six measurements (existence in Appendix E.7.1, specificity in Appendix E.7.2, antipodal mirror in Appendix E.7.3, axis-only and probe-axis decodability in Appendix E.7.4, cross-domain consistency in Appendix E.7.5, and causal intervention in Appendix E.7.6) jointly indicate that $\hat{\mathbf{v}}_d^\ell$ extracted by difference-in-means in the readout state behaves as a magnitude-amplified, antipodally structured, and causally relevant axis of motion direction. The vanilla model exhibits a weak version of these properties, which fine-tuning amplifies. The remaining gap between the instruction-tuned model and `DeltaDirect` concentrates in OOD magnitude. `DeltaDirect` preserves the direction geometry while restoring signal strength, matching the shape of the OOD binding gap analyzed in Section 4.4.

Table 26. **Cross-domain alignment at layer 21 across Video-LLM backbones.** Cross-domain cosine similarity at $\ell = 21$, averaged over the six unordered domain pairs and four directions, after instruction tuning on MODIRECT-INST. The alignment generalizes across backbones.

Backbone	Cross-domain cosine
LLaVA-Video-7B	0.93
LLaVA-OneVision-7B (Li et al., 2025a)	0.93
Qwen3-VL-4B (Bai et al., 2025a)	0.85

E.8. Diagnostic Intervention: Additional Results and Controls

Intervention procedure. The intervention adjusts the on-axis magnitude of the readout state along the gold-direction concept vector to match the average Primitive-on-Syn magnitude, leaving the orthogonal component unchanged. For a sample with ground-truth direction d in domain A, we first measure the current on-axis magnitude

$$\alpha = \langle \mathbf{h}^\ell - \mathbf{g}_A^\ell,\ \hat{\mathbf{v}}_{d,A}^\ell \rangle, \tag{7}$$

where $\mathbf{g}_A^\ell$ is the per-domain mean readout state and $\hat{\mathbf{v}}_{d,A}^\ell$ is the unit concept-vector orientation (Section 4.4). We then shift the readout state along this axis by the deficit between α and the target magnitude $m_{P-Syn} = \mathbb{E}_d \|\mathbf{v}_{d,P-Syn}^\ell\|$ (the

Table 27. **Controls confirm the causal specificity of the direction axis.** We intervene at layer ℓ=21 on LLaVA-Video and rescale the readout state to match the average Primitive-on-Syn magnitude along four axis types: *Canonical* (motion-direction concept vector $\hat{\mathbf{v}}^{\ell}_{d,A}$), *Random* (a fresh unit vector $\hat{\mathbf{r}} \sim \mathcal{N}(0, I/D)$), *Magnitude-only* (rescale $\|\mathbf{h}^{\ell}\|$ to the source-domain mean without any axis re-projection), and *Wrong-anti* (the antipodal direction's concept vector $\hat{\mathbf{v}}^{\ell}_{\bar{d},A}$, e.g., up↔down). Only the canonical axis recovers MCQ accuracy across all OOD domains, with the largest gain of +14.6 pp on Cutout-on-Real. Random and magnitude-only interventions yield only minor effects ($\leq +1.4$ pp), an order of magnitude smaller than the canonical recovery. Wrong-anti interventions instead degrade accuracy substantially (up to −39.3 pp on Primitive-on-Real), confirming that the on-axis projection causally drives answer-option binding.

	Cutout-on-Syn		Primitive-on-Real		Cutout-on-Real	
Axis type	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)
No intervention	80.7	—	74.7	—	60.5	—
Canonical	**88.9**	**+8.2**	**85.2**	**+10.5**	**75.1**	**+14.6**
Random	81.8	+1.1	75.8	+1.1	61.9	+1.4
Magnitude-only	81.6	+0.9	75.4	+0.7	61.8	+1.3
Wrong-anti	49.6	−31.1	35.4	−39.3	21.9	−38.6

average Primitive-on-Syn magnitude):

$$\mathbf{h}^{\ell}_{\text{new}} = \mathbf{h}^{\ell} + (m_{P-Syn} - \alpha)\,\hat{\mathbf{v}}^{\ell}_{d,A}. \tag{8}$$

Downstream layers then run normally.

The intervention is layer-localized around the readout: ℓ=20, 21 is best on LLaVA-Video. We first identify the optimal intervention layer on LLaVA-Video. Table 28 reports MCQ accuracy on MoDirect-SynBench when applying the intervention at layers near the readout. Single-layer interventions are most effective around ℓ=21 and ℓ=22, with ℓ=22 slightly best on C-Syn and ℓ=21 matching or improving the strongest recovery on P-Real and C-Real. Combining the two adjacent layers, ℓ=20, 21, yields the best overall result, improving accuracy to 89.6 on C-Syn, 85.9 on P-Real, and 76.0 on C-Real. In contrast, interventions at more distant layers have much smaller effects, especially at ℓ=14 and ℓ=25. This shows that the magnitude deficit is localized to a narrow late-layer region near the readout, consistent with the per-layer magnitude profile inSection 4.4.

The recovery is direction-specific, not norm-driven. We additionally verify that the recovery is driven by the motion-direction concept vector itself, not by a generic perturbation effect, by replacing $\hat{\mathbf{v}}^{\ell}_{d,A}$ with three control axes at the same target magnitude on LLaVA-Video: (i) a random unit vector, (ii) the antipodal direction's concept vector (e.g., up↔down), and (iii) magnitude-only rescaling of $\mathbf{h}^{\ell}$ with no axis re-projection. As shown in Table 27, random and magnitude-only interventions yield negligible change (≤ 1.4 pp across all domains), ruling out a generic norm-scaling effect. The antipodal axis catastrophically reduces accuracy (−38.6 pp on Cutout-on-Real), confirming that pushing the readout state along an incorrect direction axis actively misleads the answer-option binding. The on-axis magnitude along the correct concept-vector orientation—not the norm itself—is what drives OOD recovery.

E.9. Delta Feature Validation

Section 5.1 introduces the delta descriptor as the substrate of `DeltaDirect`'s auxiliary supervision:

$$\delta_t = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{F}_{t+1}[n] - \mathbf{F}_t[n]) \in \mathbb{R}^D, \qquad t = 1, \ldots, T-1. \tag{9}$$

This expression encodes three core design choices: subtraction, an adjacent frame pair, and the order $\mathbf{F}_{t+1} - \mathbf{F}_t$. We verify each choice on the vanilla LLaVA-Video backbone, without any `DeltaDirect` training, by linearly probing the projector output for two semantic targets — object identity and motion direction. We exploit a structural property of LLaVA-Video: the image encoder processes each frame independently, so $\mathbf{F}_t$ and $\mathbf{F}_{t+1}$ are independent encodings of two visually similar frames; their difference cancels shared content and isolates motion-induced change.

E.9.1. Setup

We probe the projector output of vanilla LLaVA-Video on the four held-out MoDirect-SynBench domains: Primitive-on-Syn, Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real. Each video has $T = 8$ frames. We compare five feature constructions built from the per-frame projector outputs $\mathbf{F}_t \in \mathbb{R}^{N \times D}$, each spatially averaged to a single D-dimensional descriptor before probing:

- Single: a single mid-clip frame $\mathbf{F}_{T/2}$.
- T-mean: the temporal mean $(1/T)\sum_t \mathbf{F}_t$.
- Stack: the concatenation of all T frames.
- Delta: a single signed delta $\mathbf{F}_{T-1} - \mathbf{F}_0$.
- Concat. Delta (consecutive differences) : the concatenation of the seven adjacent deltas $\{\mathbf{F}_{t+1} - \mathbf{F}_t\}_{t=0}^{T-2}$.

Table 28. **Layer ablation of the magnitude intervention on LLaVA-Video.** We apply the intervention at different layers and report MCQ accuracy on MoDirect-SynBench. Δ denotes the change relative to no intervention. The intervention reaches its full effect at ℓ=21 and remains stable through ℓ=22 and ℓ=23, with partial effect at ℓ=20 and negligible effect at distant layers (ℓ=14, ℓ=25). This confirms that the OOD magnitude deficit is localized to a narrow window around the readout layer.

	MoDirect-SynBench					
	C-Syn		P-Real		C-Real	
ℓ	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)	Acc. (%)	Δ (pp)
No intervention	80.7	—	74.7	—	60.5	—
14	81.6	+0.9	75.5	+0.8	61.6	+1.1
20	84.5	+3.8	80.0	+5.3	68.4	+7.9
21	88.9	+8.2	85.2	+10.5	75.1	+14.6
22	89.1	+8.4	85.0	+10.3	75.1	+14.6
23	88.9	+8.2	84.6	+9.9	74.9	+14.4
25	82.2	+1.5	76.2	+1.5	62.9	+2.4
20, 21	**89.6**	**+8.9**	**85.9**	**+11.2**	**76.0**	**+15.5**

We train two linear probes independently: an identity probe (object class for Cutout-on-Syn/Cutout-on-Real, shape for Primitive-on-Syn/Primitive-on-Real; 26-30 classes) and a 4-way motion direction probe (left, right, up, down). Each probe is a single linear layer trained for 50 epochs with AdamW (lr 10^{-3}, weight decay 10^{-2}) on a 70/30 stratified split, with per-dimension z-scoring fit on the train split. This protocol matches the probing analyses in Section 3 and Appendix E.7.

E.9.2. Identity Cancels, Motion Direction Survives

The first design choice is the subtraction itself. Within the delta descriptor

$$\delta_t \;=\; \frac{1}{N}\sum_{n=1}^{N}(\mathbf{F}_{t+1}[n]-\mathbf{F}_t[n]),$$

the subtraction is what removes content shared between the two frames. Table 29 shows that all five feature constructions encode object identity to a high degree, with one exception: Delta and Concat. Single-frame, temporal-mean, and stacked features classify identity at 59.7 %-90.2 % accuracy. Delta drops to 8.9 %-21.8 % and Concat. Delta to 8.1 %-29.7 % — a 55-68 percentage-point reduction relative to T-mean, within a few points of the per-domain chance level ($1/26$ to $1/30 \approx$ 3-4 %). Object class, color, shape, and place remain stable across adjacent frames, and subtraction cancels them by construction.

By contrast, the same Delta representation reaches 86.7 %-99.1 % on the 4-way motion direction probe (chance 25 %), at parity with or above Stack. Single-frame and temporal-mean features are essentially uninformative about motion direction (23.6 %-41.6 %, near or below chance). Motion direction therefore lives in the *change* between frames, not in any single frame's content. On the harder OOD domains (Primitive-on-Real, Cutout-on-Real), Delta outperforms Stack by 30.6 pp on Primitive-on-Real and 19.5 pp on Cutout-on-Real — with the static appearance gone, the linear probe finds the motion direction signal more easily.

Figure 10 summarizes this trade-off in the (identity, motion direction) plane, with one point per feature construction averaged over the four domains. Single and T-mean land in the high-identity, near-chance-direction region; Stack sits in the high-identity, high-direction region; only Delta and Concat. Delta occupy the low-identity, high-direction region. This region — appearance erased, motion direction preserved — is the regime that makes delta features a clean substrate for the MVP loss in Section 5.2.

E.9.3. Why Adjacent? Granularity, Not Quality

The second design choice is the adjacency of the frame pair. We can use any frame gap k in

$$\Delta_k(t) \;=\; \mathbf{F}_{t+k}-\mathbf{F}_t, \qquad k\in\{1,\dots,T-1\}.$$

Table 30 reports the motion direction probe accuracy for $k\in\{1,\dots,7\}$ on the four domains, averaged over all valid starting points t. Motion direction probe accuracy is essentially identical between k=1 and k=7, with less than 1 percentage point difference on every domain. For example, on Primitive-on-Syn post-projector, accuracy is 99.0 % at $k=1$ and 99.1 % at $k=7$; on Cutout-on-Real post-projector, 86.4 % at $k=1$ and 86.7 % at $k=7$. The delta operator preserves motion direction at every gap.

The choice of $k=1$ is therefore not about representation quality but about *supervision granularity*. The MVP super-

Table 29. **Differencing cancels identity while preserving motion direction.** Linear-probe accuracy on the post-projector representation of vanilla LLaVA-Video, no `DeltaDirect` training. Identity classes are 30 (shape) for Primitive-on-Syn/Primitive-on-Real and 26 (object) for Cutout-on-Syn/Cutout-on-Real; motion direction is 4-way. Single, temporal-mean, and stacked features encode identity at 74-90 %, while Delta and Concat. Delta drop near chance (8-30 %); the same delta representations preserve motion direction at parity with or above Stack (chance 25 %).

		Identity probe acc. (%)					**Motion direction probe acc. (%)**				
Domain	chance	single	tmean	stack	Delta	Concat. Delta	single	tmean	stack	Delta	Concat. Delta
Primitive-on-Syn (shape, 30-cls)	3.3	84.7	90.2	88.3	**21.8**	**29.7**	39.6	29.2	96.7	**99.1**	**95.6**
Cutout-on-Syn (object, 26-cls)	3.8	88.0	87.8	88.2	**21.1**	**17.9**	38.1	33.4	84.5	**93.7**	**76.7**
Primitive-on-Real (shape, 30-cls)	3.3	60.8	64.1	59.7	**8.9**	**8.1**	30.1	25.2	58.4	**89.0**	**68.4**
Cutout-on-Real (object, 26-cls)	3.8	73.8	75.6	72.6	**10.2**	**11.4**	34.3	31.6	67.2	**86.7**	**67.4**

Table 30. **Motion direction is preserved at all frame gaps k.** 4-way motion direction probe accuracy on $\Delta_k = \mathbf{F}_{t+k} - \mathbf{F}_t$, averaged over valid t. Variation across k is below 1 pp on every domain; the choice of $k = 1$ is therefore not driven by representation quality but by supervision granularity ($T - 1$ adjacent pairs per clip, each with a well-defined instantaneous motion direction target).

Domain	k=1	k=2	k=3	k=4	k=5	k=6	k=7
Primitive-on-Syn	**99.0**	98.9	98.8	98.9	98.9	99.1	99.1
Cutout-on-Syn	**93.8**	94.2	94.3	93.8	94.5	93.9	93.6
Primitive-on-Real	**89.2**	89.1	87.3	86.7	87.3	88.7	89.1
Cutout-on-Real	**86.4**	86.8	86.8	85.2	86.9	87.2	86.7

vision uses synthetic motion vectors as targets:

$$\mathbf{m}_t \;=\; \frac{\mathbf{c}_{t+1} - \mathbf{c}_t}{\|\mathbf{c}_{t+1} - \mathbf{c}_t\|_2 + \varepsilon} \;\in\; \mathbb{R}^2,$$

where $\mathbf{c}_t$ is the object center at frame t (Section 5.2). A clip of T frames yields $T - 1$ adjacent pairs, each with a well-defined instantaneous motion direction target $\mathbf{m}_t$. A coarser gap collapses multiple instantaneous motion directions into a single coarse target, reducing the number of supervised motion vectors per clip and discarding fine-grained information about how motion direction evolves within the clip — a particularly relevant concern for compound trajectories such as zigzag, roundtrip, or curved motion. We adopt $k = 1$ to match the granularity at which $\mathbf{m}_t$ is cleanly defined, not because $k = 1$ is empirically distinguished from $k > 1$ at the representation level.

E.9.4. Delta Features Carry Signed Motion Direction

The third design choice is the order of subtraction. The delta operator computes $\mathbf{F}_{t+1} - \mathbf{F}_t$, not $|\mathbf{F}_{t+1} - \mathbf{F}_t|$ or $\|\mathbf{F}_{t+1} - \mathbf{F}_t\|$. Sign matters because the MVP target is a signed unit vector,

$$\mathbf{m}_t \;\in\; \mathbb{R}^2, \qquad \|\mathbf{m}_t\|_2 = 1,$$

where left and right map to opposite values. A representation that confuses them cannot be supervised by L_{MVP}.

We test whether the delta representation respects this sign. We train the motion direction probe on forward deltas $\Delta_t = \mathbf{F}_{t+1} - \mathbf{F}_t$ as before. We then evaluate the same probe, without retraining, on *reverse* deltas $-\Delta_t = \mathbf{F}_t - \mathbf{F}_{t+1}$, which corresponds to feeding the temporally reversed clip. A genuinely signed representation should send each motion direction to its antipodal class: left ↔ right and up ↔ down.

Table 31. **Delta features carry signed motion direction.** A motion direction probe trained on forward deltas Δ_t is evaluated, without retraining, on reverse deltas $-\Delta_t$. Reverse accuracy collapses to near-zero, but the *antipodal* accuracy (rate of predicting the opposite class: left↔right, up↔down) recovers most of the forward accuracy. The representation inverts the sign of motion direction under temporal reversal, exactly as a signed displacement signal should.

Domain	forward acc.	reverse acc.
Primitive-on-Syn	98.7	0.0
Cutout-on-Syn	93.7	0.7
Primitive-on-Real	88.9	0.6
Cutout-on-Real	87.0	1.4

Table 31 confirms this prediction. Forward accuracy is high across all domains (87.0 %-98.7 % on post-projector), while reverse accuracy collapses to near zero (0.0 %-1.4 %). The delta representation does not lose motion direction information under temporal reversal; it inverts the sign of that information, exactly as a signed displacement signal should. This property allows the MVP loss to supervise the full $\mathbf{m}_t \in \mathbb{R}^2$ rather than a discrete 4-class motion direction label.

E.9.5. Synthesis

Each component of the delta descriptor maps to a property of the vanilla backbone, before any `DeltaDirect`

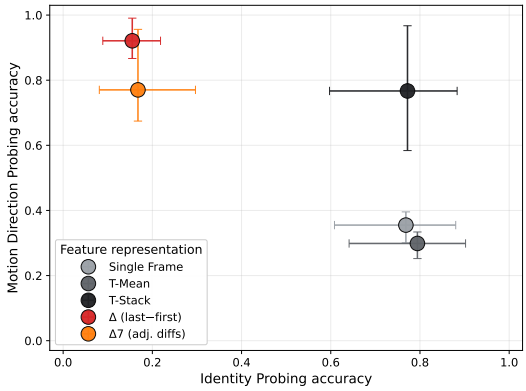


Figure 10. **Identity-vs-motion-direction trade-off on the post-projector representation.** Each point is one feature construction (Single, T-mean, Stack, Delta, Concat. Delta), averaged over the four MoDirect-SynBench domains; error bars span the domain min and max. Single-frame and temporal-mean features encode identity well but barely exceed chance (25 %) on motion direction (right-low region). Temporal-stack additionally captures motion direction, but retains identity (right-high region). Only Delta (Δ, last−first) and Concat. Delta (Δ_7, seven adjacent differences) occupy the upper-left region, where identity probing falls near per-domain chance (3-4 %) while motion direction probing remains at parity with or above Stack. This regime — appearance erased, motion direction preserved — is unique to the differencing operator and is what makes delta features a clean substrate for the MVP loss in Section 5.2.

training: the subtraction yields a representation in which appearance is canceled and motion direction is preserved (Appendix E.9.2); the adjacency of the frame pair is not required for motion direction preservation but matches the per-pair granularity of the MVP target (Appendix E.9.3); the ordering of the subtraction is faithfully reflected in the antipodal structure of the delta representation, making a signed 2-D target well-defined (Appendix E.9.4). `DeltaDirect` does not impose these properties on the backbone — it exploits properties that the operator already produces, and aligns the projector output with a signed motion direction target via the MVP loss.

F. Additional Experimental Result

F.1. Ablation Study of `DeltaDirect`

This section provides ablation studies for `DeltaDirect`. Our goal is to verify that the improvement does not come from additional training alone, but from the proposed motion direction supervision and its feature construction. We therefore compare alternative variants that modify the auxiliary objective, the motion feature used for supervision, and the training configuration while keeping the remaining components fixed.

F.1.1. Setup

We vary one design axis at a time around an anchor configuration and report the impact on direction performance. The anchor configuration follows Section 5: LLaVA-Video (Qwen2 (Yang et al., 2024) LLM, SigLIP (Zhai et al., 2023) vision encoder), $T = 8$ frames, LoRA rank $r = 64$ with $\alpha = 128$, projector and LLM both LoRA-tunable, $\lambda_{\text{MVP}} = 1.0$, mean spatial pooling, a linear motion head, the subtraction temporal operator, and MSE loss against unit-vector targets.

Evaluation suite. We report seven direction tasks and two general-video tasks. The seven direction tasks are: 4-way directional MCQ on the four MoDirect-SynBench; chance 25%, 1,500 samples per domain), and the three MoDirect-RealBench sub-tasks KTH-VP, ssv2-VP, and TOMATO direction. For compactness, we abbreviate the four MoDirect-SynBench domains as P-Syn, C-Syn, P-Real, and C-Real, where P/C denote Primitive/Cutout foregrounds and Syn/Real denote synthetic/real backgrounds. *Dir avg.* denotes the unweighted mean over these seven tasks. NextQA-MC and MVBench (mean over 20 sub-tasks) monitor general video understanding. The notation – in subsequent tables indicates evaluations in progress at submission time.

F.1.2. Supervision Location

Pre-LLM supervision is the only effective tap. We compare supervision at the projector output (anchor) against three alternative tap families: (i) vision-encoder intermediate layers (SigLIP blocks 7, 14, 21), (ii) LLM hidden states (layers 7, 14, 21), and (iii) two-tap combinations of pre- and post-LLM supervision. Table 32 shows that only the anchor improves Dir avg. over the no-auxiliary baseline by a substantial margin (+6.8 pp); every other tap point lies within ±1 pp of the baseline or below it.

Vision-encoder taps leave the downstream representation almost unchanged. SigLIP already encodes motion direction densely (Appendix E.9), and supervising it adds gradient that does not propagate to the LLM-side readout. LLM-side taps fail in a different way: early-layer supervision (L7) collapses MCQ formatting and pushes MoDirect-

Table 32. **Ablation: location of MVP supervision (full breakdown).** During MoDirect-Inst instruction tuning, we apply MVP supervision at different points along the Video-LLM pipeline and report Top-1 accuracy (%) on the four MoDirect-SynBench domains (P-Syn, C-Syn, P-Real, C-Real; chance 25%) and three MoDirect-RealBench sub-tasks (SSv2, KTH, TOMATO). Avg. Direction is the unweighted mean over MoDirect-SynBench and MoDirect-RealBench; Harm. is the harmonic mean over Avg. Direction and MVBench (Li et al., 2024a). ℓ_v and ℓ denote the layer index of the vision encoder (Zhai et al., 2023) ($L_v = 27$) and LLM (Yang et al., 2024) ($L = 28$), respectively.

Stage	Feat.	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench	Harm.
		P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO			
None	–	99.5	80.7	74.7	60.5	72.4	66.6	35.2	69.9	60.9	65.1
Vision encoder	$\mathbf{V}^{\ell_v=7}$	99.8	81.6	76.6	62.3	72.6	66.7	35.5	70.7	60.6	65.3
	$\mathbf{V}^{\ell_v=14}$	99.8	82.3	76.7	62.7	73.4	69.2	35.2	71.3	60.9	65.7
	$\mathbf{V}^{\ell_v=21}$	99.7	81.6	76.3	62.1	73.1	67.3	35.5	70.8	60.7	65.4
Pre-projector	$\mathbf{V}^{L_v}$	99.8	86.2	86.9	71.4	78.1	71.6	37.0	75.7	60.2	67.1
Post-projector (Ours)	$\mathbf{F}$	**99.7**	**84.9**	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	**60.7**	**67.8**
LLM visual tokens	$\mathbf{z}^{\ell=7}$	27.2	27.1	27.6	26.9	50.6	42.0	20.6	31.7	56.6	40.6
	$\mathbf{z}^{\ell=14}$	83.8	72.9	63.1	60.0	73.8	68.6	32.5	65.0	57.4	61.0
	$\mathbf{z}^{\ell=21}$	79.0	62.1	63.6	51.5	66.9	71.4	26.6	60.2	57.5	58.8
Final readout	$\mathbf{h}^{L}$	28.5	26.7	30.1	25.4	56.8	50.2	20.5	34.0	59.9	43.4
Two-tap	$\mathbf{F}, \mathbf{z}^{L}$	63.6	38.0	53.3	41.3	60.8	55.1	27.3	48.5	60.3	53.8
	$\mathbf{F}, \mathbf{z}^{\ell=4}$	45.2	32.2	38.3	32.8	54.8	54.7	24.1	40.3	57.6	47.4

SynBench accuracies to chance, while mid- and late-layer supervision (L14, L21) underperforms even the no-auxiliary baseline. Direct supervision of post-LLM hidden states interferes with the autoregressive readout that produces the answer letter.

F.1.3. Direction Head

Table 33. **Ablation: direction head.** Linear head ($D \to 2$, anchor) versus MLP head ($D \to 256 \to 2$ with GELU). Linear wins on 5 of 7 direction tasks; the motion direction signal is low-rank.

Head	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO		
Linear (anchor)	99.7	84.9	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	**60.7**
MLP	**99.8**	**87.2**	82.8	68.1	78.1	71.1	35.2	74.6	60.5

The motion direction signal is low-rank; a linear head suffices. We compare a linear head ($D \to 2$, anchor) against an MLP head ($D \to 256 \to 2$ with GELU). The linear head wins on 5 of 7 direction tasks (Dir avg. 76.7 vs. 74.6); the MLP variant wins only on Cutout-on-Syn (+2.3 pp) and Primitive-on-Syn (+0.1 pp, ceiling). Extra head capacity over-fits noise without recovering signal, so we adopt the linear head.

F.1.4. Pool Mode

Mean pooling matches the structure of the MVP target; flatten over-fits. After the temporal operator, the per-patch delta tensor $\Delta_t \in \mathbb{R}^{N \times D}$ is reduced to a D-dimensional descriptor by spatial pooling. We compare mean (anchor, head input dimension D) against flatten (head input dimension $N \cdot D = 702{,}464$, ~ 1.4M extra parameters). Flatten under-performs mean by 5.7 pp on Dir avg. (76.7 vs. 71.0); MVBench is essentially identical (60.7 vs. 60.6). The MVP target is a single 2-D vector per frame pair, so spatial information beyond the per-patch mean is not directly supervised — and consistent with the low-rank head ablation (Appendix F.1.3), the extra capacity over-fits noise.

Table 34. **Ablation: spatial pool mode.** Mean pooling (anchor) reduces the per-patch delta tensor $\Delta_t \in \mathbb{R}^{N \times D}$ to D dimensions; flatten preserves all $N \cdot D = 702{,}464$ entries (~ 1.4M extra parameters). Flatten under-performs mean by 5.2 pp on Avg. Direction while leaving MVBench unchanged.

Pool	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO		
Mean (anchor)	99.7	**84.9**	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	**60.7**
Flatten	**99.8**	82.0	77.8	61.2	72.3	67.3	36.7	71.0	60.6

F.1.5. Direction Loss

Table 35. **Ablation: direction loss.** MSE on $(\cos\theta, \sin\theta)$ (anchor), cosine loss. Cosine drops 4.0 pp on Avg. Direction vs. MSE due to its magnitude invariance.

Loss	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO		
MSE (anchor)	**99.7**	84.9	85.2	71.7	**81.5**	74.8	**38.8**	76.7	**60.7**
Cosine	**99.7**	84.0	78.2	64.5	75.9	69.3	35.5	72.7	60.4

Table 36. **Ablation: temporal mixing operator.** We compare subtraction, concat_linear, and gated_diff. Although gated_diff achieves the highest direction average, subtraction provides the best overall trade-off when direction accuracy and general video understanding are combined by harmonic mean.

Operator	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench	Harm. Mean
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO			
Subtract (anchor)	99.7	84.9	85.2	71.7	81.5	74.8	38.8	76.7	**60.7**	**67.8**
concat_linear	99.5	79.4	76.4	60.6	74.9	68.9	36.2	70.8	60.6	65.3
gated_diff	**99.8**	**89.8**	**90.5**	**83.4**	**82.1**	**80.6**	**42.9**	**77.3**	58.9	66.9

F.1.6. Temporal Mixing Operator

Subtraction provides the best overall trade-off. We compare the anchor operator, subtraction ($\mathbf{F}_{t+1} - \mathbf{F}_t$), with concat_linear (Linear($[\mathbf{F}_t; \mathbf{F}_{t+1}]$)) and gated_diff ($\sigma(\mathbf{W}\mathbf{F}_t) \odot (\mathbf{F}_{t+1} - \mathbf{F}_t)$). As shown in Table 36, concat_linear under-performs subtraction by 5.9 pp on Dir Avg. (70.8 vs. 76.7), suggesting that the subtraction prior is useful under the limited supervision of MoDirect-Inst. gated_diff achieves the highest Dir Avg. (77.3, +0.6 pp over subtraction), but reduces MVBench performance (58.92 vs. 60.7). When combining Dir Avg., NExT-QA, and MVBench with a harmonic mean, subtraction performs best (67.8), outperforming both concat_linear (65.3) and gated_diff (66.9). We therefore adopt subtraction as the default operator, as it offers the best balance between direction improvement and general video understanding.

F.1.7. Loss Weight

Table 37. **Ablation: loss weight λ_{MVP}.** Total objective $L = L_{\text{LM}} + \lambda_{\text{MVP}} \cdot L_{\text{MVP}}$. Direction performance scales monotonically with λ in the explored range; MVBench is flat across the sweep.

λ_{MVP}	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO		
0.1	**99.8**	82.5	76.3	62.2	72.8	68.6	35.5	71.1	**60.7**
0.5	**99.8**	84.5	82.4	67.4	76.6	67.3	36.5	73.5	60.4
1.0 (anchor)	99.7	**84.9**	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	**60.7**

Direction performance is monotone in λ_{MVP}; MVBench is flat across the sweep. The total training objective is $L = L_{\text{LM}} + \lambda_{\text{MVP}} \cdot L_{\text{MVP}}$; we sweep $\lambda_{\text{MVP}} \in \{0.1, 0.5, 1.0\}$. Dir avg. scales monotonically: $71.1 \rightarrow 73.5 \rightarrow 76.7$ for $\lambda \in \{0.1, 0.5, 1.0\}$. The anchor at $\lambda = 1.0$ sits at or below the knee; values $\lambda \geq 1.5$ are worth probing in future work. MVBench moves by less than 0.4 pp across the sweep (60.42-60.73), indicating that λ controls direction-specific signal without perturbing general video understanding.

F.1.8. LoRA Rank

Direction performance is monotone in rank; $r = 64$ is on the budget knee. We use LoRA with rank r and $\alpha = 2r$ on both the projector and the LLM, with $r \in \{16, 32, 64\}$. $r = 16$ already learns the in-domain task (Primitive-on-Syn 98.0%) but degrades on harder splits and on real-world directional tasks. MVBench is essentially flat across the range (60.2-60.7), so the cost of higher rank is borne entirely by the direction signal. We adopt $r = 64$ as the anchor.

Table 38. **Ablation: LoRA rank.** LoRA rank r with $\alpha = 2r$ on both projector and LLM. Direction performance is monotone in rank; MVBench is essentially flat across the range.

Rank r	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench
	P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO		
16	98.0	70.8	73.5	57.5	67.3	58.1	34.0	65.6	60.2
32	99.3	80.1	80.7	63.4	72.2	64.4	34.5	70.7	60.5
64 (anchor)	**99.7**	**84.9**	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	**60.7**

F.1.9. Number of Frames

Table 39. **Ablation: frame count.** Frames per video T at fixed effective batch size (144). $T = 8$ is the optimal frame budget for both baseline and `DeltaDirect`; $T = 16$ and $T = 32$ both under-perform. The relative gain of `DeltaDirect` over baseline is preserved at $T = 32$. *Avg. Direction* is the unweighted mean over the seven direction tasks; *Harm. Mean* is the harmonic mean over Avg. Direction and MVBench.

Method	T	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction	MVBench	Harm. Mean
		P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO			
Baseline	8	99.5	80.7	74.7	60.5	72.4	66.6	35.2	69.9	**60.9**	65.1
	32	96.6	71.1	63.3	51.2	67.3	59.8	27.8	58.4	60.1	59.2
`DeltaDirect`	**8 (anchor)**	**99.7**	**84.9**	**85.2**	**71.7**	**81.5**	**74.8**	**38.8**	**76.7**	60.7	**67.8**
	16	96.9	79.8	82.1	68.2	75.9	66.3	31.0	71.5	59.6	65.0
	32	96.4	77.6	82.3	68.1	76.5	69.6	28.5	71.3	59.4	64.8

8 frames sufficient We compare $T \in \{8, 16, 32\}$ frames per video at fixed effective batch size (144). $T = 8$ (anchor) is best on Dir avg. (76.7); $T = 16$ drops to 71.5 and $T = 32$ to 71.3. The relative gain of `DeltaDirect` over baseline is preserved at $T = 32$ (+12.9 pp; baseline drops by 11.5 pp from $T = 8$ to $T = 32$). The synthetic clips contain a single coherent motion that 8 frames already capture, so extra frames add gradient noise (smaller per-step batch, more padding) without adding new content. The MVBench evaluation for $T = 16$ is in progress at submission time.

F.1.10. Design of the auxiliary branch.

Table 40. **Ablation: auxiliary branch design.** We compare three auxiliary branch architectures for supervising motion direction from adjacent-frame feature deltas and report accuracy (%).

Method	MoDirect		MVBench
	SynBench	RealBench	
Residual injection	80.5	58.0	60.2
Dual-stream fusion	84.4	62.0	60.4
`DeltaDirect`	**85.2**	**64.1**	**60.6**

Design of the auxiliary branch. We compare `DeltaDirect` against two alternative branch architectures (Table 40). Residual injection compresses the delta into a bottleneck and adds a learned residual back to the LLM input tokens. Dual-stream fusion combines pre-

and post-projector deltas to recover motion information lost through the nonlinear projection. `DeltaDirect` outperforms both with a simpler design: a single linear head on the post-projector delta.

F.1.11. What supervision target to use.

In Table 3 of the main paper, we compare `DeltaDirect` against three alternative auxiliary objectives.

Frame order. We attach a small classification head to LLM hidden states at layer 16 that predicts each frame's temporal position via cross-entropy with target $[0, 1, \ldots, T-1]$. This post-LLM objective pressures the language model to maintain temporally discriminative representations, without requiring any direction labels.

Concatenating feature deltas. We compute per-pair frame deltas $\Delta_t = F_{t+1} - F_t$ from the projector output, average them over the spatial dimension, and concatenate the resulting $T-1$ motion tokens to the LLM input sequence. No auxiliary loss or learnable parameter is introduced; the LLM must learn to exploit the appended motion tokens through the standard language modeling objective alone.

Delta equivariance. We collect all spatially averaged frame deltas across the batch from the projector output and compute their pairwise cosine similarity matrix. An MSE loss aligns this representation similarity matrix with the corresponding direction ground-truth similarity matrix. This regularizer constrains relative structure rather than regressing direction directly, and introduces no additional learnable parameters.

Results. `DeltaDirect` outperforms all alternatives, and the direction-agnostic objectives provide limited gains, confirming that explicit direction supervision on the signed 2-D motion vector is necessary.

F.2. `DeltaDirect` Across Video-LLM Backbones

To verify that `DeltaDirect` generalizes beyond our main backbone, we apply it to additional Video-LLMs with minimal architecture-specific changes. We keep the loss formulation, head design, and training recipe identical across backbones, adjusting only the layer from which the head reads visual features. For Qwen3-VL-4B, the vision encoder merges every two adjacent input frames through a 3D convolution at its entry. We therefore apply the MVP loss directly on the output of the projection module, where each token already corresponds to a two-frame chunk.

`DeltaDirect` improves direction accuracy across all tested backbones. We replicate `DeltaDirect` on three Video-LLM backbones spanning two LLM families and two vision encoders: LLaVA-Video-7B (Qwen2 LLM, SigLIP encoder), LLaVA-OneVision-7B (Qwen2 LLM, SigLIP encoder), and Qwen3-VL-4B (Qwen3 LLM). `DeltaDirect` improves Avg. Direction over the corresponding instruction-tuning baseline on every backbone, with the largest gain on Qwen3-VL (+24.0 pp) and consistent gains on LLaVA-Video (+6.3 pp) and LLaVA-OneVision (+0.9 pp). The smaller gain on LLaVA-OneVision reflects a stronger baseline (Avg. Direction 70.7 pp), already close to that of LLaVA-Video. We adopt LLaVA-Video-7B as the primary backbone for the strongest baseline and the cleanest signal for ablation analysis throughout the paper.

F.3. Quantitative Results

F.3.1. Full MoDirect-SynBench and MoDirect-RealBench Results

Setup. This section extends Table 1 with all evaluated baselines. We evaluate on MoDirect-SynBench (Primitive-on-Syn, Cutout-on-Syn, Primitive-on-Real, Cutout-on-Real) and MoDirect-RealBench (SSv2, KTH, TOMATO). The table groups methods into closed-source models, open-source Video-LLMs, our LoRA post-tuning of LLaVA-Video-7B, and our full fine-tuning of a 0.5B Video-LLM. The Full-FT group contains three variants on the same Qwen2-0.5B + SigLIP backbone. The first variant trains on VideoChat2 instructions alone. The second variant adds MoDirect-Inst as additional instruction data. The third variant trains on the same data as the second but applies `DeltaDirect`, isolating the contribution of motion-vector supervision.

Existing Video-LLMs exhibit directional motion blindness. Table 42 shows that most open-source Video-LLMs score near chance on MoDirect-SynBench and MoDirect-RealBench. The strongest baseline (Qwen3-VL-4B) reaches only 51.0% Overall Avg., and the best closed-source model (Gemini 2.5 Flash) plateaus at 47.6%.

`DeltaDirect` closes the gap in both LoRA and Full-FT regimes. Table 42 also shows that LLaVA-Video-7B w/ MoDirect-Inst reaches 69.9% Overall Avg., and LLaVA-Video-7B w/ `DeltaDirect` raises it to 76.7%. Full-FT-0.5B w/ `DeltaDirect` achieves a comparable 76.9% on a 14× smaller backbone, surpassing all 7B baselines and closed-source models.

`DeltaDirect` closes the gap in both LoRA and Full-FT regimes. LLaVA-Video-7B w/ MoDirect-Inst reaches 69.9% Overall Avg., and LLaVA-Video-7B w/ `DeltaDirect` raises it to 76.7%. Full-FT-0.5B w/ `DeltaDirect` achieves a comparable 76.9% on a 14× smaller backbone, surpassing all 7B baselines and closed-source models.

Table 41. **Ablation: backbone robustness.** `DeltaDirect` replicated across four Video-LLM backbones. `DeltaDirect` consistently improves Avg. Direction across most backbones, with the largest gain of +24 pp on Qwen3-VL.

Backbone	Variant	MoDirect-SynBench				MoDirect-RealBench			Avg. Direction
		P-Syn	C-Syn	P-Real	C-Real	SSv2	KTH	TOMATO	
LLaVA-Video-7B (Zhang et al., 2025c)	Vanilla	27.6	23.4	26.9	25.8	52.2	50.3	26.8	33.3
	MoDirect-Inst	99.7	81.7	75.7	61.8	72.3	66.1	35.7	70.4
	DeltaDirect	99.7	84.9	85.2	71.7	81.5	74.8	38.8	76.7
LLaVA-OneVision-7B (Li et al., 2025a)	Vanilla	28.3	23.4	29.9	29.3	52.2	50.3	26.8	34.3
	MoDirect-Inst	99.8	88.2	82.2	76.9	65.0	51.6	31.3	70.7
	DeltaDirect	99.6	89.6	83.2	78.3	66.6	51.5	32.0	71.6
mPLUG-Owl3-7B (Ye et al., 2025)	Vanilla	25.6	24.2	25.8	25.4	43.4	50.3	21.3	30.9
	MoDirect-Inst	100.0	99.9	93.6	89.5	94.4	91.3	37.2	86.6
	DeltaDirect	100.0	99.2	92.6	92.8	93.2	87.1	31.5	85.2
Qwen3-VL-4B (Bai et al., 2025a)	Vanilla	66.6	50.2	40.8	41.1	60.9	62.1	35.0	51.0
	MoDirect-Inst	74.6	51.8	61.6	56.7	72.9	98.8	31.2	63.9
	DeltaDirect	**99.9**	**97.8**	**86.7**	**87.0**	**92.1**	**100.0**	**51.6**	**87.9**

F.3.2. General Video Understanding Results

In Table 43, we report the performance of our Full-FT-0.5B model on general video understanding benchmarks (Li et al., 2024a; Xiao et al., 2021; Patraucean et al., 2023; Mangalam et al., 2023; Jang et al., 2017; Liu et al., 2024b; Zhang et al., 2024a; Tu et al., 2025; Hong et al., 2025). The training configuration is described in Appendix B.4.

F.4. Additional Analysis of `DeltaDirect`

The previous subsection shows that `DeltaDirect` substantially improves motion direction accuracy. We now ask *why*, tracing the effect from the projector output through the LLM using the diagnostic framework from Section 3 and Section 4.

Domain-invariant direction signals emerge at the projector output. Specifically, we train a four-way direction probe on Primitive-on-Syn and evaluate it on the OOD domains of MoDirect-SynBench (Table 44). Instruction tuning alone remains close to the vanilla model, with an OOD average of 42.2% and only 27.5% on Cutout-on-Real. With `DeltaDirect`, the OOD average increases to 68.6%, and Cutout-on-Real accuracy rises to 55.4%.

A small projector shift amplifies the OOD direction signal. The LLaVA-Video-7B projector is a two-layer MLP that maps SigLIP vision tokens into the LLM embedding space. Each row of a linear layer's weight matrix is one output channel, so we compare the MoDirect-Inst and `DeltaDirect` projectors via row-wise cosine similarity and relative magnitude difference across both layers. `DeltaDirect` preserves the direction of every projector row and rescales only its magnitude. Row-wise cosine similarity with the MoDirect-Inst projector stays above 0.99, while the relative magnitude difference is approximately 4%. This magnitude-only shift selectively amplifies the motion-direction signal on OOD domains. Table 45 shows that the direction concept vector magnitude at layer 21 rises by +20.4% on the most OOD Cutout-on-Real, while the in-domain Primitive-on-Syn stays nearly unchanged (−2.2%). This is consistent with the magnitude deficit diagnosed in Section 4.4.

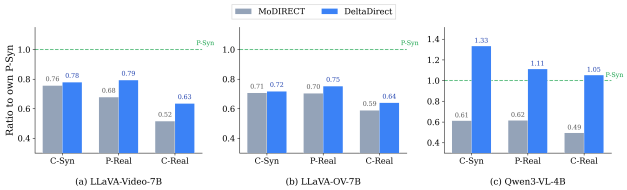


Figure 11. **`DeltaDirect` restores the OOD motion direction concept vector magnitude across video-LLM backbones.** For each backbone, we plot the direction concept vector magnitude on each OOD domain (Cutout-on-Syn, Primitive-on-Real, Cutout-on-Real) as a ratio to the same model's source-domain Primitive-on-Syn magnitude. The green dashed line marks the Primitive-on-Syn reference at 1.0. The MoDirect-Inst baseline (gray) shows a clear magnitude deficit on every backbone, while `DeltaDirect` (blue) consistently narrows the gap. Qwen3-VL-4B exhibits the largest recovery, where `DeltaDirect` pushes every OOD ratio above the Primitive-on-Syn reference.

`DeltaDirect` restores the OOD magnitude across video-LLM backbones. We apply `DeltaDirect` to LLaVA-OneVision-7B (Li et al., 2025a) and Qwen3-VL-4B (Bai et al., 2025a) to test whether `DeltaDirect` also recovers the OOD magnitude beyond LLaVA-Video-7B (Zhang et al., 2025c). Figure 11 reports the direction concept vector magnitude on each OOD domain as a ratio to the same model's source domain Primitive-on-Syn

Table 42. **`DeltaDirect` achieves state-of-the-art motion direction understanding.** We report Top-1 accuracy (%) on MoDirect-SynBench and MoDirect-RealBench. P-Syn, C-Syn, P-Real, and C-Real denote Primitive-on-Syn, Cutout-on-Syn, Primitive-on-Real, and Cutout-on-Real, respectively. All averages are macro averages over the corresponding domains or splits.

Method	MoDirect-SynBench					MoDirect-RealBench				Overall Avg.
	P-Syn	C-Syn	P-Real	C-Real	Avg.	SSv2 (Goyal et al., 2017)	KTH (Schuldt et al., 2004)	TOMATO (Shangguan et al., 2025)	Avg.	
Human	100.0	100.0	100.0	100.0	100.0	100.0	99.9	99.9	99.9	100.0
Random Chance	25.0	25.0	25.0	25.0	25.0	50.0	50.0	20.0	40.0	31.4
GPT-4o-mini (Hurst et al., 2024)	29.5	27.6	28.6	27.1	28.2	48.6	51.2	22.1	40.6	33.5
GPT-4o (Hurst et al., 2024)	46.6	37.8	48.0	40.7	43.3	51.0	52.3	34.5	45.9	44.4
Gemini 2.5 Flash (Comanici et al., 2025)	58.1	43.4	61.7	50.6	53.5	20.0	69.4	25.6	38.3	47.0
Video-LLaVA-7B (Lin et al., 2024)	28.1	27.8	25.5	27.2	27.2	49.2	49.5	14.6	37.8	31.7
LLaVA-OneVision-SI-7B	24.4	23.4	26.2	26.6	25.2	52.5	47.5	21.6	40.5	31.7
VideoChat2-HD-7B (Li et al., 2024a)	26.1	23.5	24.6	24.1	24.6	50.0	54.2	20.1	41.4	31.8
LLaMA-VID-7B (Li et al., 2024c)	25.3	25.6	24.7	25.0	25.2	51.4	54.2	16.9	40.8	31.9
LLaVA-NeXT-Video-7B	25.8	24.8	25.0	25.2	25.2	50.8	52.4	21.1	41.4	32.2
LLaVA-OneVision-7B (Li et al., 2025a)	28.3	23.4	29.9	29.3	27.7	52.2	50.3	26.8	43.1	34.3
VideoLLaMA3-2B	31.2	27.5	28.1	28.3	28.8	46.1	49.4	21.8	39.1	33.2
Qwen2.5-VL-7B (Bai et al., 2025b)	45.1	32.5	30.8	30.2	34.7	52.6	44.2	25.3	40.7	37.2
Qwen3-VL-4B (Bai et al., 2025a)	66.6	50.2	40.8	41.1	49.7	60.9	62.1	35.0	52.7	51.0
InternVL-2.5-4B (Chen et al., 2024b)	31.9	31.8	30.4	31.4	31.4	66.1	50.7	26.1	47.6	38.3
mPLUG-Owl3-7B (Ye et al., 2025)	25.6	24.2	25.9	25.8	25.4	43.4	50.3	21.3	38.3	30.9
VideoLLaMA3-7B (Zhang et al., 2025a)	56.4	46.6	50.0	48.3	50.3	56.5	52.2	19.9	42.9	47.1
LLaVA-Video-7B (Zhang et al., 2025c)	27.6	23.4	26.9	25.8	25.9	52.2	50.3	26.8	43.1	33.3
LLaVA-Video-7B w/ FlashVID (Fan et al., 2026)	25.2	25.1	24.3	24.8	24.9	52.5	50.3	21.8	41.5	32.0
LLaVA-Video-7B w/ MoDirect-Inst	99.5	80.7	74.7	60.5	78.9	72.4	66.6	35.2	58.1	69.9
LLaVA-Video-7B w/ `DeltaDirect`	**99.7**	**84.9**	**85.2**	**71.7**	**85.4**	**81.5**	**74.8**	**38.8**	**65.0**	**76.7**
Full-FT-0.5B w/ VideoChat2 (Li et al., 2024a)	26.1	23.5	24.6	24.1	24.6	53.4	54.2	19.6	42.4	32.2
Full-FT-0.5B w/ VideoChat2 + MoDirect-Inst	99.5	97.3	62.5	51.5	77.7	59.0	55.2	20.6	44.9	63.7
Full-FT-0.5B w/ `DeltaDirect`	**99.7**	**98.7**	**91.0**	**80.1**	**92.4**	**77.8**	**69.4**	**21.7**	**56.3**	**76.9**

Table 43. **`DeltaDirect` preserves general video understanding.** We compare `DeltaDirect` with the base LLaVA-Video-7B on standard and fine-grained video benchmarks. EgoSchema uses the validation set; Vinoground reports the Group score.

Method	MVBench (Li et al., 2024a)	NExT-QA (Xiao et al., 2021)	PerpTest (Patraucean et al., 2023)	EgoSchema (Mangalam et al., 2023)	TGIF-QA (Jang et al., 2017)	Avg.
LLaVA-Video-7B (Zhang et al., 2025c)	59.5	82.7	64.6	59.4	80.7	69.4
LLaVA-Video-7B w/ `DeltaDirect`	**60.7**	82.1	**65.9**	**61.2**	80.4	**70.1**
Full-FT-0.5B w/ videochat2-it (Li et al., 2024a)	48.3	66.9	48.2	**47.6**	80.2	58.4
Full-FT-0.5B w/ videochat2-it + MoDirect-Inst	47.4	67.8	48.4	47.2	**82.7**	58.7
Full-FT-0.5B w/ `DeltaDirect`	**48.9**	**69.3**	**49.2**	**47.6**	**82.7**	**59.4**

Method	TempCompo. (Liu et al., 2024b)	Vinoground (Zhang et al., 2024a)	FAVOR (Tu et al., 2025)	MotionBench (Hong et al., 2025)	Avg.
LLaVA-Video-7B (Zhang et al., 2025c)	70.5	15.6	47.2	56.0	47.3
LLaVA-Video-7B w/ `DeltaDirect`	**73.4**	**16.4**	**48.2**	**56.8**	**48.7**
Full-FT-0.5B w/ VideoChat2 (Li et al., 2024a)	44.1	**7.0**	**32.6**	31.2	28.7
Full-FT-0.5B w/ VideoChat2 + MoDirect-Inst	56.3	5.8	32.0	37.1	32.8
Full-FT-0.5B w/ `DeltaDirect`	**57.2**	6.8	30.1	**39.1**	**33.3**

Table 44. **`DeltaDirect` improves cross-domain direction probing at the projector output.** We train a direction probe on Primitive-on-Syn projector outputs and evaluate out of domain. OOD Avg. is the macro average over C-Syn, P-Real, and C-Real.

Model	OOD Avg.	C-Real
LLaVA-Video-7B	42.8	28.5
LLaVA-Video-7B w/ MoDirect-Inst	42.2	27.5
LLaVA-Video-7B w/ `DeltaDirect`	**68.6**	**55.4**

magnitude. The MoDirect-Inst baseline exhibits a magnitude deficit on every backbone, with OOD ratios falling between 0.49 and 0.76. `DeltaDirect` consistently narrows this gap on all three models. On LLaVA-Video-7B, `DeltaDirect` raises the OOD direction-vector magnitude by +16% on Primitive-on-Real and +20% on Cutout-on-Real. LLaVA-OneVision-7B shows the same trend with modest gains of +7% on Primitive-on-Real and +9% on Cutout-on-Real. Qwen3-VL-4B exhibits the largest recovery, where `DeltaDirect` more than doubles the OOD magnitude (over +100% on Cutout-on-Syn and Cutout-on-Real) and pushes every OOD ratio above the in-domain Primitive-on-Syn reference. This consistent recovery across architectures supports our claim that the OOD failure reflects a magnitude deficit rather than a missing direction.

Table 45. **`DeltaDirect` selectively amplifies the direction concept vector magnitude on OOD domains.** Direction concept-vector magnitude $\|\mathbf{v}_d^{\ell=21}\|$ at layer 21 under MoDirect-Inst and `DeltaDirect`. The relative change $\Delta(\%)$ grows from near-zero on the in-domain Primitive-on-Syn to +20.4% on the most OOD Cutout-on-Real.

Domain	MoDirect-Inst	`DeltaDirect`	$\Delta(\%)$
Primitive-on-Syn (IND)	28.92	28.29	−2.2
Cutout-on-Syn	20.49	20.85	+1.8
Primitive-on-Real	18.47	21.37	+15.7
Cutout-on-Real (OOD)	14.15	17.03	+20.4

G. Case Study

Unlike multiple-choice evaluation, open-ended video description requires the model to jointly capture object identity, motion dynamics, and temporal relationships without explicit answer options.

G.1. Direction-aware Motion Understanding

We present additional qualitative examples demonstrating how direction supervision improves motion-aware video understanding under open-ended generation settings. Unlike multiple-choice evaluation, open-ended video description requires the model to jointly capture object identity, motion dynamics, and temporal progression without explicit answer candidates.

Figure 18 shows an example from Something-Something v2. Given the prompt "Describe the video in detail," the baseline LLaVA-Video produces a partially wrong description, incorrectly interpreting the manipulated object as a "pen" or "marker" and describing a nonexistent "ink trail" on the paper. Although the model recognizes that motion occurs, it describes the movement only vaguely as a "sweeping motion" without explicitly grounding the motion direction or trajectory. In contrast, `DeltaDirect` generates a more spatially and temporally grounded description by explicitly identifying the leftward movement of the object before the final placement event on the paper. This example suggests that direction supervision improves not only directional prediction itself, but also motion grounding in free-form video generation.

Figure 19 presents an additional example from TOMATO under the prompt "Share a detailed breakdown of the video's storyline and visuals." While the baseline model summarizes the activity only coarsely as a sequence of generic hand gestures followed by a final arm extension, `DeltaDirect` produces a more fine-grained motion description by explicitly identifying intermediate directional transitions, including leftward and rightward hand movements before the final extension. Rather than compressing the action into a single high-level summary, the generated response decomposes the motion into sequential motion events that more closely align with the observed video dynamics.

Overall, these examples suggest that direction supervision encourages more precise motion-aware descriptions, improving the grounding of motion trajectories and directional transitions in open-ended video understanding.

G.2. Failure Cases in Open-ended Generation

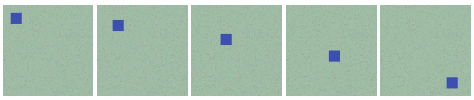
Although direction supervision often improves motion-aware generation, it can sometimes encourage the model to produce overly specific motion descriptions even when the visual evidence is ambiguous. Figure 20 shows an example from YouCook2 under the prompt "How does the hand interact with the sandwich in the video?" The baseline LLaVA-Video generates a simple but visually grounded response describing the placement of cheese slices on the sandwich. In contrast, `DeltaDirect` additionally introduces the phrase "moving in a clockwise direction," even though the observed hand movement does not clearly exhibit a clockwise trajectory. This example suggests that explicit motion supervision may increase the tendency to generate detailed motion descriptions, occasionally leading to unsupported or inaccurate motion interpretations in open-ended video understanding settings.

H. Limitations

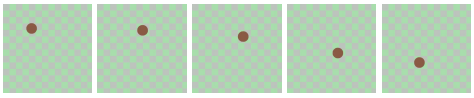
This work focuses on signed 2-D image-plane motion direction, a basic but limited form of motion understanding. Our benchmarks and supervision do not cover depth motion, rotation, acceleration, non-rigid deformation, multi-object interaction, or long-horizon event dynamics. Moreover, `DeltaDirect` uses synthetic videos where the 2-D motion vector is analytically available; extending the same supervision to arbitrary real videos may require pseudo-labels from tracking, optical flow, or human annotation. Our controlled synthetic benchmark assumes a single dominant moving object and a clear viewer-centric direction label, whereas real videos may contain camera motion, occlusion, or multiple objects moving in different directions. Finally, our main analysis uses LoRA post-finetuning as the primary training regime. We include from-scratch full finetuning results to verify that the benefit of `DeltaDirect` is not limited to adapter-based training, but a broader full-training study across larger backbones and training recipes remains computationally expensive and is left for future work. `DeltaDirect` adds no inference-time computation because the MVP head is removed after training, but it incurs a small training-time overhead for the auxiliary MVP loss. Our diagnostic analyses also require additional probe training, which is used only for analysis and not for deployment.

I. Broader Impacts

This work identifies *directional motion blindness* in Video-LLMs: motion direction is linearly decodable from internal representations, yet the model fails to reliably bind it to the correct verbal response. Our diagnosis reveals that this failure stems from a direction binding gap rather than from missing visual perception. We show that `DeltaDirect` improves motion direction understanding on both synthetic and real-world videos, which may benefit applications that require fine-grained motion understanding, e.g., , robotics, autonomous navigation, and assistive technologies. Our results suggest that improving Video-LLMs may require diagnosis-driven supervision beyond scaling models or datasets. However, `DeltaDirect` relies on synthetic motion supervision with simplified dynamics and controlled trajectories, which may not fully capture the complexity of real-world motion. Improved motion understanding capabilities could also be applied to surveillance or tracking systems. We encourage future work on responsible deployment and evaluation of such technologies.



(a) MODIRECT-INST Perturbed Linear motion example



(b) MODIRECT-INST Roundtrip motion example

Figure 12. Examples of MODIRECT-INST.

(a) Something-Something v2

(b) KTH

(c) TOMATO (Direction-Object)

(d) TOMATO (Direction-Simulated)

Figure 14. Examples of MODIRECT-REALBENCH.

(a) Primitive-on-Syn

(b) Cutout-on-Syn

(c) Primitive-on-Real

(d) Cutout-on-Real

Figure 13. Examples of MODIRECT-SYNBENCH.

Temporal

Visual Prompting: **Plain**

[Task] Watch the video carefully and identify the direction of movement of the main object.
Let's think **step by step**.
First, describe what object is moving in the video.
Second, describe where the object is located at the beginning of the video.
Third, describe where the object is located at the end of the video.
Finally, based on the position change, select the correct answer.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, in which direction is the object moving in this video?
A. Left B. Right

Visual Prompting: **Color Edge**

[Task] Watch the video carefully. The video has colored borders on each edge: Red on top, Blue on bottom, Green on left, Yellow on right. Identify which colored border the main object is moving toward.
Let's think **step by step**.
Step 1: Describe which colored border the object is closest to at the beginning of the video.
Step 2: Describe which colored border the object is closest to at the end of the video.
Step 3: Based on the position change, determine which colored border the object is moving toward and select the correct answer.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, which colored edge does the object move toward?
A. Yellow B. Green

Visual Prompting: **Text Edge**

[Task] Watch the video carefully. The video has directional text labels on each edge: 'Up' on top, 'Down' on bottom, 'Left' on the left, 'Right' on the right. Identify which text label the main object is moving toward.
Let's think **step by step**.
Step 1: Describe which text label the object is closest to at the beginning of the video.
Step 2: Describe which text label the object is closest to at the end of the video.
Step 3: Based on the position change, determine which text label the object is moving toward and select the correct answer.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, which text label does the object move toward?
A. Left B. Right

Figure 16. **Temporal Prompting Example.**

Default

Visual Prompting: **Plain**

[Question] From the viewer's perspective, which colored edge does the object move toward?
A. Green B. Yellow

Visual Prompting: **Color Edge**

[Question] From the viewer's perspective, in which direction is the object moving in this video?
A. Right B. Left

Visual Prompting: **Text Edge**

[Question] From the viewer's perspective, which colored edge does the object move toward?
A. Green B. Yellow

Figure 15. **Default Prompting Example**

Something-Something v2

User Prompt	Describe the video in detail.
LLaVA-Video	The video opens with a close-up of a hand holding a small, dark-colored object, which appears to be a pen or marker. The hand is positioned over a blank white sheet of paper placed on a bed with a striped pattern and floral designs. The hand moves the object in a sweeping motion across the paper, leaving a trail of ink behind. The background includes a pillow with a brown cover and another pillow with an orange cover. The video concludes with the hand moving away from the paper, leaving the ink trail visible on the paper.
DeltaDirect	The video features a close-up view of a hand holding a blue USB flash drive. The background consists of a bed with a striped and floral-patterned bedspread. The hand moves the USB flash drive to the left around the paper, then places it on the paper. The video concludes with the USB flash drive resting on the paper.

Figure 18. **Qualitative comparison on Something-Something v2 with an open-ended description prompt.** Compared to the baseline LLaVA-Video, `DeltaDirect` generates a more grounded description of both the object and its leftward motion. Blue text highlights motion- and direction-related expressions.

Grid Prompt

Visual Prompting: **Plain**

[Task] Watch the video carefully and identify the direction of movement of the main object.
[Grid] 8x8, [0,0]=bottom-left, [7,7]=top-right. right=+x, left=-x, up=+y, down=-y.
Let's think step by step.
Step 1: Estimate the object's start positions on the grid.
Step 2: Estimate the object's end positions on the grid.
Step 3: Based on coordinate change, determine direction.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, which text label does the object move toward?
A. Left B. Right

Visual Prompting: **Color Edge**

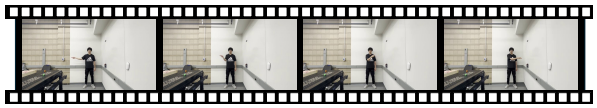
[Task] Watch the video carefully and identify the direction of movement of the main object.
[Grid] 8x8, [0,0]=bottom-left, [7,7]=top-right. right=+x, left=-x, up=+y, down=-y.
Let's think step by step.
Step 1: Estimate the object's start positions on the grid.
Step 2: Estimate the object's end positions on the grid.
Step 3: Based on coordinate change, determine direction.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, which colored edge does the object move toward?
A. Yellow B. Green

Visual Prompting: **Text Edge**

Task] Watch the video carefully and identify the direction of movement of the main object.
[Grid] 8x8, [0,0]=bottom-left, [7,7]=top-right. right=+x, left=-x, up=+y, down=-y.
Let's think step by step.
Step 1: Estimate the object's start positions on the grid.
Step 2: Estimate the object's end positions on the grid.
Step 3: Based on coordinate change, determine direction.
End your response with: Therefore, the answer is <your choice>.
[Question] From the viewer's perspective, in which direction is the object moving in this video?
A. Left B. Right

Figure 17. **Grid Prompting Example.**

TOMATO



User Prompt	Share a detailed breakdown of the video's storyline and visuals.
LLaVA-Video	The video features a person standing in an indoor setting with a plain white wall and a gray floor. The individual is dressed in a black Adidas t-shirt, black cargo pants, and white sneakers. Throughout the video, the person performs a series of hand gestures, starting with a raised right hand, followed by a waving motion, then a pointing gesture, and finally extending their right arm horizontally to the side. The lighting in the room is even, and there are no other objects or people visible in the frames.
DeltaDirect	The video features a person standing in an indoor setting with a plain background. The individual is dressed in a black T-shirt with a white Adidas logo, black cargo pants, and white sneakers. Throughout the video, the person performs a sequence of hand movements. Initially, the right hand is raised with the palm facing outward. As the video progresses, the hand moves left, then right, and finally extends to the side with the palm facing forward.

Figure 19. **Qualitative comparison on Something-Something v2 with an open-ended description prompt.** Compared to the baseline LLaVA-Video, `DeltaDirect` generates a more grounded description of both the object and its leftward motion. Red text highlights motion- and direction-related expressions.

YouCook2

User Prompt	How does the hand interact with the sandwich in the video?
LLaVA-Video	The hand is seen adding more slices of cheese to the sandwich, ensuring an even distribution.
DeltaDirect	The hand is seen adding cheese slices to the sandwich, moving in a clockwise direction.

Figure 20. **Qualitative comparison on YouCook2 under an open-ended video understanding prompt.** Compared to the baseline LLaVA-Video, `DeltaDirect` tends to generate explicit directional motion descriptions, which can sometimes be unsupported by the actual video content. Red text highlights potentially incorrect expressions.