

Diagnosing and Selecting Synthetic Motion Tasks for Video RLVR

Yishu Lei^{1*} Aofan Liu^{2*} Danxiang Zhu¹ Canming Ye²
Dan Zhang¹ Songhe Deng¹ Xianlong Luo¹ Shikun Feng^{1†}

¹ERNIE Team, Baidu ²Tsinghua University

{leiyishu, zhudanxiang, zhangdan20,
dengsonghe, luoxianlong, fengshikun01}@baidu.com
{laf24, ycm24}@mails.tsinghua.edu.cn

Abstract

Synthetic video data with verifiable rewards offers a scalable source of supervision for temporal motion understanding in video-language models, yet it remains unclear which synthetic tasks transfer reliably to real-video benchmarks. We investigate this question through a diagnostic framework that treats each synthetic task as a controlled intervention. Starting from a shared Qwen3-VL-2B checkpoint with frozen vision encoder, we independently train nine programmatic motion tasks with RLVR and evaluate each resulting model across six temporal-motion dimensions, such as direction, rotation, and velocity, yielding an interference matrix of cross-dimensional transfer.

The matrix reveals substantial variation in transfer that surface-level label alignment fails to explain: a frame-to-frame velocity task yields a large gain on its target dimension, whereas a nominally aligned trajectory-shape task degrades the dimension it is intended to improve. We find that transfer is better explained by **computational alignment**, namely whether the training task induces the local, order-sensitive temporal computation that the downstream capability requires. Counterfactual tasks that share semantic targets but differ in computational demands support this account, and the patterns hold on InternVL3-2B.

We further introduce a training-free answer-invariance test under frame shuffling that flags shortcut-prone tasks prior to RLVR training. Using the interference matrix to filter tasks yields a task mixture that consistently outperforms uniform mixing on TOMATO, with positive transfer trends on MVBench, TempCompass, and TemporalBench. Our framework thus reframes synthetic video generation as an empirically grounded task-selection problem.

1 Introduction

Video-language models have made substantial progress in visual understanding, yet fine-grained

temporal motion reasoning remains a persistent bottleneck (Shangquan et al., 2025; Li et al., 2024; Liu et al., 2024; Cai et al., 2024). Reinforcement learning with verifiable rewards (RLVR) (DeepSeek-AI, 2025; Ouyang et al., 2022) on synthetic video data offers a promising path forward: programmatic rendering provides exact, hallucination-free supervision at near-zero annotation cost. Recent studies further show that synthetic motion tasks and agentic or self-supervised RLVR pipelines can improve real-video temporal understanding (Shen et al., 2025; Zhang et al., 2025; He et al., 2025; Feng et al., 2025; Qin et al., 2026).

However, synthetic tasks are cheap to *generate* but not equally useful to *train on*. As programmatic generation scales, the central challenge shifts from obtaining supervision to selecting data: which synthetic tasks actually teach transferable temporal reasoning? A natural answer is to choose tasks whose labels appear aligned with the desired capability. We find that this surface-label strategy is unreliable. Tasks that look semantically related can induce very different learning signals, and a task with an apparently matched label can still hurt the capability it is meant to improve: for example, a trajectory-shape classification task appears aligned with trajectory reasoning, yet training on it degrades that very dimension (§4.2). This suggests that synthetic supervision should be selected by the temporal computation it induces, not merely by the words used to describe its labels.

We propose a **diagnostic approach** to synthetic video RLVR. Instead of training large mixtures directly, we treat each synthetic task as a controlled intervention and measure how it changes downstream temporal behavior. Using programmatic geometric animations, we construct nine motion tasks, each isolating a distinct temporal property, and train them independently from the same Qwen3-VL (Team, 2025) checkpoint with a frozen vision encoder. We then evaluate each resulting check-

* Equal contribution, † Corresponding author.

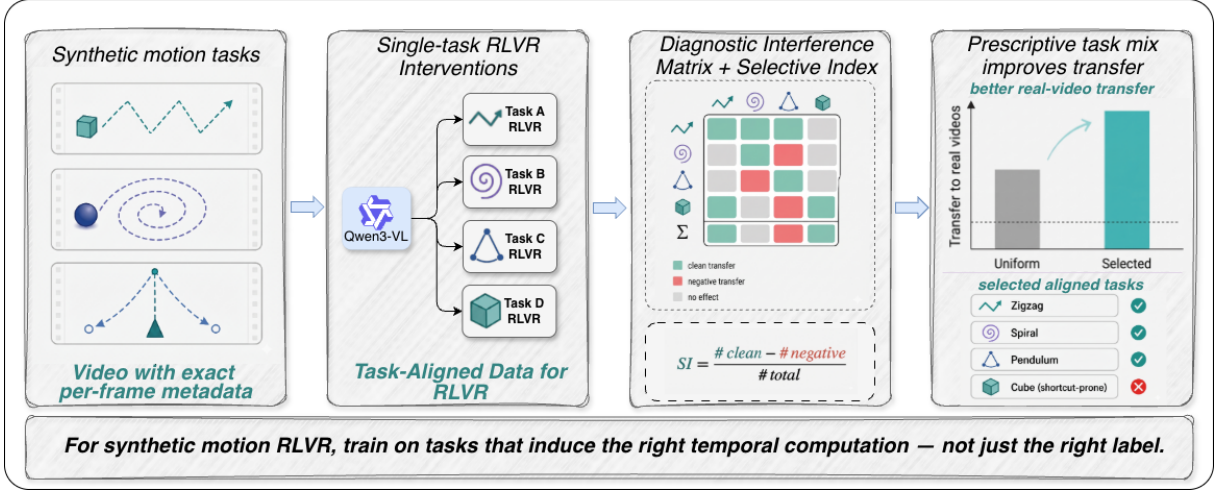


Figure 1: Overview of our diagnostic framework for synthetic video RLVR. Programmatic synthesis provides hallucination-free verifiable rewards from exact rendering parameters (top). We train candidate synthetic tasks as controlled interventions and measure their cross-dimensional transfer profiles (middle). The resulting interference matrix distinguishes tasks that improve real-video temporal understanding from shortcut-prone tasks that interfere despite surface label alignment (bottom).

point across multiple real-video temporal dimensions, yielding a 9×6 **interference matrix** that maps how each synthetic task transfers across temporal skills.

This matrix reveals a consistent pattern: synthetic tasks transfer when they induce the *precise temporal computation* required downstream, not when they merely share surface labels. We call this principle **computational alignment**. Local, order-sensitive temporal operations tend to transfer cleanly, whereas tasks solvable through global or static shortcuts often fail to help and can even interfere. We treat computational alignment as a predictive heuristic rather than a fully established causal mechanism: it reliably identifies transferable tasks in our setting, while the underlying causal pathway remains open for future study.

The diagnostic signal is also practical beyond post-hoc analysis. A zero-cost answer-invariance test can flag shortcut-prone tasks before training, and SI-guided exclusion yields a prescriptive synthetic mix that consistently outperforms uniform mixing. Thus, the interference matrix is not only an analysis tool after training, but also a recipe for selecting synthetic RLVR data before scaling.

Our contributions are as follows:

1. **A diagnostic interference matrix for synthetic video RLVR** (§4.2). We independently train nine synthetic motion tasks and measure their transfer across real-video temporal dimensions, exposing distinct transfer profiles that are invis-

ble from on-task accuracy alone.

2. **Computational alignment as a selection signal** (§4.3–4.5). We show that transfer is better predicted by whether a task requires the right temporal computation than by surface label match. This signal predicts transfer direction, holds on counterfactual tasks, and replicates on InternVL3 (Zhu et al., 2025), supporting its use as a robust empirical predictor.
3. **A practical filtering pipeline for synthetic task selection** (§4.6). We combine answer-invariance screening with SI-guided exclusion to construct a prescriptive task mix that outperforms uniform mixing across seeds. We further validate the single-task selection signal on external temporal-video benchmarks.

2 Related Work

Temporal and motion-centric video understanding benchmarks. Recent benchmarks have increasingly targeted fine-grained temporal reasoning in video-language models. TOMATO (Shangguan et al., 2025), MVBench (Li et al., 2024), TempCompass (Liu et al., 2024), and TemporalBench (Cai et al., 2024) evaluate models across temporal axes such as motion, ordering, direction, speed, and trend. These benchmarks expose failures that are often hidden by coarse video QA accuracy, showing that models can recognize objects and scenes while still struggling with how events unfold over

time. Recent work further sharpens this distinction: TimE (Wei et al., 2025) introduces multi-level temporal reasoning challenges grounded in real-world scenarios, showing that many video benchmarks can be partially solved with limited temporal understanding.

These benchmarks are primarily designed for evaluation. They reveal where temporal reasoning fails, but provide limited guidance on what kind of training data should improve a specific temporal capability. In this work, we use their temporally structured evaluation dimensions as a diagnostic testbed for synthetic RLVR: rather than asking only whether a model improves overall, we ask which synthetic motion tasks transfer to which temporal skills.

Synthetic data and RLVR for vision-language models. Synthetic data is especially attractive for temporal video learning because programmatic generation can provide exact motion parameters, controllable interventions, and verifiable rewards at low annotation cost. Recent RLVR methods typically build on GRPO (Shao et al., 2024), a critic-free variant of proximal policy optimization (Schulman et al., 2017). In the vision-language setting, works such as R1-V, VLM-R1 (Shen et al., 2025), and DPG extend verifiable-reward reinforcement learning to multimodal reasoning. For video, SynRL applies synthetic temporal primitives to RLVR and demonstrates transfer to real-video benchmarks, while Video-R1 (Feng et al., 2025), ReWatch-R1 (Zhang et al., 2025), EasyVideoR1 (Qin et al., 2026), and VideoSSR (He et al., 2025) further develop RLVR, agentic data synthesis, efficient training, and self-supervised reward construction for video reasoning.

These methods build on rapid progress in video-language foundation models, from temporal-aware visual backbones such as SlowFast (Feichtenhofer et al., 2019) and TimeSformer (Bertasius et al., 2021), to frozen-encoder vision-language pretraining (Radford et al., 2021; Li et al., 2023) and video-adapted LLMs such as VideoLLaMA 2 (Cheng et al., 2024). While prior work demonstrates promising end-to-end gains, it primarily optimizes aggregate performance. Our focus is complementary: we treat individual synthetic motion tasks as controlled interventions to diagnose which temporal computations they teach, when they transfer, and when they interfere. This diagnostic view turns synthetic RLVR from a data-generation prob-

lem into a data-selection problem. Related concerns also arise in studies of spurious rewards in RLVR (Shao et al., 2025) and reward model overoptimization (Gao et al., 2023), which show that reward signals must be designed and interpreted carefully.

Data selection, negative transfer, and task interference. Data composition has long been known to shape downstream performance (Bengio et al., 2009). DSIR (Xie et al., 2023b), Skill-It (Chen et al., 2023), and DoReMi (Xie et al., 2023a) study data selection, curriculum design, and domain weighting for language-model training. In multi-task learning, PCGrad (Yu et al., 2020), TAG (Fifty et al., 2021), and related work (Caruana, 1997) analyze task interference and negative transfer. In vision-language models, recent studies examine concept-skill transferability (Lee et al., 2024) and systematic task transfer patterns (Sachdeva et al., 2025), revealing both positive synergies and harmful interference that may be invisible from isolated on-task metrics. Multimodal post-training has also begun to address task conflicts directly, including imbalanced gradients in RL post-training (Wu et al., 2026) and LoRA-MoE methods for mitigating interference (Chen et al., 2024).

Our interference matrix follows the diagnostic spirit of Taskonomy (Zamir et al., 2019) and later task-grouping work (Standley et al., 2020), which use task-transfer measurements to guide architecture and data decisions. We adapt this idea to synthetic video RLVR, where the source tasks are programmatically generated, the rewards are verifiable, and the target capabilities are temporal-motion skills. Our key departure is to explain transfer through *computational alignment*: tasks help when the temporal computation induced by training matches the downstream requirement, and they fail when apparent label alignment hides shortcut solutions. This distinction is not captured by label-level task-skill matching alone (formalized as nominal semantic alignment in §3.2), and need not be visible from aggregate performance or gradient-conflict metrics, i.e., measures of directional disagreement between per-task gradients in multi-task training.

3 A Diagnostic Framework for Synthetic RLVR

We focus on *motion trajectory understanding*: the ability to reason about how objects move, change

Task	Question Template	Nominal Skill Axis
MOTION_DIR	Which direction?	direction
VELOCITY	Accelerating or decelerating?	speed change
ROTATION_DIR	CW or CCW?	rotation
MOTION_TYPE	What trajectory shape?	shape & trend
TREND_CHANGE	Does the trend change?	trend change
OBJECT_COUNT	How many objects?	object numerosity
ACTION_COUNT	How many repetitions?	event numerosity
TEMP_GROUND	When does motion occur?	temporal localization
SPEED_COMP	Which object is faster?	relative speed
COLLISION (neg.)	How many collisions?	—

Table 1: Synthetic training tasks and their nominal temporal skill axes. The first nine rows form the main task set; COLLISION is used only as a negative control and has no aligned skill axis.

speed, repeat, rotate, and interact over time. Our central question is not how to generate more synthetic videos, but how to decide which generated tasks are worth training on. Among the many synthetic tasks available, which ones teach transferable temporal computations, and can we identify them before expensive mixture training?

3.1 Programmatic Video Tasks

We render synthetic videos of abstract geometric objects whose trajectories are sampled from 12 motion primitives, including zigzag, spiral, pendulum, and bounce. Because each video is generated programmatically, we have exact per-frame metadata for object positions, velocities, events, and motion types. This allows us to compute ground-truth answers directly from rendering parameters, yielding fully verifiable rewards without human annotation noise.

From this shared motion vocabulary, we derive nine primary training tasks and one negative control, COLLISION. The nine primary tasks form the rows of our main interference matrix; COLLISION is included only as a negative control and is excluded from the selectivity-index (SI; §3.2) and transfer-profile analyses. Each task contains 1,000 multiple-choice samples with 3–6 answer options.

Table 1 summarizes the synthetic tasks and their *nominal temporal skill axes*. These axes are not used to tune the tasks to a specific benchmark. Instead, they provide an analysis scaffold for asking whether intuitive, label-level task–skill correspondences—which we call *nominal semantic alignment* throughout the paper—predict actual transfer. In later experiments, we instantiate these axes using structured temporal-motion evaluation dimensions.

3.2 Interference Matrix, SI, and Transfer Profiles

Interference matrix. Let $\mathcal{P} = \{p_1, \dots, p_P\}$ denote the set of synthetic training tasks and $\mathcal{D} = \{d_1, \dots, d_D\}$ the set of evaluation dimensions. We define the **interference matrix** $\mathbf{M} \in \mathbb{R}^{P \times D}$ as

$$M_{ij} = s(d_j; \theta_{p_i}) - s(d_j; \theta_0), \quad (1)$$

where $s(d_j; \theta)$ denotes accuracy on evaluation dimension d_j under model parameters θ , θ_0 is the shared base checkpoint, and θ_{p_i} is the checkpoint after RLVR training on task p_i .

Each row of $\mathbf{M}$ measures how training on a single synthetic task changes performance across a set of temporal-motion evaluation dimensions. To analyze whether nominal semantic alignment is predictive, we define a nominal mapping $\tau : \mathcal{P} \rightarrow 2^{\mathcal{D}}$ from synthetic tasks to aligned evaluation dimensions. We fixed τ before inspecting any transfer outcomes, by directly matching each task’s question template and nominal skill axis (Table 1) to the corresponding evaluation-dimension definition; it was not revised afterwards to fit the matrix. This mapping is used only for analysis: it partitions each row into aligned entries ($\mathbf{m}_i^{\text{align}}$) and non-aligned entries ($\mathbf{m}_i^{\text{other}}$), rather than serving as a training objective or selection rule.

Selectivity Index (SI). For tasks with at least one aligned evaluation dimension, we define the **selectivity index** as

$$\text{SI}(p_i) = \frac{\bar{M}_i^{\text{align}}}{\sqrt{\frac{1}{D - |\tau(p_i)|} \sum_{j: d_j \notin \tau(p_i)} M_{ij}^2} + \epsilon}, \quad (2)$$

where $\bar{M}_i^{\text{align}}$ is the average improvement on aligned dimensions and $\epsilon = 0.01$ is a small constant for numerical stability.

SI measures aligned improvement relative to unintended effects on other dimensions. Large positive values indicate selective transfer, values near zero indicate weak or diffuse effects, and negative values indicate that a task harms its nominally aligned capability. SI is left undefined for tasks without aligned dimensions, such as TEMP_GROUND and COLLISION.

Transfer profile classification. We categorize each task according to its aligned responsiveness $R(p_i) = \bar{M}_i^{\text{align}}$ and its effects on non-aligned dimensions, using a threshold of $\delta = 2$ percentage

points:

$$\text{Profile}(p_i) = \begin{cases} \text{NT} & R \leq 0 \\ \text{WP} & 0 < R \leq \delta \\ \text{CP} & R > \delta, \exists j \notin \tau(p_i): M_{ij} < -\delta \\ \text{BS} & R > \delta, \exists j \notin \tau(p_i): M_{ij} > \delta \\ \text{CL} & R > \delta, \max_{j \notin \tau(p_i)} (-M_{ij}) < \delta \end{cases} \quad (3)$$

Here, NT, WP, CP, BS, and CL denote No Transfer, Weak Positive, Competing, Bonus, and Clean profiles, respectively. When multiple conditions apply, categories are assigned in the order listed. The threshold $\delta = 2$ pp was fixed before inspecting task-level transfer outcomes, as a simple, interpretable cutoff that avoids assigning substantive profile labels to sub-2-pp changes; it affects only this descriptive taxonomy, not the matrix itself, the SI values or ranking, or the mixture-selection rule.

This taxonomy separates selective gains from harmful interference, incidental bonus effects, and outright failures. These distinctions are invisible from on-task training accuracy and can also be obscured by aggregate downstream scores.

4 Experiments

4.1 Experimental Setup

Training protocol. Each synthetic task is trained as an independent RLVR intervention from the same Qwen3-VL-2B checkpoint. We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with a format reward (weight 0.3) and an exact-match answer reward (weight 1.0), training for 500 steps with 8 rollout generations per sample on 8×H800 GPUs. The vision encoder is frozen throughout, and only the LLM backbone is updated. This shared-initialization, single-factor design ensures that downstream differences reflect the synthetic task intervention rather than initialization, visual encoder drift, or mixture artifacts. Both training and evaluation use a multiple-choice format; Appendix A.2 specifies the prompt format, gives a worked example, and discusses potential format bias.

Evaluation. We first evaluate each task-trained checkpoint on a structured temporal-motion evaluation suite and report Δ accuracy in percentage points relative to the shared base model. The primary suite is the full TOMATO benchmark (Shang-guan et al., 2025), evaluated along its six temporal-motion dimensions: count, direction, rotation, shape & trend, velocity & frequency, and visual

	cnt	dir	rot	s&t	v&f	vis	Mean
Base accuracy	25.68	34.24	23.43	28.25	20.48	32.86	27.49

Table 2: Shared base checkpoint (pre-RLVR Qwen3-VL-2B) accuracy on the six TOMATO temporal-motion dimensions under identical decoding and evaluation settings. Column abbreviations, used throughout the paper, denote count (cnt), direction (dir), rotation (rot), shape & trend (s&t), velocity & frequency (v&f), and visual cues (vis).

cues. Table 2 reports the shared base checkpoint’s accuracy on each dimension; no dimension is near ceiling. This produces one transfer row per synthetic task and forms the interference matrix. To test whether the resulting diagnostic patterns extend beyond the primary evaluation suite, we additionally evaluate selected checkpoints on MVBench, TempCompass, and TemporalBench.

Derivation-validation split. We partition the synthetic tasks into a *derivation set* for forming the computational-alignment hypothesis and a *validation set* for testing it. Held-out validation tasks share coarse semantic targets with derivation tasks but require different computations, enabling directional tests of whether computation, rather than label match, predicts transfer. For example, TREND_CHANGE was designed after observing the failure of MOTION_TYPE: it keeps the same broad trajectory-related target but replaces global template matching with local transition detection.

4.2 The Interference Matrix Reveals Inconsistent Transfer

Figure 2 visualizes the full 9×6 interference matrix. Each row shows how one synthetic RLVR intervention changes performance across temporal-motion evaluation dimensions. Table 3 summarizes the corresponding selectivity indices and transfer profiles, with derivation and validation tasks separated.

Some synthetic tasks transfer cleanly. VELOCITY yields a large gain on speed-related reasoning (+25.71 pp, SI= 17.81), and ROTATION_DIR improves rotation selectively (+5.59 pp, SI= 5.77). Both tasks require local, order-sensitive temporal computations, suggesting that synthetic RLVR can teach transferable motion reasoning when the practiced computation matches the evaluated capability.

Nominal semantic alignment is not sufficient. By contrast, MOTION_TYPE is nominally aligned

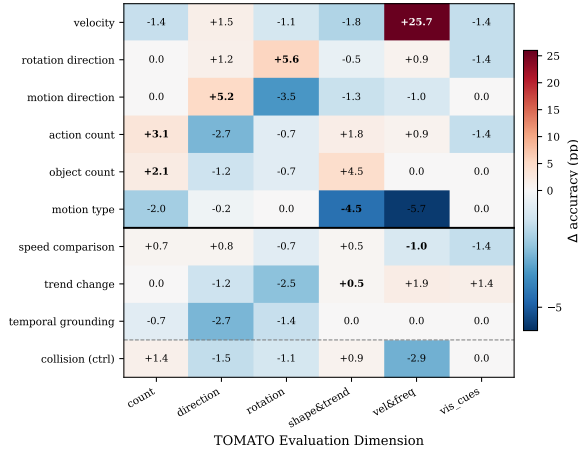


Figure 2: Full 9×6 interference matrix heatmap. Rows are grouped into derivation tasks used to build the framework, held-out validation tasks (SPEED_COMP, TREND_CHANGE, TEMP_GROUND) used to test predictions in §4.4, and the COLLISION negative control. Color intensity indicates Δ pp relative to the shared base checkpoint, i.e., the common pre-RLVR Qwen3-VL-2B model from which every task-specific intervention starts.

with shape & trend, yet it degrades its aligned evaluation dimension (-4.48 pp) and also harms speed-related reasoning (-5.72 pp). The task admits a global template-matching shortcut (Geirhos et al., 2020), reinforcing a computation that bypasses the temporal reasoning required downstream. This failure shows why nominal semantic alignment is an unreliable selection rule.

Transfer is cross-dimensional. Other tasks reveal that synthetic RLVR affects multiple temporal skills at once. MOTION_DIR improves direction but degrades rotation, yielding a competing profile. OBJECT_COUNT improves counting and unexpectedly benefits trajectory-related reasoning, exposing a bonus route through spatial grounding. These interactions motivate the full interference matrix: on-task accuracy or aligned-dimension accuracy alone would miss both harmful interference and incidental positive transfer.

4.3 Computational Alignment Predicts Transfer

The interference matrix suggests that transfer is governed less by nominal semantic alignment than by the computation a task induces during training. We call this relation **computational alignment**: a synthetic task is aligned with an evaluation dimension when solving the task exercises the operation

Task	Axis	$\bar{M}_{align}$	SI	Profile
<i>Derivation set (6 tasks)</i>				
VELOCITY	speed change	+25.71	17.81	CL
ROTATION_DIR	rotation	+5.59	5.77	CL
MOTION_DIR	direction	+5.21	3.01	CP
ACTION_CNT	event count	+3.09	1.84	CP
OBJECT_CNT	object count	+2.06	0.98	BS
MOTION_TYPE	shape & trend	-4.48	-1.65	NT
<i>Validation set (3 held-out tasks; §4.4)</i>				
SPEED_COMP	rel. speed	-0.96	-1.10	NT
TREND_CHG	trend change	+0.45	0.28	WP
TEMP_GROUND*	—	—	—	—

Table 3: Transfer profiles. Profile codes follow Eq. 3: NT (No Transfer), WP (Weak Positive), CP (Competing), BS (Bonus), CL (Clean). The derivation set exposes the major observed profile types; the validation set tests predictions derived from the framework. *No aligned evaluation dimension; SI undefined.

that the dimension tests. This separates two notions that are often conflated. A task may be semantically related to a capability but computationally misaligned if it admits a shortcut; conversely, a task may transfer beyond its nominal label when it practices a useful underlying operation.

Table 4 summarizes the computational structure of the synthetic tasks. Clean-transfer tasks, such as VELOCITY and ROTATION_DIR, require local comparisons over ordered frames. In contrast, MOTION_TYPE can be solved by global template matching and produces negative transfer despite its apparent relevance to trajectory reasoning. The positive transfer of OBJECT_COUNT is a useful boundary case: it is not temporally order-sensitive, but it improves spatial grounding, which can support some trajectory-related evaluations. Thus, computational alignment is not equivalent to temporal order sensitivity alone; it asks whether the operation practiced by the synthetic task is useful for the evaluated capability.

We further quantify computational alignment with two task–dimension scores. *Temporal necessity* (TNS) measures whether the answer depends on frame order. *Skill alignment score* (SAS) measures whether the operation practiced by the synthetic task overlaps with the operation required by the evaluation dimension. Their product, $TNS \times SAS$, serves as a simple proxy for computational alignment.

This scoring rule recovers the qualitative pattern observed in the matrix. Tasks that combine high skill overlap with a genuine need for temporal ordering tend to transfer, while tasks miss-

Task	Computation	Grain	OSS	Profile	SI
<i>Derivation set</i>					
VELOCITY	Frame-pair Δv	Local	+5.7	CL	17.8
ROT. DIR	Sequential angle	Local	+1.7	CL	5.8
MOT. DIR	Movement trend	Semi	+2.2	CP	3.0
ACT. CNT	Repetition segments	Semi	0.0	CP	1.8
OBJ. CNT	Spatial counting	Frame	-1.4	BS	1.0
MOT. TYPE	Template matching	Global	-2.2	NT	-1.7
<i>Validation set (predictions in §4.4)</i>					
SPEED_C.	Displacement average	Global	-1.0	NT	-1.1
TREND_CHG	Transition detection	Local	+1.3	WP	0.3

Table 4: Computational structure of the synthetic tasks. OSS denotes the differential frame-shuffle accuracy drop after training (trained – baseline; §4.6); positive values indicate increased reliance on temporal order. Profile codes follow Eq. 3 (Table 3). The derivation set is used to formulate the alignment hypothesis, while the validation set is reserved for directional tests in §4.4.

ing either condition are unreliable. This explains why MOTION_TYPE fails despite its trajectory-related label: its answers are largely obtainable through global template matching. It also explains why SPEED_COMP does not behave like VELOCITY: both concern speed, but only VELOCITY requires local frame-pair comparison. Conversely, TREND_CHANGE avoids the shortcut in MOTION_TYPE by requiring local transition detection. Quantitatively, TNS×SAS correlates with observed transfer across the analyzed task–dimension pairs ($\rho = 0.70$, $p = 0.008$; details in Appendix A.5).

We use computational alignment as a predictive diagnostic rather than a complete causal account. SAS is assigned from task definitions before observing transfer outcomes, and Appendix A.5 (Table 11) shows that the correlation remains stable under score perturbations and an independent LLM-based annotation protocol. This makes TNS×SAS useful for prospective task selection, while leaving the underlying mechanism for future work.

4.4 Held-Out and Prospective Validation

A stronger test is whether computational alignment predicts outcomes for tasks not used to derive the framework. We evaluate three held-out tasks that preserve coarse semantic labels while changing the induced computation, yielding falsifiable directional predictions.

Held-out counterfactual tasks. SPEED_COMP targets relative speed like VELOCITY, but it can be solved by global displacement averaging (TNS=0). Computational alignment predicts weak or

Task	Predicted	Target Δ	SI	Result
RHYTHM_CNT	SI > 0	+2.66	+1.36	✓

Table 5: Prospective validation. The prediction is made from task computation *before* RLVR training.

negative transfer, which is confirmed by its negative outcome (-0.96 pp, SI= -1.10). Conversely, TREND_CHANGE targets trajectory dynamics like MOTION_TYPE, but requires local transition detection (TNS= 1.0). The framework predicts non-negative transfer, and the task indeed reverses the direction of MOTION_TYPE’s failure ($+0.45$ pp, SI= 0.28). Finally, TEMP_GROUND serves as a held-out boundary control: it is a genuinely temporal task, but our six-axis evaluation suite contains no temporal-localization dimension, so it has no aligned evaluation dimension and its SI is undefined by construction (§3.2). It tests whether a temporal task without a matched evaluation target produces incidental cross-dimensional transfer; as predicted, it produces no consistent gain (mean $\Delta = -0.3$ pp). Its later exclusion from the prescriptive mix (§4.6) follows from the selection rule requiring a measurable aligned target and positive SI, not from this negative mean.

Prospective task design. We further design RHYTHM_COUNTING (“How many tempo changes?”) after the framework is fixed. Although it has little lexical overlap with the original VELOCITY task, it requires local, order-sensitive temporal computation. Computational alignment predicts positive transfer to speed-related reasoning, which is confirmed after RLVR training ($+2.66$ pp, SI= 1.36; Table 5).

4.5 Generalization and Robustness

Controls and stability. We rule out two trivial explanations: that any synthetic RLVR run improves temporal-motion performance, and that arbitrary verifiable labels suffice. The COLLISION negative control shows no consistent improvement (mean -0.52 pp), and randomized-label training likewise fails (mean -0.30 pp). Stability checks show that changes on visually biased control dimensions remain within $[-1.43, 0]$ pp, and target-dimension standard deviations across three seeds are small (1.3–2.6 pp).

Cross-architecture replication. To test whether the diagnostic patterns are architecture-specific, we

Task	cnt	dir	rot	s&t	v&f	vis
VELOCITY	0.0	-1.5	0.0	+0.9	+28.1	+1.4
MOTION_DIR	+1.7	+2.0	+0.7	+1.4	-1.9	-1.4
MOTION_TYPE	-1.4	+4.0	+1.1	-4.0	-4.3	0.0
Mix (v+d)	+1.0	+2.2	0.0	+1.8	+22.4	+1.4

Table 6: Partial cross-architecture replication on InternVL3-2B. Values are Δ accuracy relative to the InternVL3 baseline. Column abbreviations as in Table 2.

replicate representative cases on InternVL3-2B-Instruct (Zhu et al., 2025) (Table 6). VELOCITY again yields a large gain on speed-related reasoning (+28.1 pp), while MOTION_TYPE again degrades trajectory-related reasoning (-4.0 pp) and speed-related reasoning (-4.3 pp). This suggests that computational alignment reflects a task-capability relation rather than an artifact of a single base model.

Model-scale check. Appendix A.4 reports a single-seed check in which the top-3 SI mixture derived from the 2B matrix is applied to Qwen3-VL-8B-Instruct without reconstructing a full 8B interference matrix. The macro average improves by +6.89 pp on the 8B model, with non-negative changes on all six dimensions, providing initial evidence that 2B-derived task selection can transfer to a larger scale.

External benchmark transfer. We next ask whether profiles derived from structured temporal-motion diagnostics provide directional evidence beyond the primary evaluation suite. Table 7 evaluates selected single-task checkpoints on MVBench, TempCompass, and TemporalBench. The absolute gains are modest (≤ 1.1 pp) and likely within evaluation noise for individual benchmarks, so we interpret them cautiously. Still, the ordering is consistent with the interference matrix: the highest-SI task transfers positively to all three external benchmarks, while the lowest-SI task is positive on only one.

4.6 From Diagnosis to Data Selection

We finally ask whether the interference matrix can guide mixture construction. All filtering decisions are fixed before mixture training and before external evaluation. The procedure is simple: design candidate tasks, screen for answer-invariance shortcuts, train the remaining tasks independently, compute SI, and exclude tasks with negative or

Task	SI	Prim.	MVB	TC	TB	Pos.
VELOCITY	17.81	+25.71	+0.37	+0.94	+1.12	3/3
ROTATION_DIR	5.77	+5.59	-0.21	+0.71	-2.74 [†]	1/3
MOTION_DIR	3.01	+5.21	+0.84	-0.15	+0.30	2/3
ACTION_CNT	1.84	+3.09	+0.77	-0.77	+0.12	2/3
TREND_CHG	0.28	+0.45	+0.17	+0.73	+0.10	3/3
MOTION_TYPE	-1.65	-4.48	+0.34	-0.63	-0.15	1/3

[†]The TemporalBench drop for ROTATION_DIR is concentrated in the FineGym subset; we treat this task as benchmark-sensitive rather than uniformly transferable.

Table 7: Cross-benchmark transfer for single-task models, ordered by SI. “Prim.” is the aligned-dimension gain on the primary temporal-motion suite; MVB, TC, and TB denote MVBench, TempCompass, and TemporalBench; “Pos.” counts benchmarks with $\Delta > 0$ out of three.

Task	Inv.	SI	Task	Inv.	SI
ROTATION_DIR	0%	+5.8 ✓	OBJECT_CNT	100%	+1.0 $\times$ [†]
ACTION_CNT	0%	+1.8 ✓	SPEED_COMP	100%	-1.1 ✓
TREND_CHG	31%	+0.3 ✓	MOTION_TYPE	100%	-1.7 ✓
VELOCITY	34%	+17.8 ✓			
MOTION_DIR	49%	+3.0 ✓			

[†]OBJECT_COUNT transfers through spatial reasoning, orthogonal to temporal order.

Table 8: Answer invariance under frame shuffle vs. observed SI. Invariance is a training-free warning signal for temporal-order shortcuts, but not a complete predictor of transfer ($r = -0.42$).

undefined selectivity before constructing the final mix.

Training-free shortcut screening. Before training, practitioners can test whether a task’s ground-truth answer changes under random frame shuffling. A task whose answer is fully invariant to frame order cannot require temporal-order reasoning, making answer invariance a zero-cost warning signal for shortcut risk. This differs from OSS (Table 4): OSS measures a *trained* model’s temporal-order reliance, whereas answer invariance is a *task-level* property computable before training. Table 8 compares answer invariance with the observed selectivity of each task.

The screening rule is simple: all three tasks with 100% answer invariance have $SI \leq 1.0$, while all five tasks with less than 50% invariance have positive SI. The sole exception, OBJECT_COUNT, transfers through spatial grounding rather than temporal-order reasoning. Thus, answer invariance is a high-precision shortcut screen, but not a complete predictor of transfer.

Strategy	Primary temporal-motion evaluation						Mean
	cnt	dir	rot	s&t	v&f	vis	
Prescrip. (5)	+10.4	+3.5	+1.7	+1.8	+24.8	-1.5	+6.5 ±0.5
Uniform (8)	+7.3	+2.0	+2.3	+0.9	+15.7	+0.0	+4.4±0.7

Table 9: Mix training results (Δ pp vs. baseline, mean across $n=3$ seeds; column abbreviations as in Table 2). The prescriptive mix consistently outperforms uniform on the primary temporal-motion evaluation suite, with the largest gains on speed-related reasoning and counting; the advantage holds across all three seeds ($+2.1 \pm 1.1$ pp).

Matrix-guided mix construction. The interference matrix yields a concrete selection rule: include tasks with positive selectivity and a clear evaluation target, and exclude shortcut-prone or targetless tasks. The **uniform mix** trains on all eight tasks excluding the held-out TREND_CHANGE. The **prescriptive mix** removes three matrix-flagged tasks: MOTION_TYPE (SI= -1.65) and SPEED_COMP (SI= -1.10), both shortcut-prone, and TEMP_GROUND, which lacks a measurable aligned target. This yields five computationally aligned tasks.

Table 9 shows that the prescriptive mix outperforms uniform mixing across all three seeds. The largest gains occur on dimensions where shortcut-prone tasks previously interfered, while the mix still preserves broad coverage across direction and counting. Unlike velocity-only training, the prescriptive mix identifies a complementary task combination rather than a single high-performing task.

The advantage is compositional. Size-matched random 5-task subsets confirm that gains correlate monotonically with prescriptive overlap ($+3.9$ pp at 4/5 vs. $+1.3$ pp at 2/5), ruling out a task-count confound. This indicates that the benefit comes from the selected task composition rather than simply from using fewer tasks.

Mechanistic interpretation. Exploratory mechanistic analysis (Appendix A.1) suggests that computationally aligned tasks calibrate how the LLM backbone uses temporal information over frozen visual features: CKA divergence concentrates in upper LLM layers (14–27), linear probing shows elevated temporal content in those layers ($+10.9$ pp at layer 27 for VELOCITY; $p = 0.013$), and differential frame-shuffle sensitivity tracks alignment profiles. We treat this as a plausible interpretation rather than a complete causal account.

5 Conclusion

We introduce the interference matrix as a diagnostic methodology for synthetic video RLVR. By training single-factor tasks from a shared checkpoint and measuring cross-dimensional transfer, we expose transfer profiles invisible from on-target accuracy alone. Computational alignment—whether a task induces the temporal computation required downstream—serves as a strong empirical predictor of transfer direction within our setting (synthetic geometric video, frozen vision encoder, 2B-scale models), outperforming nominal semantic alignment. A training-free answer-invariance test flags shortcut-prone tasks at zero cost, and SI-guided filtering yields a prescriptive mix outperforming uniform mixing ($+6.5$ vs. $+4.4$ pp). Future work should establish whether this principle generalizes to unfrozen vision encoders, richer video domains, and larger model scales.

Limitations

Scope. Our study focuses on synthetic geometric motion tasks for motion-centric video understanding, with experiments on Qwen3-VL and InternVL3 backbones. The observed patterns suggest that computational alignment is a useful selection signal in this setting; a single-seed scale check on Qwen3-VL-8B (Appendix A.4) provides initial cross-scale evidence, but broader generalization to larger models, real-video training data, and non-motion reasoning tasks remains open.

Validation scale. The held-out and prospective tasks test whether computational alignment predicts directional transfer under controlled task changes. While these results support the diagnostic framework, they cover a small task pool. Stronger validation would require prospective predictions over larger synthetic task families and new video domains. We also do not systematically quantify within-task difficulty distributions, sample diversity, or every aspect of data quality, which may contribute to the observed transfer differences alongside computational alignment.

Cost and mechanism. The full pipeline requires independent single-task training before mixture construction, which may be costly for large candidate pools. Answer invariance provides a cheap first-pass filter, but future work should develop surrogate selection signals that reduce the need for exhaustive interventions. Our representational

analysis is consistent with a temporal-calibration interpretation, but it remains suggestive rather than causal.

References

- Yonatan Belinkov. 2022. [Probing classifiers: Promises, shortcomings, and advances](#). *Comput. Linguistics*, 48(1):207–219.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, ACM International Conference Proceeding Series, pages 41–48. ACM.
- Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. [Is space-time attention all you need for video understanding?](#) In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 813–824. PMLR.
- Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, Yao Dou, Jaden Park, Jianfeng Gao, Yong Jae Lee, and Jianwei Yang. 2024. [Tempo-ralbench: Benchmarking fine-grained temporal understanding for multimodal video models](#). *CoRR*, abs/2410.10818.
- Rich Caruana. 1997. [Multitask learning](#). *Mach. Learn.*, 28(1):41–75.
- Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2023. [Skill-it! A data-driven skills framework for understanding and training language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Zeren Chen, Ziqin Wang, Zhen Wang, Huayang Liu, Zhenfei Yin, Si Liu, Lu Sheng, Wanli Ouyang, and Jing Shao. 2024. [Octavius: Mitigating task interference in mllms via lora-moe](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. [Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms](#). *CoRR*, abs/2406.07476.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. [Slowfast networks for video recognition](#). In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6201–6210. IEEE.
- Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025. [Video-r1: Reinforcing video reasoning in mllms](#). *CoRR*, abs/2503.21776.
- Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. [Efficiently identifying task groupings for multi-task learning](#). In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27503–27516.
- Leo Gao, John Schulman, and Jacob Hilton. 2023. [Scaling laws for reward model overoptimization](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 10835–10866. PMLR.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. [Shortcut learning in deep neural networks](#). *Nat. Mach. Intell.*, 2(11):665–673.
- Zefeng He, Xiaoye Qu, Yafu Li, Siyuan Huang, Daizong Liu, and Yu Cheng. 2025. [Videossr: Video self-supervised reinforcement learning](#). *CoRR*, abs/2511.06281.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. [Similarity of neural network representations revisited](#). In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pages 3519–3529. PMLR.
- Jaewoo Lee, Boyang Li, and Sung Ju Hwang. 2024. [Concept-skill transferability-based data selection for large vision-language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 5060–5080. Association for Computational Linguistics.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. [BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 19730–19742. PMLR.
- Kunchang Li, Yali Wang, Yanan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. 2024.

- Mvbench: A comprehensive multi-modal video understanding benchmark. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 22195–22206. IEEE.
- Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. 2024. [Tempcompass: Do video llms really understand videos?](#) In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 8731–8772. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Chuanyu Qin, Chenxu Yang, Qingyi Si, Naibin Gu, Dingyu Yao, Zheng Lin, Peng Fu, Nan Duan, and Jiaqi Wang. 2026. [Easyvideor1: Easier RL for video understanding](#). *CoRR*, abs/2604.16893.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 8748–8763. PMLR.
- Bhuvan Sachdeva, Karan Uppal, Abhinav Java, and Vineeth N. Balasubramanian. 2025. [Understanding task transfer in vision-language models](#). *CoRR*, abs/2511.18787.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *CoRR*, abs/1707.06347.
- Ziyao Shangguan, Chuhan Li, Yuxuan Ding, Yanan Zheng, Yilun Zhao, Tesca Fitzgerald, and Arman Cohan. 2025. [TOMATO: assessing visual temporal reasoning capabilities in multimodal foundation models](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. 2025. [Spurious rewards: Rethinking training signals in RLVR](#). *CoRR*, abs/2506.10947.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. 2025. [VLM-R1: A stable and generalizable r1-style large vision-language model](#). *CoRR*, abs/2504.07615.
- Trevor Standley, Amir Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2020. [Which tasks should be learned together in multi-task learning?](#) In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, Proceedings of Machine Learning Research, pages 9120–9132. PMLR.
- Qwen Team. 2025. [Qwen3-vl technical report](#). *CoRR*, abs/2511.21631.
- Shaohang Wei, Wei Li, Feifan Song, Wen Luo, Tianyi Zhuang, Haochen Tan, Zhijiang Guo, and Houfeng Wang. 2025. [TIME: A multi-level benchmark for temporal reasoning of llms in real-world scenarios](#). *CoRR*, abs/2505.12891.
- Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. 2026. [Imbalanced gradients in RL post-training of multi-task llms](#). In *Findings of the Association for Computational Linguistics: EACL 2026, Rabat, Morocco, March 24-29, 2026*, Findings of ACL, pages 3137–3150. Association for Computational Linguistics.
- Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. [Doremi: Optimizing data mixtures speeds up language model pretraining](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023b. [Data selection for language models via importance resampling](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. [Gradient surgery for multi-task learning](#). In *Advances in Neural Information Processing Systems 33*.

Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Amir Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. 2019. **Taskonomy: Disentangling task transfer learning**. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 6241–6245. ijcai.org.

Congzhi Zhang, Zhibin Wang, Yinchao Ma, Jiawei Peng, Yihan Wang, Qiang Zhou, Jun Song, and Bo Zheng. 2025. **Rewatch-r1: Boosting complex video reasoning in large vision-language models through agentic data synthesis**. *CoRR*, abs/2509.23652.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, et al. 2025. **InternV3: Exploring advanced training and test-time recipes for open-source multimodal models**. *CoRR*, abs/2504.10479.

A Supplementary Analyses

A.1 Mechanistic Evidence

Since the vision encoder is frozen during RLVR, improvements must arise from changes in how the LLM backbone uses temporal information in frozen visual tokens. We conduct three exploratory analyses consistent with this temporal-calibration interpretation.

Upper-layer changes. CKA similarity (Kornblith et al., 2019) between base and velocity-trained models stays near-perfect for layers 0–13 but diverges at layers 14–27 (Figure 3).

Temporal utilization. Linear probing (Belinkov, 2022) shows elevated temporal-order accuracy at layer 27 (+10.9 pp; $p = 0.013$), co-localizing with the CKA divergence (Figure 4).

Frame-shuffle sensitivity. Frame-shuffle sensitivity follows the transfer profiles: clean-transfer tasks gain temporal reliance (VELOCITY: +5.7 pp, ROTATION_DIR: +1.7 pp), while the shortcut task loses it (MOTION_TYPE: −2.2 pp; Figure 5).

Together, these analyses suggest that aligned RLVR tasks change how the LLM backbone uses temporal evidence, but they are exploratory and do not establish a complete causal mechanism.

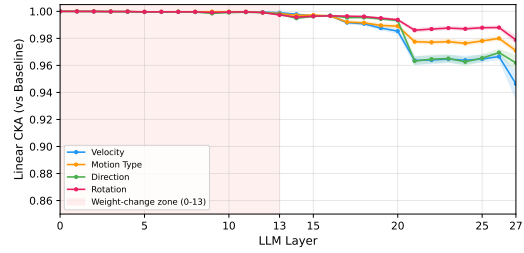


Figure 3: CKA similarity between the base and velocity-trained models. Divergence concentrates in upper LLM layers.

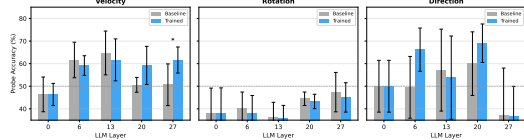


Figure 4: Temporal-order probing accuracy. The velocity-trained model shows increased temporal information in upper layers.

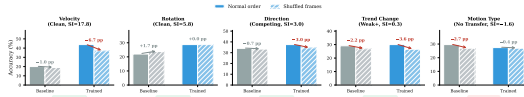


Figure 5: Frame-shuffle sensitivity (OSS). Local tasks gain temporal reliance, while shortcut-prone global tasks lose it.

A.2 Data Format and Training Prompt

Both training and evaluation use a multiple-choice question answering (MCQA) format. Each training sample pairs a synthetic video with a task-specific question template (Table 1) and 4–5 answer options, and the ground-truth option is computed directly from the rendering metadata. A representative training prompt for VELOCITY is:

```
<video>
Select the best answer to the following
multiple-choice question based on the
video. Respond with only the letter (A,
B, C or D) of the correct option. The
best answer is:
Question: What is the speed pattern of
the large green parallelogram labeled
'B's movement?
Options:
A. Accelerating.
B. Constant speed.
C. Decelerating.
D. Not moving at all.
Answer:
Output your thinking in <thinking>
</thinking> and final answer in <answer>
</answer> tags. Your answer must be a
single letter.
```

The model produces intermediate reasoning inside <thinking> tags and the final option letter inside <answer> tags; the chosen option is parsed from the <answer> tag with a regular expression.

The format reward only enforces the tag structure and a parsable option letter and does not assess correctness; format compliance is near-ceiling and nearly constant across rollouts, so it contributes little group-relative advantage under GRPO. Format-only learning would also predict similar gains across tasks, contrary to the task-specific positive and negative transfer we observe and to the failed randomized-label and COLLISION controls (§4.5). We nevertheless note that option or format bias from MCQA-style training is not fully ruled out.

A.3 Reproducibility and Release

Upon publication, we plan to release the rendering and training code¹, the task configurations and prompt templates, and a curated, documented subset of the generated data. Because all videos are generated programmatically, the released generator allows the synthetic samples to be regenerated from task configurations rather than requiring access to every stored rendering.

A.4 Model-Scale Check

To test whether diagnoses made on the 2B model remain useful at a larger scale, we select the three highest-SI tasks from the 2B interference matrix (VELOCITY, ROTATION_DIR, and MOTION_DIR; SI = 17.81, 5.77, and 3.01) and train this mixture on Qwen3-VL-8B-Instruct with the same task data and training configuration, without reconstructing a full 8B matrix. Table 10 reports the six-dimension TOMATO results. The macro average improves from 36.04 to 42.93 (+6.89 pp) on TOMATO, with non-negative changes on all six dimensions; the identical mixture improves the 2B macro average by +3.78 pp. Because this is a controlled single-seed check without a complete 8B interference matrix, it does not establish task-ranking invariance across scales; it provides initial evidence for a practical amortization route in which task diagnosis is conducted on a smaller model and the selected mixture is reused for larger-model training.

¹<https://github.com/handsomelys/pandoras-motion-box>

Model	Base Mean	Δ Mean
Qwen3-VL-2B	27.49	+3.78
Qwen3-VL-8B	36.04	+6.89

Table 10: Model-scale check on TOMATO. The top-3 SI task mixture derived from the 2B interference matrix is applied without modification to Qwen3-VL-8B-Instruct (single seed). Δ Mean is the six-dimension TOMATO macro average gain in pp; non-negative changes on all six dimensions for both models.

A.5 TNS and SAS Analysis

We use the temporal necessity score (TNS) and the skill alignment score (SAS) as lightweight task-level annotations for computational alignment. TNS measures whether the task answer depends on temporal ordering, while SAS measures whether the task computation overlaps with the evaluated skill. SAS scores are assigned from task definitions rather than fitted to observed transfer outcomes; Table 11 lists the complete TNS and SAS assignments, target dimensions, and rationales for all analyzed task–dimension pairs. As shown in Figure 6, $\text{TNS} \times \text{SAS}$ correlates with observed transfer across the analyzed task–dimension pairs. Perturbing SAS by ± 0.1 over 10K trials yields median $\rho = 0.694$ (95% CI: [0.625, 0.754]), and LLM-assigned SAS (DeepSeek-V4²) correlates with expert annotation ($\rho = 0.66$, $p = 0.015$).

Task	Tgt.	TNS	SAS	Rationale
VELOCITY	v&f	1.00	.95	Direct speed estimation
MOT. DIR	dir	1.00	.95	Movement vector tracking
ROT. DIR	rot	1.00	.95	Sequential angle comparison
MOT. TYPE	s&t	0.14	.90	Global shape $\approx$ trend
ACT. CNT	cnt	0.88	.85	Repetition $\approx$ counting
RHYTHM	v&f	1.00	.85	Tempo-shift $\approx$ speed
TREND_C.	s&t	1.00	.75	Transition $\subset$ trend
SPEED_C.	v&f	0.50	.65	Relative speed; shortcuts
MOT. REP	s&t	0.61	.60	Pattern ID $\neq$ trend
OBJ. CNT	cnt	0.00	.55	Spatial; no temporal skill
COLLISION	cnt	0.70	.45	Proximity $\neq$ counting
SEG. ORD	s&t	1.00	.25	Ordering $\neq$ pattern ID
TEMP. GRD	s&t	1.00	.15	Localization $\neq$ shape

Table 11: TNS and SAS assignment protocol for all analyzed task–dimension pairs. Target-dimension abbreviations as in Table 2. Scores are assigned from task definitions independently of observed transfer outcomes.

²<https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro>

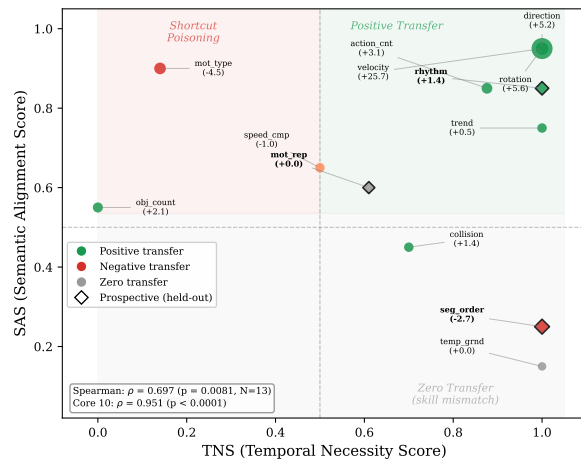


Figure 6: Task–dimension pairs in the space spanned by the temporal necessity score (TNS) and the skill alignment score (SAS). Color indicates transfer direction and marker size reflects target-dimension Δp . The product $TNS \times SAS$ correlates with observed transfer (Spearman $\rho = 0.70$, $p = 0.008$). OBJECT_COUNT is a boundary case: despite $TNS = 0$, it transfers through spatial grounding.