PRAGMATIC DYNAMICS MODELS: LEARNING EMBODIED TEMPORAL REASONING FROM FORECASTING IN LANGUAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal reasoning in embodied settings requires models to predict what will happen next and adapt when those predictions conflict with physical reality. Human motor control implements this through internal forward models that predict possible consequences and compare them with sensory feedback to determine whether adaptation is necessary. To model this process computationally, we introduce pragmatic dynamics models (PDMs), using language as a simulation space for this propose-then-verify planning loop. A generative model proposes possible futures, and the realized physical trajectory verifies what occurred. We then use pragmatic inference—a probabilistic model of how language distinguishes between possible world states—to identify unrealized futures that the model has difficulty distinguishing from reality. This yields a new action forecasting benchmark (PDM-Eval). The same pragmatic account then connects the verification process to preference learning via Direct Preference Optimization (DPO), where both feedback from physical reality and regularization constraints determine how to update based on the pragmatically selected comparison. Our theory also suggests an extension, which uses chain-of-thought explanations of the competing dynamics to further inform the update process. Empirically, our benchmark PDM-Eval produces more difficult comparisons than baseline benchmark creation strategies across five language models. Our proposed DPO-based adaptation algorithm also improves generalization of visual-temporal reasoning to new domains, compared to baselines tested on two vision-language backbones. Gains persist without generating explanations at test time, and only require updates to the language layers of the network, substantially reducing compute costs for fine-tuning.

1 INTRODUCTION

Vision-language models (VLMs) are increasingly used in settings that require predicting how the world responds to one's actions over time, or embodied temporal reasoning. Applications include robotics, autonomous driving, and video analysis (Driess et al., 2023; Brohan et al., 2023; Kim et al., 2024; Li et al., 2025c; Xie et al., 2025; Jiang et al., 2025). Although VLMs show promise in reasoning about physical action sequences (Grauman et al., 2022), they still struggle with event order, error recognition, and motion direction (Li et al., 2024; Mangalam et al., 2023; Fu et al., 2025; Liu et al., 2024). Indeed, while many future actions may be plausible from a past sequence of observations, only some will agree with how the observed world actually evolved. This uncertainty is an unavoidable component of embodied interaction. In turn, temporal reasoning requires detecting when predictions disagree with subsequent evidence and adapting accordingly.

Human motor control provides a useful model of this process. An internal forward model predicts the sensory consequences of motor commands, feedback verifies those predictions, and errors can trigger adaptation (Wolpert et al., 1995; Shadmehr et al., 2010; Izawa & Shadmehr, 2011). This propose-verify-adapt structure resembles modern planning formulations, in which a generative model proposes candidates and external verification selects failures for improvement (Kambhampati et al., 2024). One difference, however, is that motor control is situated in a forecasting setting. A more critical distinction is that motor adaptation relies on *selective feedback*: sensitivity to prediction error is non-linear in error magnitude, and only errors large enough to indicate unreliable

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

control trigger adaptation (Marko et al., 2012; Herrick & Yakovenko, 2025). Smaller errors, particularly along task-irrelevant dimensions, are instead absorbed by low-level feedback corrections (Todorov, 2004), avoiding the cost and instability of revising the internal model.

We use action forecasting in language to instantiate this temporally constrained, selective adaptation problem and call the construction a *Pragmatic Dynamics Model* (PDM). Specifically, the model generates candidate continuations from an action history without access to the next recorded event. After that event is revealed, we remove candidates that describe the recorded outcome and retain the highest-scoring remaining forecast. So, the selected continuation represents an error the model ranked highly *before* observing the verifying outcome. We then propose a benchmark *PDM-Eval*, which tests whether a forecaster can distinguish the recorded event from this unrealized forecast. In effect, the model must distinguish reality from one of the it's strongest expectations. Language, in particular, makes this comparison possible without synthesizing counterfactual video trajectories in which events evolve differently over time, a substantially harder generation problem, which requires object continuity, physical plausibility, and long-range consistency as events change across the sequence (Brooks et al., 2024; Huang et al., 2024; Motamed et al., 2026; Guo et al., 2025).

Although possibly simple at the surface, we also contribute a re-usable analysis technique for interpreting our proposed linguistic simulation, which adds to a body of literature on the connections between physically embodied interaction and social interactions in language (Wolpert et al., 2003). Specifically, we use Rational Speech Acts (RSA)—a model of how spoken messages change a listener's probabilistic beliefs about the world (Frank & Goodman, 2012)—to interpret this construction as grounding. The forecaster acts as a speaker over narrations of possible futures, while the listener interprets those narrations in terms of physical states. Under this interpretation, we show our PDM-Eval benchmark instantiates an adversarially chosen listener prior in the RSA communication model. Empirically, PDM-Eval yields larger forecasting errors than other baseline benchmark constructions across five language models.

We then use the same interpretation to derive an adaptation rule. After physical verification identifies the realized state, we ask what an RSA-optimal speaker should say to move the grounded listener toward that state. Interestingly, this analysis shows that Direct Preference Optimization (DPO) (Rafailov et al., 2023) can implement this RSA-optimal speaker when rewards are based on the grounding efficacy of the adversarial listener in our benchmark (the benchmark's ranked pairs). Our analysis also suggests sampling chain-of-thought explanations of the dynamics leading to each of the competing candidates in a preference pair. We call this algorithm PDM-DPO++.

Importantly, our motivation does *not* suggest chain-of-thought is needed at test-time to observe performance gains. Our motivation also suggests that *only* the language-layers of the model need to be updated. We verify both of these predictions hold true empirically, demonstrating performance benefits on novel domains for two vision-language backbones. These include gains compared to other language-only adaptation strategies well as similar performance to a standard action forecasting objective that fine-tunes *both* vision- and language-layers at more data and compute costs.

2 Related Work

Temporal forecasting and preference supervision. Temporal forecasting admits multiple plausible futures and substantial dependence across time, motivating evaluation through contrastive benchmarks that pair observed events with adversarially generated alternatives (Zellers et al., 2018; 2019; Bansal et al., 2024). These comparisons can also provide learning signals: recent work uses temporal preferences, counterfactual videos, and text-only temporal supervision to improve video understanding (Li et al., 2025a; Poppi et al., 2026; Li et al., 2025b), while DPO has been applied to vision-language reasoning chains to improve direct-answer prediction similar to our work (Zhang et al., 2025a;b). One core contribution of our work is a selective propose-then-verify construction motivated by motor control, which yields a unique manner of constructing contrastive pairs. This new construction also contributes to literature on evaluation of forecast distributions of LLMs (Sicilia et al., 2024). Meanwhile, our analysis also provides insight on why weakly-supervised language reasoning supports generalization to visual-temporal data (Li et al., 2025a).

Pragmatic inference and reasoning. RSA models have been incorporated into learned systems (Monroe & Potts, 2015; Teneggi et al., 2025), and recent work uses RSA to analyze or improve lan-

guage model reasoning through explicit pragmatic inference and RSA-motivated chain-of-thought procedures (Tsvilodub et al., 2025; Qiu et al., 2025). Most directly related to our optimization analysis, Nguyen (2023) interprets reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019) as inference toward a bounded pragmatic speaker. This is related to our own analysis, but focuses on social interaction. In contrast, a key distinction of our analysis is it's focus on grounding to a physical verifier, which requires a unique treatment of the RSA listener. Our analysis demonstrates how choice of prior and semantics can emulate a selective motor control problem, and in turn, builds on numerous connections between pragmatic modeling and language grounding in visual-temporal environments (Fried et al., 2023). We believe our approach can yield a useful formalism for many such inference problems that require physical grounding of language.

Motor prediction and biologically motivated algorithms. Biological theories of prediction and motor control have repeatedly motivated algorithms for temporal learning. Predictive coding accounts inspired recurrent networks for future-frame prediction (Lotter et al., 2017), optimal feedback control models have motivated neural algorithms for learning control under delayed and noisy feedback (Friedrich et al., 2021), and recent visuo-motor methods again combine forward and inverse dynamics models to learn representations for downstream control (Cui et al., 2024). Active inference provides another line from biological predictive processing to artificial agents, with generative models used for robot navigation, action selection, and temporal behavioral prediction (Çatal et al., 2020; Gong et al., 2026). Importantly, Wolpert et al. (2003) argues that this predictive machinery may also provide a computational bridge from physical control to social interaction. Our work makes physical-communicative connections operational for modern algorithm design, using a communication model to selectively reason about errors of a physical forecaster.

3 METHOD

3.1 ACTION FORECASTING AND SELECTIVE MOTOR FEEDBACK

As noted in Section 1, we use action forecasting in language to provide a linguistic proxy for embodied temporal reasoning. An action narration is a text description of a physical event recorded in video. Given a narration history $H = (Y_1, \ldots, Y_{t-1})$, where each Y_i is a text description organized by time, ordinary action forecasting trains a forecaster π_θ to predict the recorded continuation in the video $Y^+ = Y_t$ by minimizing a proper scoring loss ℓ (Grauman et al., 2022; Sicilia et al., 2024):

$$\mathcal{L}_{\mathrm{AF}}(\theta) = \mathbb{E}_{H,Y^+}\, \ell\big(\pi_\theta(\cdot \mid H), Y^+\big). \tag{1}$$

A proper scoring loss is used because it is minimized in expectation by the true outcome distribution (Gneiting & Raftery, 2007). Following prior work (Sicilia et al., 2024), we use log loss, $\ell(p, y) = -\log p(y)$, for ordinary forecasting training and Brier loss when evaluating probabilistic forecasts. With an autoregressive model π_θ used to generate the text continuation, the log loss choice yields the usual causal language modeling objective (i.e., next-token prediction with cross-entropy loss).

Selective feedback in motor planning. Interestingly, this forecasting setup has the prediction-feedback structure used in models of human motor control. An internal forward model predicts a future sensory state, and subsequent observation supplies an error signal (Fig. 1). The internal forward model acts as forecaster here. However, in motor control, adaptation is believed to be more selective, so every prediction error need not trigger a persistent change in the forecaster. Indeed, in locomotor experiments, Herrick & Yakovenko (2025) find that asymmetry in movement alone was insufficient to induce persistent adaptation, but adding asymmetric forces did produce adaptation. Their results motivate a sensory-error threshold for this response. Adaptation only occurs after *enough* realized error. We take an algorithmic commitment from this account, described next (3.2).

3.2 PDM-EVAL: COMPARING REALITY WITH A FAILED FORECAST

Selecting an alternative. Let $\pi_0 = \pi_{\theta_0}$ be a fixed initial forecasting model. Assume it generates candidate continuations from H without observing Y^+. Let $y \equiv Y$ mean that narration y describes event Y.[1] After Y^+ is revealed, semantic filtering removes candidates equivalent to that event (Appendix C.1). The surviving set $V^-(H, Y^+)$ contains the

[1] Many descriptions can narrate a physical event equally well. We model this via semantic (cosine) similarity.

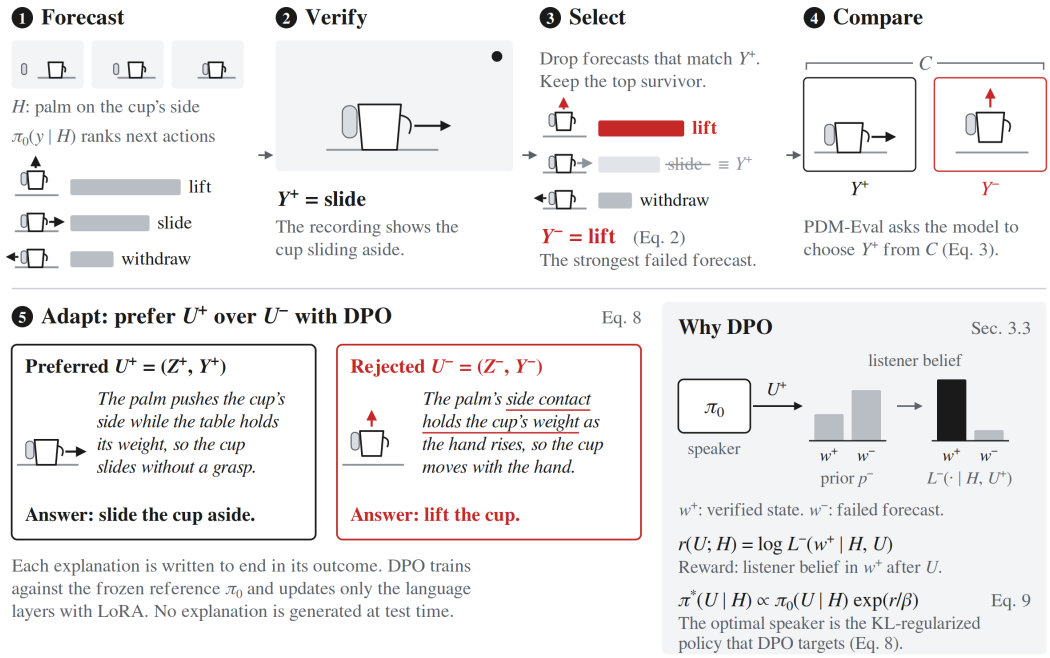


Figure 2: Pragmatic dynamics model (PDM) construction. (1) A frozen forecaster π_0 ranks next actions from the narration history H. (2) The recording reveals the realized action Y^+. (3) Y^- is selected from π_0's high-error expectations before feedback. (4) PDM-Eval tests whether a model chooses Y^+ over Y^-, and (5) PDM-DPO++ pairs each outcome with an explanation and trains on the preference with DPO, updating only the language layers. Under an RSA interpretation, this update yields the optimal speaker for a listener grounded by physical verification.

generated candidates with $y \not\equiv Y^+$, or the model's counterfactual expectations. We retain one continuation among these generated alternatives:

$$Y^- \in \arg\max_{y \in V^-} \pi_0(y \mid H), \quad C = \{Y^+, Y^-\}. \tag{2}$$

While generated alternatives are common in next-event inference problems (Zellers et al., 2018; 2019). Equation 2 implements selective correction by targeting an error the model ranked highly before feedback. This process is more faithful to the error-threshold view supplied by motor control theory, since it actually implements a form of error-thresholding behavior (derivation in Appendix B.1). Supposing there is a theoretical threshold τ^*, then the above selection procedure is guaranteed to meet this threshold in the limit on $m = |V^-(H, Y^+)|$.

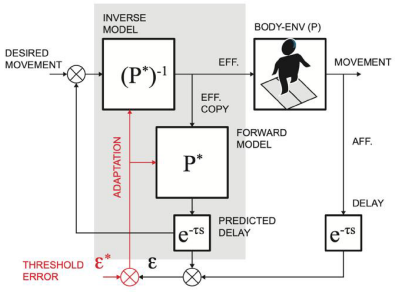


Figure 1: Motor adaptation is error-thresholded, and only prediction errors above a threshold trigger a persistent update (Herrick & Yakovenko, 2025). This motivates Eq. 2's selection rule, which retains Y^- only when it exceeds an analogous threshold τ^*.

Evaluating the comparison. We write $\pi_\theta(Y \mid H, C)$ for the forecast probability of choosing Y from the supplied pair $C = \{Y^+, Y^-\}$. For language models, conditioning on C (via the prompt, C.2) can change the model's output distribution. We use this distinction to design a new evaluation objective, tied to the selective regime in motor control:

$$\mathcal{R}_{\mathrm{PDM}}(\theta; \pi_0) = \mathbb{E}_{H, Y^+, V^-} \ell\big(\pi_\theta(\cdot \mid H, C), Y^+\big). \tag{3}$$

In experiments, PDM-Eval uses scalar binary Brier loss, $\ell(p, Y^+) = (1 - p(Y^+))^2$. The expectation includes candidate sampling and filtering (Appendix C.1).

3.3 Pragmatic Verification and Preference Learning

RSA and physical verification. Recall from Section 1: forecast narrations play the role of an RSA speaker message, while a listener interprets those messages in terms of physical states. Suppose w^+ is the ground-truth physical state described by Y^+, and let $\omega(y)$ denote the physical event described by narration y. Recall our earlier distinction

$$y \equiv Y \quad \Longleftrightarrow \quad \omega(y) = \omega(Y),$$

so each physical state corresponds to an equivalence class of narrations, with $w^+ := \omega(Y^+)$.

Given a world w, an RSA speaker S chooses it's message u based on it's effect on the listener L (Frank & Goodman, 2012). Formally, this is described

$$L_p(w \mid H, u) \propto p(w \mid H)[\![u]\!](w), \quad S_\lambda(u \mid w, H) \propto \exp\{\lambda \log L_p(w \mid H, u) - C(u, H)\} \tag{4}$$

where $p(w|H)$ is the listener's prior, C is the message's cost, and $[\![u]\!](w) \geq 0$ is the messages semantic compatibility with w. Here, $\lambda > 0$ is a weight on the speaker's communicative success, relative to message production cost. RSA commonly uses production costs to account for differences in message difficulty or availability (Goodman & Frank, 2016). One choice is surprisal under a message prior, which we select to be the frozen forecaster $C(u, H) = -\log \pi_0(u \mid H)$. Responses that were unlikely under the original forecaster are more costly speaker messages to produce.

In general, the semantics encode grounding. For a message $u = (z, y)$ consisting of reasoning z and final event description y, we define our semantics as below:

$$[\![(z, y)]\!](w) = \epsilon + (1 - \epsilon)\mathbf{1}\{\omega(y) = w\}, \qquad 0 < \epsilon < 1. \tag{5}$$

Thus, a narration describing the physical world has compatibility one, while an incompatible narration retains semantic noise ϵ. The semantics ignores the reasoning z.

PDM-Eval as an adversarial listener. After the recording reveals Y^+, each $y \in V^-$ describes an unrealized world $w_y := \omega(y)$. The initial forecast probabilities for these worlds induce a listener's prior over two represented physical states

$$p_y^0(w \mid H) \propto \pi_0(Y^+ \mid H)\mathbf{1}\{w = w^+\} + \pi_0(y \mid H)\mathbf{1}\{w = w_y\}. \tag{6}$$

PDM-Eval (Eq. 2) selects the failed forecast with strongest, prior grounded belief, pre-verification:

$$Y^- \in \arg\min_{y \in V^-} p_y^0(w^+ \mid H) = \arg\max_{y \in V^-} \pi_0(y \mid H). \tag{7}$$

We write $w^- := \omega(Y^-)$ for shorthand and give a derivation of Eq. 7 in Appendix B.2.

Cooperative adaptation and PDM-DPO++. We next fix the selected prior $p^- := p_{Y^-}^0$ and write $L^- := L_{p^-}$. Considering the speaker update for this fixed listener yields an interesting observation: the RSA-optimal speaker in Eq. 4 has identical form to the optimum of the reward objective in Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2019; Rafailov et al., 2023). Indeed, taking reward $r(U; H) = \log L^-(w^+ \mid H, U)$ for message U in the RLHF objective

$$\max_\pi \ \mathbb{E}_{U \sim \pi(\cdot|H)} r(U; H) - \beta D_{\mathrm{KL}}(\pi(\cdot \mid H) \| \pi_0(\cdot \mid H)), \qquad \beta > 0 \tag{8}$$

yields the following optimum,

$$\pi^*(U \mid H) \propto \pi_0(U \mid H) \exp\{r(U; H)/\beta\} = \pi_0(U \mid H)[L^-(w^+ \mid H, U)]^{1/\beta}. \tag{9}$$

This is the RSA-optimal speaker with $\lambda = 1/\beta$ (derived in Appendix B.3). Here, physical verification, encoded in the RSA semantics, supplies feedback that plays the role of human preferences in RLHF. Moreover, we encode physical grounding via the listener of our RSA model, so this analysis suggests the objective in Eq. 8 is an RSA-optimal update strategy to promote effective grounding.

In practice, we sample two reasoning trajectories $U^+ = (Z^+, Y^+)$ and $U^- = (Z^-, Y^-)$ from the base forecaster by supplying the corresponding terminal action in the generation prompt (Appendix C.1). The noisy listener assigns U^+ greater reward than U^-, so physical verification supplies preferences consistent with the objective in Eq 8. As a reminder, by design of our listener, these preferences also match the benchmark design in Eq. 3. However, the KL-regularized objective above is distinct from direct minimization of Eq. 3. The RSA analysis suggests necessary regularization. Thus, to train on these preferences, we optimize the forecaster with DPO (Rafailov et al., 2023), which is designed to approximate the KL-regularized objective in Eq. 8 under certain preference modeling assumptions. We call the procedure *PDM-DPO++*, or *PDM-DPO* in special case $Z = \varnothing$.

Reasoning as reusable evidence. Under the grounding interpretation, the reasoning Z^* can provide additional evidence about which physical state a narration describes. To see this, we roll the RSA model one step further and ask how a listener interprets reasoning produced by the speaker. In RSA, a pragmatic listener updates based on speaker messages by combining its prior belief over world states with the likelihood that the speaker would produce the observed message in each state (Frank & Goodman, 2012). Let $w = \omega(Y_w)$. For reasoning evidence Z, applying the optimal cooperative speaker $S^*(Z \mid w, H) := \pi^*(Z|Y_w, H)$ in the usual RSA listener update yields

$$L^*(w \mid H, Z) \propto \underbrace{p^-(w \mid H)}_{\text{listener prior}} \underbrace{S^*(Z \mid w, H)}_{\text{speaker likelihood}} = p^-(w \mid H)\pi_0(Z \mid Y_w, H), \tag{10}$$

where the second step follows because the log-listener reward depends on the terminal narration Y_w, so its factor cancels when conditioning the speaker on w. Expanding the listener prior from Eq. 7:

$$\log \frac{L^*(w^+ \mid H, Z)}{L^*(w^- \mid H, Z)} = \underbrace{\log \frac{\pi_0(Y^+ \mid H)}{\pi_0(Y^- \mid H)}}_{\text{initial forecast}} + \underbrace{\log \frac{\pi_0(Z \mid Y^+, H)}{\pi_0(Z \mid Y^-, H)}}_{\text{reasoning evidence}}. \tag{11}$$

Equation 11 separates the listener's initial belief from the additional evidence supplied by reasoning. Reasoning favors w^+ when it is more likely under Y^+ than under Y^-. This occurs (on average) when sampling from the conditional $\pi_0(Z \mid Y^+, H)$ (Appendix B.4), which motivates our implementation choice in PDM-DPO++ to construct preferred reasoning conditioned on the verified outcome. If information represented by that reasoning later appears in a visual-temporal context, the listener can use it as evidence without generating the reasoning itself. This provides a formal mechanism by which linguistic reasoning used during adaptation can improve grounding to physical states. We discuss implementation and experimentation for this hypothesis (among others), next.

4 Experiments

4.1 Experimental Setup

Our experiments test four predictions from Sections 3.2 and 3.3. First, PDM-Eval should produce harder forecasting comparisons than constructions without our selection rule. Second, PDM-DPO++ should improve visual-temporal reasoning beyond its language-only training distribution. Third, explanations used during adaptation should help without being generated at test time. Finally, narration-only supervision should reduce the cost of adaptation relative to full vision-language fine-tuning. We test the first prediction with language-only forecasting to isolate the counterfactual construction, and the remaining three on two vision-language backbones. Appendix C.1–C.2 provides implementation details.

4.1.1 Evaluating Pragmatic Counterfactuals

Benchmark Construction. We construct PDM-Eval from held-out Ego4D narrations (Grauman et al., 2022) using the procedure in Section 3.2. A frozen reference model, LLaMA-3.2-1B-Instruct, proposes continuations of each history ($m = 10$). We use one small, fixed reference model rather than each evaluated forecaster, so that all forecasters are compared against counterfactuals from a common source. More filtering and selection details are in Appendix C.1.

Counterfactual Baselines. We compare PDM counterfactuals with two constructions. *Random* draws a recorded action from another Ego4D video and tests whether any unrealized future produces a difficult comparison. *Semantic* draws a recorded action that shares a content noun with the current example to test impact of linguistic relatedness. This is similar in spirit to use of semantically plausible contrast captions (e.g., model-generated action perturbation) used in other vision-language benchmarks (Bansal et al., 2024), however it is adapted to the forecasting setting, by more reliably using the available ground-truth narration data.

Forecasting Problems. Our primary evaluation uses a closed format, which supplies both possible futures in the prompt and implements the candidate-conditioned risk in Eq. 3. We also report an open format, which scores each future without its competitor as in ordinary action forecasting Eq. 1,

to test whether supplying the comparison changes difficulty. Supplying the competing future indeed increases likelihood-based Brier error over the open format for all five forecasters and all three constructions (Figure 6), confirming that ordinary action forecasting alone does not measure this difficulty. Probabilities come from normalized model likelihoods, and we repeat each evaluation with verbalized probability reports (Tian et al., 2023; Sicilia et al., 2024) as a second elicitation mechanism. We report scalar binary Brier scores (↓) for LLaMA-3.1-8B and LLaMA-3.2-11B (Grattafiori et al., 2024), Mistral-7B (Jiang et al., 2023), Qwen3-8B (Yang et al., 2025), and Phi-3-mini (Abdin et al., 2024).

Discriminative Transfer. As a diagnostic baseline, we train LLaMA-3.2-11B to directly discriminate contrastive pairs based on Eq. 3 using a cross-entropy (log) loss. This implements traditional, multiple-choice (MC) discriminative fine-tuning (Zellers et al., 2018; 2019). We do so for each of the three constructions and evaluate each trained model on held-out constructions. This tests whether directly minimizing the candidate-conditioned error on one construction generalizes to the others.

4.1.2 Adaptation and Temporal Transfer

Training. All adaptation data come from Ego4D narrations. For each backbone, π_0 is the initial checkpoint of the model being trained. It generates and selects counterfactuals with Eq. 2 and remains frozen as the DPO reference (Rafailov et al., 2023) for PDM-DPO and PDM-DPO++ as implemented in Section 3.3. Any explanations are generated once before training, conditioned on the corresponding terminal action (Appendix C.1). LoRA (Hu et al., 2021) updates only the language backbone, and vision parameters remain frozen. This tests our hypothesis about language-only grounding, as motivated by our RSA analysis. All adapted models answer directly at evaluation, without generating explanations, which tests re-usability of chain-of-thought evidence (also hypothesized under our RSA model).

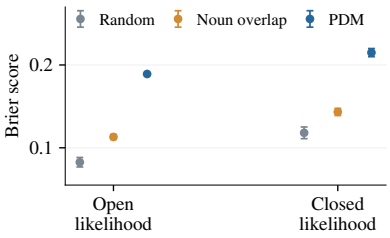


Figure 3: Brier score by counterfactual construction (Random, noun overlap, PDM) under open and closed likelihood scoring. PDM produces the largest error in every condition, confirming it yields harder comparisons than distractors selected by chance or lexical overlap.

Baselines. We test a number of language-only baselines, with each removing aspects of our theoretical motivation, or implementing common alternatives. *AF* trains on the same histories and recorded continuations without a comparison, implementing standard action forecasting (Eq. 1). *MC* is the diagnostic multiple-choice baseline described previously (Zellers et al., 2018; 2019), which lacks regularization or reasoning traces as motivated by our analysis. *AF + Thresh* retains only high-error examples under an empirical Brier cutoff (Appendix Fig. 5), a static version of the motor-control threshold. CoT and ToT apply reasoning at test time instead of during training (Wei et al., 2022; Kojima et al., 2022; Yao et al., 2023). Meanwhile, our PDM-DPO implementation removes explanations from PDM-DPO++.

Compositional Baselines. While we use LLaMA-3.2-11B (Grattafiori et al., 2024) as our primary backbone, we also experiment with Qwen2.5-VL-32B (Bai et al., 2025), a larger model explicitly trained for video understanding and further post-trained with preference optimization. Its training strategy incorporates several ingredients used independently in related visual-temporal methods, including video-specific supervision, reasoning traces, and preference learning. Thus, applying PDM-DPO++ to Qwen tests whether our own construction provides complementary supervision on top of an already temporally trained and preference-optimized VLM. We separately use Qwen3-14B (Yang et al., 2025), a text-only member of the same model family, as the frozen reasoning model that generates chosen and rejected explanations for PDM-DPO++ (Appendix C.1).

Temporal Evaluation. We evaluate on TemporalBench (Cai et al., 2024) without task-specific fine-tuning and report Multiple Binary Accuracy (MBA, ↑), which counts a video group as correct only when all its binary comparisons are correct. The full benchmark serves as the aggregate measure of transfer to new visual-temporal environments. TemporalBench aggregates 9,867 binary temporal-ordering questions, grouped into 2,179 per-video comparisons, drawn from seven

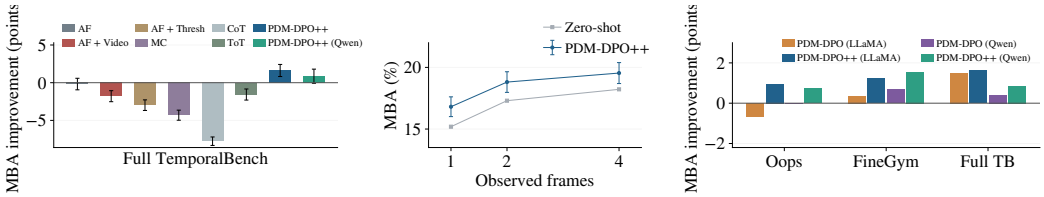


Figure 4: Temporal reasoning gains from PDM-DPO++. (a) MBA improvement over zero-shot on full TemporalBench across baselines and test-time reasoning controls. (b) MBA as a function of observed frames, zero-shot vs. PDM-DPO++. (c) MBA improvement on Oops, FineGym, and full TemporalBench for PDM-DPO vs.PDM-DPO++, on LLaMA and Qwen. PDM-DPO++ is the only method that improves consistently across baselines, frame counts, and backbones, raising full-TemporalBench MBA from 15.19 to 16.81 (LLaMA) and 24.97 to 25.83 (Qwen), at substantially lower cost than AF+Video: 22 GB/9.7 h versus 48 GB/27.8 h (Appendix Figure 10; Table 2).

source datasets spanning egocentric (EgoExo4D), instructional (COIN), failure/unintentional-action (Oops), fine-grained gymnastics (FineGym), household (Charades), narrated-film (Movie Description), and open-domain (ActivityNet) video. We also report two subsets tied to the motor-control motivation: Oops (Epstein et al., 2020), which contains unintentional and failed actions, and FineGym (Shao et al., 2020), which distinguishes closely related gymnastics movement states. These engage with the motivating constraints of motor adaptation (Section 1), including error-correction and adaptive dexterity. Unless stated otherwise, models observe one historical frame, and figures show percentage-point changes from each backbone's zero-shot score. We also supply one, two, or four frames to test whether gains persist as more motion evidence becomes available.

Broader Evaluation and Cost. We use the active and episodic-memory tasks of OpenEQA (Majumdar et al., 2024) to check whether the temporal update degrades broader vision-language performance. We also compare the peak GPU memory and wall-clock training time of PDM-DPO++ with AF trained on video using LoRA in both the language and vision stacks.

4.2 Results

4.2.1 Pragmatic Selection Produces Harder Comparisons

Figure 4 compares the three constructions within each elicitation condition. PDM produces the largest Brier error in all 20 matched comparisons against Random and all 19 available comparisons against Semantic, and the mean error increases from Random to Semantic to PDM in every condition. While linguistic relatedness accounts for part of counterfactual difficulty, as used in prior benchmark designs (Bansal et al., 2024), our proposed selection rule adds substantially more difficulty. This ordering is predicted in Section 3.2: the retained counterfactual is the failure the reference model found most plausible, which Eq. 2 associates with the largest pairwise error and Eq. 7 with an adversarial listener prior. The failures, selected by a 1B reference model, also remain difficult for forecasters up to 11B, suggesting that they are not specific to the model that proposed them.

Figure 8 shows two effects. First, training on Random or Semantic pairs leaves PDM error at 0.1957 and 0.2380, far above the 0.0394 reached by training on PDM itself, so the difficulty of PDM comparisons is not learned from the other constructions. Second, PDM-trained MC fits its own construction but reaches 0.3085 on Random and 0.3415 on Semantic, above the zero-shot closed-likelihood errors for the same model (Figure 7). This indicates that directly minimizing Eq. 3 without regularization over-fits to the supplied alternatives, failing to generalize. The KL term in Eq. 8 is intended to promote generalization, and we evaluate the regularized update through downstream transfer next.

4.2.2 Pragmatic Adaptation Improves Temporal Generalization

Figure 4 reports temporal reasoning after training only on Ego4D narrations. PDM-DPO++ improves LLaMA by $+0.96$ MBA points on Oops, $+1.24$ on FineGym, and $+1.62$ on full Temporal-

Bench, from 15.19 to 16.81. On the video-tuned Qwen backbone, it improves by $+0.74$, $+1.54$, and $+0.86$, from 24.97 to 25.83 on the full benchmark (Figure 4). Supervision constructed entirely in language transfers to visual temporal judgments in new domains on both backbones, while only the language layers are updated. Figure 11 also checks whether the temporal update degrades broader vision-language performance on LLaMA, confirming it does not.

On LLaMA, PDM-DPO++ is the only method that improves all three benchmarks, and each baseline isolates a different explanation for this gain. AF trains on the same histories without a comparison, so exposure to Ego4D narrations alone does not account for the improvement. MC trains on the same pairs without reference regularization, showing consistent over-fitting behavior as in Figure 8. AF + Thresh also falters, so selecting high-error examples without the verified comparison creates unstable performance gains. Finally, test-time CoT and ToT remain below zero-shot on every reported benchmark, so reasoning at inference does not substitute for our designed adaptation.

4.2.3 Reasoning during Adaptation

Figure 4 compares answer-only PDM-DPO with PDM-DPO++. Explanations give higher scores in all six comparisons, although neither model generates them at test time. This agrees with the interpretation in Eq. 11: an explanation conditioned on the verified outcome acts as evidence during adaptation, and the adapted model need not produce that evidence to benefit.

4.2.4 Adaptation under Sparse Visual Evidence

Performance Gains. Figure 9 varies the number of observed frames. PDM-DPO++ exceeds zero-shot in all 18 combinations of backbone, benchmark, and frame count (Appendix D). The adapted one-frame LLaMA also exceeds the zero-shot model given four frames (18.25 versus 17.71), so on this benchmark adaptation compensates for part of the missing visual input. Importantly, this compensation is tested on novel visual-temporal domains, implying generalization.

Visual Evidence and Action Priors. Because PDM-DPO++ trains on narrations, part of its gain could come from stronger action priors rather than better use of visual evidence. The sweep does not fully separate these explanations. Gains persist as frames are added, and on LLaMA FineGym they widen from 1.24 to 2.39 points. This provides some evidence suggesting better use of visual evidence in some cases.

Comparison with Vision Fine-tuning. Figure 10 compares PDM-DPO++ with AF trained on video using LoRA in both the language and vision stacks. AF + video requires 48 GB of peak GPU memory and 27.8 hours, compared with 22 GB and 9.7 hours for PDM-DPO++, or 54% lower memory and 65% lower training time. Most of this difference follows from the construction: PDM-DPO++ trains on narrations, so it neither processes video nor backpropagates through the vision encoder, even though DPO adds reference forward passes.

5 Discussion

Our results suggest that language can stand in for costly counterfactual simulation: a model rehearses its own failures in text, and adaptation transfers to visual-temporal domains without generating any counterfactual videos. More broadly, we believe the RSA model underlying this construction is reusable. Because physical verification is encoded in the semantics and grounding is carried by the listener, new verifiers, such as simulators, robot sensors, etc. yield new benchmarks and update rules without entirely new theory. Our analysis also suggests two directions for current methods. First, it identifies confident failures as a useful supervision signal for proposal-verification planning loops (Kambhampati et al., 2024). For preference methods such as GRPO (Shao et al., 2024), the adversarial listener prior (Eq. 7) offers a principled criterion for which failures to contrast. Second, our results echo prior work which finds that textual reasoning supervision can transfer to video (Li et al., 2025a; Zhang et al., 2025a). Interestingly, Eq. 11 offers a formal explanation by separating the listener's initial forecast from newly learned reasoning evidence. The same decomposition could, in turn, be used across domains to ask whether gains stem from stronger action priors or from evidence that grounds them, an open question for VLM evaluation (Xu et al., 2026).

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, discovery of related works, typesetting LaTeX (including tables and mathematical equations), coding of algorithms and experiments designed by the authors, and verifying mathematical arguments derived by the authors. We have not used generative AI tools for generating research ideas, and other tasks with required disclosure are not applicable to this work. We have reviewed all AI-assisted work. Related-work suggestions were checked against the original sources before citation, and AI-assisted LaTeX edits, including equation and table formatting, were proofread against the intended content. AI-assisted code for algorithms and experiments was tested and verified for correctness by the authors, and AI-assisted checks of our mathematical derivations were cross-checked against the paper's stated theory by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Marah Abdin, Sam Adé Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Singh Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio C'esar Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allison Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Young Jin Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Corby Rosset, Sambudha Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Olatunji Ruwase, Praneetha Vaddamanu, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Andre Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonal Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, Xi-Rui Zhou, and Yifan Yang. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, abs/2404.14219, 2024. URL https://api.semanticscholar.org/CorpusID:269293048.

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*, 2025.

Hritik Bansal, Yonatan Bitton, Idan Szpektor, Kai-Wei Chang, and Aditya Grover. VideoCon: Robust video-language alignment via contrast captions. In *Proceedings of CVPR*, 2024. URL https://arxiv.org/abs/2311.10111.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024.

Ozan Çatal, Samuel Wauthier, Tim Verbelen, Cedric De Boom, and Bart Dhoedt. Deep active inference for autonomous robot navigation. In *Bridging AI and Cognitive Science Workshop*, pp. 1–12, 2020.

Zichen Jeff Cui, Hengkai Pan, Aadhithya Iyer, Siddhant Haldar, and Lerrel Pinto. DynaMo: In-domain dynamics pretraining for visuo-motor control. In *Advances in Neural Information Processing Systems*, volume 37, pp. 33933–33961, 2024. doi: 10.52202/079017-1069.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 916–926. IEEE, 2020.

Michael C. Frank and Noah D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998, 2012. doi: 10.1126/science.1218633.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. Pragmatics in language grounding: Phenomena, tasks, and modeling approaches. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12619–12640, 2023.

Johannes Friedrich, Siavash Golkar, Shiva Farashahi, Alexander Genkin, Anirvan M. Sengupta, and Dmitri B. Chklovskii. Neural optimal feedback control with local learning rules. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16358–16370, 2021.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Zhiren Gong, Chao Yang, Wendi Ren, and Shuang Li. Learning human habits with rule-guided active inference. In *The Fourteenth International Conference on Learning Representations*, 2026.

Noah D Goodman and Michael C Frank. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829, 2016.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18995–19012, 2022.

Xuyang Guo, Jiayan Huo, Zhenmei Shi, Zhao Song, Jiahao Zhang, and Jiale Zhao. T2vphysbench: A first-principles benchmark for physical consistency in text-to-video generation. *arXiv preprint arXiv:2505.00337*, 2025.

Emily M Herrick and Sergiy Yakovenko. Evidence of sensory error threshold in triggering locomotor adaptations in humans. *PLoS One*, 20(4):e0321949, 2025.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Jun Izawa and Reza Shadmehr. Learning from sensory and reward prediction errors during motor adaptation. *PLoS computational biology*, 7(3):e1002012, 2011.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, October 2023. URL http://arxiv.org/abs/2310.06825. arXiv:2310.06825 [cs].

Tiancheng Jiang, Henry Wang, Md Sirajus Salekin, Parmida Atighehchian, and Shinan Zhang. Domain adaptation of vlm for soccer video understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 6101–6111. IEEE, 2025.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. Llms can't plan, but can help planning in llm-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024.

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 *International Conference on Learning Representations*, volume 2025, pp. 74070–74102, 2025a.

Rui Li, Xiaohan Wang, Yuhui Zhang, Orr Zohar, Zeyu Wang, and Serena Yeung-Levy. Temporal preference optimization for long-form video understanding. *arXiv preprint arXiv:2501.13919*, 2025b.

Yue Li, Meng Tian, Zhenyu Lin, Jiangtong Zhu, Dechang Zhu, Haiqiang Liu, Yueyi Zhang, Zhiwei Xiong, and Xinhai Zhao. Fine-grained evaluation of large vision-language models in autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9431–9442. IEEE, 2025c.

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 *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, 2024.

William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations*, 2017.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16488–16498, 2024.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

Mollie K Marko, Adrian M Haith, Michelle D Harran, and Reza Shadmehr. Sensitivity to prediction error in reach adaptation. *Journal of neurophysiology*, 108(6):1752–1763, 2012.

Will Monroe and Christopher Potts. Learning in the rational speech acts model. In *Proceedings of the 20th Amsterdam Colloquium*, pp. 1–12. ILLC, 2015. doi: 10.53437/ph73fp56.

Saman Motamed, Laura Culp, Kevin Swersky, Priyank Jaini, and Robert Geirhos. Do generative video models understand physical principles? In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 948–958. IEEE, 2026.

Khanh Nguyen. Language models are bounded pragmatic speakers: Understanding RLHF from a bayesian cognitive modeling perspective. In *Proceedings of the First Workshop on Theory of Mind in Communicating Agents*, volume 202 of *Proceedings of Machine Learning Research*, 2023.

Tobia Poppi, Burak Uzkent, Amanmeet Garg, Lucas Porto, Garin Kessler, Yezhou Yang, Marcella Cornia, Lorenzo Baraldi, Rita Cucchiara, and Florian Schiffers. Countervid: Counterfactual video generation for mitigating action and temporal hallucinations in video-language models. *arXiv preprint arXiv:2601.04778*, 2026.

Linlu Qiu, Cedegao E. Zhang, Joshua B. Tenenbaum, Yoon Kim, and Roger P. Levy. On the same wavelength? evaluating pragmatic reasoning in language models across broad concepts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19913–19935. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.1008.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Reza Shadmehr, Maurice A Smith, and John W Krakauer. Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience*, 33(1):89–108, 2010. doi: 10.1146/annurev-neuro-060909-153135.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2613–2622. IEEE, 2020.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Anthony Sicilia, Hyunwoo Kim, Khyathi Chandu, Malihe Alikhani, and Jack Hessel. Deal, or no deal (or who knows)? forecasting uncertainty in conversations using large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11700–11726, 2024.

Jacopo Teneggi, Zhenzhen Wang, Paul H Yi, Tianmin Shu, and Jeremias Sulam. Direct preference optimization for adaptive concept-based explanations. *arXiv preprint arXiv:2505.15626*, 2025.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023.

Emanuel Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915, 2004.

Polina Tsvilodub, Kanishk Gandhi, Haoran Zhao, Jan-Philipp Fränken, Michael Franke, and Noah D. Goodman. Non-literal understanding of number words by language models. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 47, 2025. URL `https://escholarship.org/uc/item/08c3s3jt`.

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. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Daniel M Wolpert, Zoubin Ghahramani, and Michael I Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995.

Daniel M. Wolpert, Kenji Doya, and Mitsuo Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society B*, 358 (1431):593–602, 2003. doi: 10.1098/rstb.2002.1238.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are vlms ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6285–6297. IEEE, 2025.

Zhongxing Xu, Chengzhi Liu, Qingyue Wei, Juncheng Wu, James Zou, Xin Wang, Yuyin Zhou, and Sheng Liu. More thinking, less seeing? assessing amplified hallucination in multimodal reasoning models. *Advances in Neural Information Processing Systems*, 38:82878–82905, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of EMNLP*, pp. 93–104, 2018. URL https://aclanthology.org/D18-1009/.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of ACL*, pp. 4791–4800, 2019. URL https://aclanthology.org/P19-1472/.

Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander G Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 694–717, 2025a.

Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1662. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.acl-long.82.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

B Derivations for Pragmatic Verification and Adaptation

B.1 Maximum Selection and Forecast Error

Fix H, Y^+ and write

$$a = \pi_0(Y^+ \mid H) > 0, \qquad b = \pi_0(y \mid H) > 0.$$

Restricting π_0 to the pair Y^+, y assigns probability $a/(a+b)$ to the observed event Y^+. The resulting scalar binary Brier and log losses are

$$B(b) = \left(\frac{b}{a+b}\right)^2, \qquad J(b) = \log(1 + b/a).$$

Their derivatives are

$$B'(b) = \frac{2ab}{(a+b)^3} > 0, \qquad J'(b) = \frac{1}{a+b} > 0.$$

Thus, among accepted alternatives, selecting the candidate with largest $\pi_0(y \mid H)$ also selects the candidate inducing the largest pairwise Brier or log loss.

Now suppose an accepted counterfactual description y^* attains a desired error threshold τ^*, and let

$$p^* = P\big(\pi_0(Y^- \mid H) \geq \pi_0(y^* \mid H) \,\big|\, H, Y^+\big) > 0$$

for a candidate Y^- which has passed the verifier filter in our counterfactual construction.

For m independent accepted candidates, Equation 2 fails to select an alternative at least as likely as y^* only if all m candidates fall below this likelihood. Hence

$$P\left(\max_{1 \leq i \leq m} \pi_0(Y_i^- \mid H) \geq \pi_0(y^* \mid H)\right) = 1 - (1 - p^*)^m \longrightarrow 1 \qquad \text{as } m \to \infty.$$

Because both losses above increase with counterfactual likelihood, the selected continuation meets or exceeds τ^* with probability approaching one as the number of accepted candidates grows.

B.2 Physical States and the Adversarial Listener

Fix H, Y^+ and write $w^+ = \omega(Y^+)$ for the verified physical state. Each accepted alternative y represents a different state $w_y = \omega(y)$. Set $a = \pi_0(Y^+ \mid H) > 0$ and $b_y = \pi_0(y \mid H) > 0$. Normalizing Eq. 6 gives

$$p_y^0(w^+ \mid H) = \frac{a}{a + b_y}, \qquad p_y^0(w_y \mid H) = \frac{b_y}{a + b_y}.$$

Since $\frac{d}{db_y}[a/(a+b_y)] = -a/(a+b_y)^2 < 0$, the listener's prior on w^+ is a decreasing function of b_y. Selecting y to minimize this prior is the same as selecting y to maximize b_y:

$$\arg\min_{y \in V^-} p_y^0(w^+ \mid H) = \arg\max_{y \in V^-} \pi_0(y \mid H),$$

which proves Eq. 7.

B.3 Log-Listener Reward and the RSA Speaker

Fix H, Y^+ and the selected prior $p^- = p^0_{Y^-}$. Write $a = \pi_0(Y^+ \mid H) > 0$ and $b = \pi_0(Y^- \mid H) > 0$. For a complete response $U = (Z, Y)$, define

$$r(U; H) = \log L^-(w^+ \mid H, U).$$

The target w^+ and selected comparison remain fixed throughout this calculation. The reward depends on the terminal narration through its semantics.

Listener probabilities and preferences. Under Eq. 5, a response ending in Y^+ has compatibility one with w^+ and ϵ with w^-, whereas a response ending in Y^- reverses these values. Hence

$$l_+ := L^-(w^+ \mid H, (z, Y^+)) = \frac{a}{a + \epsilon b}, \qquad l_- := L^-(w^+ \mid H, (z, Y^-)) = \frac{\epsilon a}{\epsilon a + b}.$$

Both probabilities are positive, and

$$r(U^+; H) - r(U^-; H) = \log \frac{\epsilon a + b}{\epsilon(a + \epsilon b)} > 0.$$

Positivity follows because the numerator exceeds the denominator when $0 < \epsilon < 1$. Since the semantics ignores z, this reward ordering holds for any explanations accompanying the selected terminal actions. So, physical verification supplies preferences consistent with log-listener reward.

For a response describing neither represented state, both compatibilities are ϵ and the listener probability is $a/(a + b)$. Thus the reward is also defined outside the selected pair. More generally, $\epsilon p^-(w^+ \mid H) \le L^-(w^+ \mid H, U) \le 1$, so the log reward is finite and bounded for any fixed comparison.

Matching the speaker to the regularized optimum. Substituting the message cost $C(U, H) = -\log \pi_0(U \mid H)$ into the RSA speaker yields

$$S_\lambda(U \mid w^+, H) \propto \pi_0(U \mid H) \exp\{\lambda \log L^-(w^+ \mid H, U)\}.$$

With $\lambda = 1/\beta$, this is the normalized policy

$$\pi^*(U \mid H) = \frac{\pi_0(U \mid H) e^{r(U;H)/\beta}}{B(H)}, \qquad B(H) = \sum_{U'} \pi_0(U' \mid H) e^{r(U';H)/\beta}.$$

The rewards are bounded, which ensures $0 < B(H) < \infty$.

B.4 Reasoning Evidence and the Pragmatic Listener

Take representatives $Y_{w^+} = Y^+$ and $Y_{w^-} = Y^-$. Because log-listener reward depends only on terminal narration,

$$\pi^*(Z \mid Y_w, H) = \frac{\pi_0(Z, Y_w \mid H) e^{r(Y_w;H)/\beta}}{\sum_{Z'} \pi_0(Z', Y_w \mid H) e^{r(Y_w;H)/\beta}} = \pi_0(Z \mid Y_w, H).$$

Write $q_w(Z) = \pi_0(Z \mid Y_w, H)$. Combining this speaker likelihood with the selected prior yields

$$L^*(w \mid H, Z) = \frac{p^-(w \mid H) q_w(Z)}{p^-(w^+ \mid H) q_{w^+}(Z) + p^-(w^- \mid H) q_{w^-}(Z)}.$$

This proves Eq. 10. Taking odds (and assuming positive support) we recover

$$\log \frac{L^*(w^+ \mid H, Z)}{L^*(w^- \mid H, Z)} = \log \frac{a}{b} + \log \frac{q_{w^+}(Z)}{q_{w^-}(Z)}.$$

This is Eq. 11. The second term favors w^+ exactly when the observed reasoning is more likely conditional on Y^+ than on Y^-. If $Z^+ \sim q_{w^+}$ and q_{w^+} is absolutely continuous with respect to q_{w^-}, then

$$\mathbb{E}_{Z^+ \sim q_{w^+}} \log \frac{q_{w^+}(Z^+)}{q_{w^-}(Z^+)} = D_{\mathrm{KL}}(q_{w^+} \| q_{w^-}) \ge 0.$$

Thus, such sampling as implemented in PDM-DPO++ increases the second log odds on average.

C Experimental Details and Source Verification

C.1 Candidate and Response Construction

Reference models for evaluation and adaptation. During development, a fixed reference forecaster, `meta-llama/Llama-3.2-1B-Instruct`, was used to generate candidate pairs for distractors built with an LLM, loaded independently of the forecaster under evaluation, to compare three candidate construction strategies (random sampling, shared noun matching, and distractors generated with an LLM) under a controlled distractor generator. This specific benchmark construction is an approximation of the propose-and-verify structure in the main text, but was useful during development for isolating the effect of distractor construction strategy from the effect of the forecaster being evaluated.

This comparison informed the choice of distractor construction approach used in subsequent experiments. For the main comparison table, no fixed reference forecaster is used. Instead, both the LLaMA and Qwen vision backbones generate their own distractors, so in those runs the evaluated forecaster and the candidate generating model are the same model.

For adaptation, the methods identify π_0 with the initial checkpoint of the forecaster being trained. It supplies both the generated comparisons for training and remains frozen as the DPO reference.

Constructing candidates for adaptation. To generate counterfactuals, we generate continuations from H, filter candidates that describe Y^+, and retain one highest-scoring surviving continuation. The accepted candidate set V^- is distinct from both the raw proposals and this single selected continuation. The archived filter uses a frozen `all-MiniLM-L6-v2` encoder and a cosine similarity threshold of 0.85 to exclude near paraphrases.

Candidates are generated one at a time in a loop, stopping early once ten valid, unique candidates are collected, or after a hard ceiling of 30 raw generation attempts (three attempts allowed per target candidate), whichever comes first. Typically, far fewer than 30 are needed in practice. One distractor is retained per question.

Each raw candidate must clear the following criteria before counting toward the target.

- Meets a minimum and maximum length.
- Ends in sentence-final punctuation.
- Does not end on a stopword.
- Contains no garbage substrings.
- Differs from the ground truth.
- Overlaps with the ground truth in fewer than 0.6 of its words by Jaccard overlap.
- Does not duplicate previously accepted raw text (checked by exact matching).

This process is applied before, and independently of, the semantic (MiniLM and cosine) filter described above.

Proposer checkpoint and decoding. The proposer checkpoint depends on the training regime and backbone. In the runs that produced the reported results, each backbone self-generates its own negative and distractor pool from its own base checkpoint, drawn once and frozen for the remainder of training: `meta-llama/Llama-3.2-11B-Vision-Instruct` for LLaMA, and `Qwen/Qwen2.5-VL-32B-Instruct` for Qwen. Neither backbone relies on an external model to propose candidates. Qwen3-14B serves solely as the explanation generator described below and never proposes the underlying action candidates for either backbone. An alternative iterative self-play regime instead regenerates negatives repeatedly during training, mining them against the trainee's own evolving checkpoint saved from the previous training chunk of 150 videos, so the negative pool evolves against the policy being trained as it improves.

Decoding uses sampling with cumulative probability 0.9 and temperature 0.8. One sequence is generated per call, capped at $\max(1.5 \times |Y^+|,\ |Y^+| + 10)$ new tokens, with a custom end-of-sequence condition that truncates generation to a single line.

When fewer than the target number of candidates survive the gauntlet, the shortfall is filled by resampling with replacement from the successful candidates, with duplicates permitted. When zero candidates survive, the example is skipped entirely, with no retry or regeneration of that step.

Selection scoring. For the selection score (Eq. 2), the implementation computes the *mean* per-token log probability over tokens from the completion only (prompt tokens are excluded), with no length penalty exponent beyond this simple mean. This is a practical length control: since candidates vary in token length, the mean approximates $\pi_0(y \mid H)$ while avoiding a systematic bias toward shorter or longer continuations. These log probabilities are computed through a separate, clean forward pass over each candidate's full token sequence using the model's raw, unmodified output distribution. That is, a plain softmax at $T{=}1$ with no truncation from top k or sampling. This decouples the scoring distribution from the sampling hyperparameters used during candidate generation. The semantic filter is applied before this scoring step to exclude candidates that are near paraphrases of Y^+ prior to selecting the highest-scoring survivor as Y^-.

Random and Semantic baselines. Distractors sourced through Random sampling draw actions from other videos, while Semantic sampling restricts the draw to an action with a shared object or other content noun. The shared noun is drawn primarily from Y^+ itself. If Y^+ yields no usable noun, it falls back to the preceding history or context. If that also fails, the condition falls through to unconstrained random sampling. In a representative production run with `LLaMA-3.2-11B-Vision` on 100 videos, 95.21% of examples used a genuine Y^+ noun match, 4.76% fell back to context nouns, and only 0.03% fell through to pure random selection.

Nouns are extracted with spaCy's small English pipeline `en_core_web_sm`, using only its part-of-speech tagger, keeping tokens tagged as nouns or proper nouns, lemmatized and lowercased, and filtered against a short stopword list. The same extraction procedure builds the candidate index and extracts nouns from ground truth and context strings.

Both Random and Semantic sample uniformly at random. Semantic additionally relaxes its required noun overlap count by one, down to zero, if too few candidates qualify at the configured minimum. Both controls exclude candidates from the same video and the exact ground truth string. Semantic additionally excludes, at indexing time, any candidate sentence with zero non-stopword nouns. Neither control applies participant-level exclusion. Candidate source and selection both differ from PDM, so these controls do not isolate ranking within a common generated pool.

Constructing chosen and rejected responses. The base model is the fixed DPO reference. Each verified pair supplies terminal actions for generated chosen and rejected responses, with this instruction for the rejected response.

> Generate a rationale containing a specific causal or physical reasoning error that leads to the wrong conclusion. The error should be subtle, and the reasoning should seem plausible to someone not paying close attention, but contain a specific physical or causal mistake.

The instruction leaves the nature and location of the reasoning error to the generator. The source specifies validation of the rejected final answer against Y^-. DPO scores both complete response sequences under H, excluding construction instructions that reveal the target outcome. Histories advance along recorded outcomes.

Explanation generation for PDM-DPO++. Explanations are generated by a separate, frozen, text-only reasoning model, `Qwen/Qwen3-14B`, distinct from the vision model being trained. The sole exception is the iterative self-play variant, which uses the currently training model itself as the explanation generator.

The chosen (Z^+) prompt instructs the generator to ground its reasoning exclusively in the supplied context, follow a clear causal chain from observation to conclusion, and end with the exact provided answer. The rejected (Z^-) prompt instructs the generator to produce an internally consistent but subtly flawed chain arriving at the incorrect distractor, grounded in the same context and format, leaving the location and nature of the physical or causal error to the generator.

Decoding for the explanation generator uses temperature 0.6 with sampling at cumulative probability 0.95, up to 1024 new tokens, and standard end-of-sequence stop conditions. A candidate explanation

is discarded if it is empty, omits a labeled step line, lacks an extractable labeled answer line, or its extracted answer does not match the intended target string. Discarded pairs are dropped with no fallback text and no automatic retry within a run.

Explanations are generated once and frozen prior to training in the static pipeline and are not regenerated per hyperparameter configuration. The iterative self-play variant instead regenerates a fresh explanation set once per self-play chunk, against that round's newly mined negatives, which is itself then frozen for that chunk. Note that the elaborate instructions above are used only to elicit the reasoning text, not the prompt seen at DPO training time. Across a representative production run, this filtering removed 717 of 103,159 candidate pairs (about 0.7%).

C.2 Training Protocol and Reproducibility Status

Next token loss baseline (AF). History is provided as a sliding window of 50 prior steps. Context tokens are masked from the loss, and only the answer continuation is scored. Training uses the same seed "1759" as the rest of the pipeline. Splits are constructed at the video level, so no video appears in both the train and validation portions.

Elbow thresholded baseline (AF and Threshold). Candidates are scored by a non-finetuned model, distinct from the model being trained. The score is the mean per-token cross-entropy over answer-only tokens, converted to a log-likelihood, passed through a softmax over all options for a given question, and compared against the groundtruth using a Brier score. This Brier-based score is what the elbow cutoff is applied to. The elbow point itself is found by a manual max perpendicular distance from the chord of the sorted score curve. This selection uses training data only. The specific production threshold used to retain 83.9% of examples was obtained from the results file of the LLaMA training run.

Multiple-choice discrimination. Discrimination is computed as the conditional probability of an option given the full option set. Context tokens are excluded from scoring through an index-based slice rather than a masked label array. The number of options is fixed at four options, with one ground truth and three distractors.

Hyperparameter search. Baseline training recipe settings (warmup steps, learning-rate schedule, batch size and gradient accumulation, and related optimizer settings shared across methods) were established through a separate hyperparameter search for the next-token-loss (AF) baseline, selected using accuracy, and then held fixed for the subsequent DPO search described below.

For DPO, we conducted a random hyperparameter search, seeded with 1759, generating all configurations in a single deterministic pass rather than sequentially or adaptively. The search evaluated 15 total configurations: one fixed baseline plus 14 random draws.

Hyperparameter	Range or distribution
β	log-uniform, 0.03–0.5
loss type	categorical: sigmoid, IPO
learning rate	log-uniform, 5×10^{-7}–2×10^{-5}
weight decay	categorical: $0.0, 0.0, 0.01, 0.05$ (0.0 upweighted $2\times$)
LoRA rank	categorical: $8, 16, 32, 64$
LoRA dropout	uniform, 0.0–0.15
LoRA scale	derived, always $2\times$ rank (not independently searched)

Warmup steps (40), the learning-rate scheduler (plain cosine, without restarts, restarts are used only in the final long production run), the validation split, seed, batch size, gradient accumulation steps (effective batch size 16), gradient clipping, the paged 8-bit AdamW optimizer, and the step budget were held fixed across every configuration rather than searched.

The winning configuration was selected using DPO's own internal reward-accuracy metric, and used $\beta = 0.33$, sigmoid loss, learning rate 1.912×10^{-5}, weight decay 0.0, LoRA rank 16, scale 32, and dropout 0.006.

PDM-DPO. Training uses $\beta = 0.33$, the paged 8-bit AdamW optimizer, learning rate 1.912×10^{-5}, weight decay 0.0, and gradient clipping of 1.0.

The reported PDM-DPO run trains for a single epoch, with an effective batch size of 16 (a per device batch size of one combined with 16 gradient accumulation steps), using approximately 103,000 preference pairs.

PDM-DPO++. Training uses the same optimizer settings as PDM-DPO: $\beta = 0.33$, the paged 8-bit AdamW optimizer, learning rate 1.912×10^{-5}, weight decay 0.0, and gradient clipping of 1.0.

Language attention query, key, value, and output projections receive adapters with rank 16, scale 32, and dropout 0.006, applied identically across both backbones and all reported PDM-DPO++ runs.

The prompt format used for the underlying chosen and rejected responses is:

> Context: {context}
> What is the most likely next action?

For runs that include generated reasoning, the chosen and rejected reasoning chains described earlier in Appendix C.1 are used as the literal training text.

No generation occurs during DPO training itself, since training operates on static, pre-tokenized text. Any reasoning chains used for training are produced offline, beforehand, using the decoding settings described in Appendix C.1. At evaluation time, generation is greedy and deterministic, with sampling disabled.

Test-time reasoning controls (ToT). ToT search uses a beam width of 3 and a maximum of 3 steps, and is evaluated at a single frame. At each depth, every current beam generates beam-width many new candidate continuations (nine candidates from three beams), each generated one at a time rather than in a batch to avoid holding multiple key-value caches simultaneously. Each candidate is scored by a separate, clean forward pass giving the base model's own log-probability for that step, using the model's raw, unmodified output distribution rather than the temperature- and nucleus-adjusted distribution used to sample the candidate. This score is added to its beam's running total. Beams are pruned back to the top three by this cumulative score before proceeding to the next depth. Candidate generation uses sampling with temperature 0.7 and nucleus sampling at cumulative probability 0.9, with one continuation returned per call.

The base prompt shares a common structure across all beam conditions, consisting of an image token prefix, a preamble, and the question text with any "answer directly" style boilerplate stripped out. In the single-frame setting used, the preamble instructs the model to analyze the visual content carefully and use its knowledge of the activity type to answer the question.

Each reasoning step is capped at 120 new tokens, and the final answer step is capped at 10 new tokens. In production, this amounts to up to 27 generation calls per question (beam width squared, times the number of depths), each run sequentially to control memory rather than batched.

Shared training settings. Training uses bf16 precision throughout, and we use 40 warmup steps with a cosine schedule that includes restarts, and no early stopping. The iterative self-play family instead uses 10 warmup steps, a plain cosine schedule, and an early stopping patience of five evaluations without improvement.

Checkpoint selection for reporting is based on the reward metric alone, consistent with the practice already described in the main text. The reference model used for DPO is the same frozen base model as the policy being trained, with its adapters disabled, rather than a separately loaded copy to better manage memory usage.

Scope of visual input. With three exceptions, training and evaluation described in the experiments operate on narrated textual histories only. This is an intentional scope choice, not an omission. The base checkpoints are vision-language models, and for every method outside these three exceptions, vision parameters are carried through and frozen rather than removed, with no video frames supplied as input.

The first exception is AF+Video, which trains on actual video frames with both the language and vision layers updated, included specifically as a comparison point against our approach.

The second exception is TemporalBench, where genuine frame-conditioned evaluation is used, with frames for the 1, 2, and 4-frame conditions selected as evenly spaced positions spanning the full length of the video, always including the first and last frame. This selection is a deterministic function of the video and the requested number of frames, with no randomness involved, and the same selection is used across every method and model evaluated at a given frame count, so comparisons at a fixed frame count use identical frames. The final evaluation prompt is the task question itself, optionally preceded by a short framing instruction when reasoning chains are used, and scoring is based on an exact match between the predicted and ground truth option label.

The third exception is OpenEQA, which also uses genuine frame-conditioned evaluation, sampling 20 frames by default from each episode's pre-recorded observation history. Frames are chosen at uniformly spaced positions spanning the full episode, from the first frame to the last inclusive, using the same underlying idea as TemporalBench's frame selection, though the two differ in how fractional indices are rounded, so the exact frame indices chosen are not guaranteed to be identical between the two even at the same requested frame count. OpenEQA evaluation here uses supplied observations rather than active exploration.

Proposer sharing and regeneration. For static runs, the proposer is specific to each backbone, so each backbone's distractors are generated by that backbone's own base checkpoint. For the iterative self-play variant, the proposer is instead the trainee's own evolving checkpoint from the previous chunk, and this applies identically to both the LLaMA and Qwen backbones, with proposals regenerated once per chunk against that chunk's current adapter. Static production runs generate proposals once, upfront, and freeze them for the remainder of training.

Dataset counts and splits. Reported dataset counts and split construction are treated identically across both backbones. Splits for all DPO training data are constructed at the video level, so that no video contributes examples to both the training and held-out portions, consistent with the approach used for the next token loss and multiple choice pipelines.

Elicitation prompts and likelihood normalization. Open elicitation under log-likelihood scoring uses no additional instruction text, scoring each candidate as a raw continuation of the context alone. Open elicitation under verbalized probability instead uses the following prompt.

> Context: {context}
> What is the probability (0.0 to 1.0) that the next sentence is
> "{option}"
> Answer with ONLY a decimal number between 0 and 1 (e.g., 0.75 or 0.3). No explanations
> or words.

Closed elicitation under log-likelihood scoring presents a lettered list of options and scores each option in its own slot. Closed elicitation under verbalized probability uses the following prompt.

> Context: {context}
> Consider these possible next actions
> {options}
> What is the probability (0.0 to 1.0) that option {label} is the correct next action?
> Answer with ONLY a decimal number...

Both log-likelihood modes normalize by mean, rather than sum, per-token cross-entropy, then pass the result through a numerically stable softmax over options. Verbalized modes instead parse a probability directly out of the generated text and rescale the parsed values to sum to one across options, without a softmax step. Responses that cannot be parsed as a probability are treated as a near-zero value rather than a neutral or missing value. One Brier sample unit is a single question and option pair, so a four-option question contributes four terms to the overall mean.

Run-level bookkeeping. All reported runs use a single training seed, 1759, with no multi-seed replication. Training was run on four NVIDIA RTX 6000 GPUs. Measured wall clock time varies

by run type, with evaluation runs typically completing in under two and a half hours, and the corresponding next token loss baseline completing in approximately four and a half hours.

C.3 Standard Errors and Across-Model Standard Deviations

BA and MBA are distinct metrics with different denominators. TemporalBench's MBA evaluator groups binary questions by video and counts a group as correct only when all of that video's questions are answered correctly. The individual data source counts are 1,542 EgoExo4D, 1,550 COIN, 1,571 Oops, 1,594 FineGym, 957 Charades, 1,467 Movie Description, and 1,186 ActivityNet, for 9,867 binary questions in total. The current figures use $n = 1{,}571$ for Oops, $1{,}594$ for FineGym, and $9{,}867$ for full TemporalBench at every reported frame count and for both backbones.

The MBA evaluator is a group-level metric, with the group count as its denominator rather than the question count. This distinguishes BA (9,867 questions) from MBA (2,179 groups), with official group counts of 307, 385, 294, 288, 298, 326, and 281 in the same dataset order. The two metrics therefore have different valid sample sizes for any standard error calculation.

Interpretation of the displayed error bars. For an accuracy of $100\hat{p}$, the displayed half width is $100\sqrt{\hat{p}(1-\hat{p})/n}$. This estimates one standard error when n counts independent Bernoulli outcomes. It represents neither a 95% confidence interval nor variation across training runs. In improvement plots, the observed zero-shot accuracy is subtracted from the point and both endpoints, so the bars treat the baseline as fixed. Standard errors of paired improvements require paired predictions and their covariance, which the available aggregate scores do not supply on their own. No average is reported across Oops, FineGym, and full TemporalBench, since the full benchmark includes both subsets.

Forecasting summary and OpenEQA. The forecasting summary shows the population standard deviation across the four or five available models per condition, rather than a standard error over items. OpenEQA reports graded scores, so even its recorded sample counts, 557 for A-EQA and 1,636 for EM-EQA, do not justify binomial bars. Training cost measurements have no reported repetitions, and every timing value reflects a single run.

D Numerical Results and Additional Diagnostics

Method	Oops	FineGym	Full TemporalBench
LLaMA-3.2-11B			
Zero-shot (MBA)	6.12	17.01	15.19
AF	-0.03	+0.35	-0.18
AF + Thresh	+1.70	-9.72	-2.98
MC	-0.34	-11.80	-4.31
CoT	-2.38	-12.15	-7.76
ToT	-0.22	-8.32	-1.57
PDM-DPO	-0.68	+0.35	+1.50
PDM-DPO++	+0.96	+1.24	+1.62
Qwen-2.5-VL-32B			
Zero-shot (MBA)	15.31	11.81	24.97
PDM-DPO	+0.00	+0.69	+0.41
PDM-DPO++	+0.74	+1.54	+0.86

Table 1: Zero-shot MBA followed by absolute changes in percentage points for Figures 4.

Dataset	Method	1 frame	2 frames	4 frames
LLaMA-3.2-11B				
Oops	Zero-shot	6.12	9.86	10.20
Oops	PDM-DPO++	7.08	12.22	11.54
FineGym	Zero-shot	17.01	17.36	17.71
FineGym	PDM-DPO++	18.25	19.40	20.10
Full TemporalBench	Zero-shot	15.19	17.30	18.22
Full TemporalBench	PDM-DPO++	16.81	18.81	19.54
Qwen-2.5-VL-32B				
Oops	Zero-shot	15.31	21.09	23.13
Oops	PDM-DPO++	16.05	22.11	23.63
FineGym	Zero-shot	11.81	16.67	13.19
FineGym	PDM-DPO++	13.35	16.92	14.89
Full TemporalBench	Zero-shot	24.97	29.19	32.49
Full TemporalBench	PDM-DPO++	25.83	29.28	33.04

Table 2: MBA percentages underlying Figure 9, reporting zero-shot and PDM-DPO++ scores for each dataset and frame count.

Frames	EgoExo4D	Charades	FineGym	Movie	Oops	ActivityNet	COIN	Overall
1	10.77	14.42	18.25	20.67	7.08	23.42	17.10	16.81
2	14.68	16.77	19.40	24.01	12.22	23.06	20.71	18.81
4	15.01	18.17	20.10	24.31	11.54	23.06	23.34	19.54
8	14.36	17.44	17.32	24.93	10.86	24.81	21.78	18.67

Table 3: Full TemporalBench frame sweep for PDM-DPO++ on LLaMA-3.2-11B. Overall reflects the pooled score across all 2,179 groups, weighted by each dataset's size, which differs from an unweighted average of the seven per-dataset percentages.

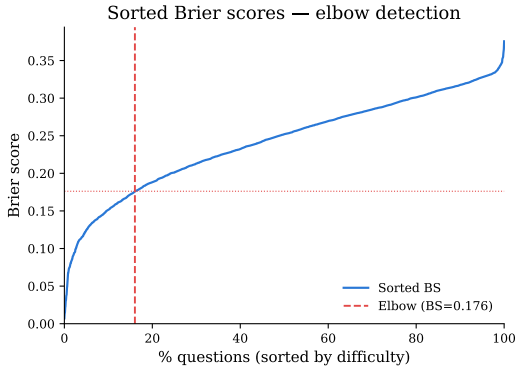


Figure 5: Elbow selection for AF+Thresh: sorted Brier scores across training examples, with the elbow marking an empirical cutoff of 0.176. This purely data-driven cutoff serves as a static proxy for the theoretical error threshold τ^* motivated in Section 3.2, without estimating the biological threshold itself.

D.1 Forecasting Elicitation and Model Breakdown

Open-ended judgments omit the competitor, while closed-ended comparisons supply it. We separately compare likelihood scores and verbalized probability reports, since elicitation need not produce interchangeable estimates.

Supplying the Competing Future. Appendix Figure 6 compares the closed and open formats on the same underlying pairs. Supplying the competing future increases likelihood-based Brier error for all five forecasters and all three constructions, with three exceptions under verbalized probabilities (Figure 7). A future that appears unlikely when scored alone can thus be difficult to reject in direct competition with the recorded outcome. Ordinary action forecasting does not measure this difficulty, which supports evaluating the candidate-conditioned risk in Eq. 3 separately.

Both open- and closed-ended probabilities are normalized identically: each candidate's per-token likelihood is scored and then passed through a softmax over the option set (Appendix C.2). The formats differ only in what the model conditions on while scoring, closed-ended presents all options together in context ($P(A \mid A, B, C, D)$-style), while open-ended scores each option independently before the softmax is applied, not in how the resulting probabilities are normalized. Since PDM-Eval compares exactly two candidates ($C = \{Y^+, Y^-\}$), the valid reference chance line for a Brier Skill Score here is $\mathrm{bs}_{\mathrm{ref}} = \frac{1}{2}(1 - \frac{1}{2}) = 0.25$.

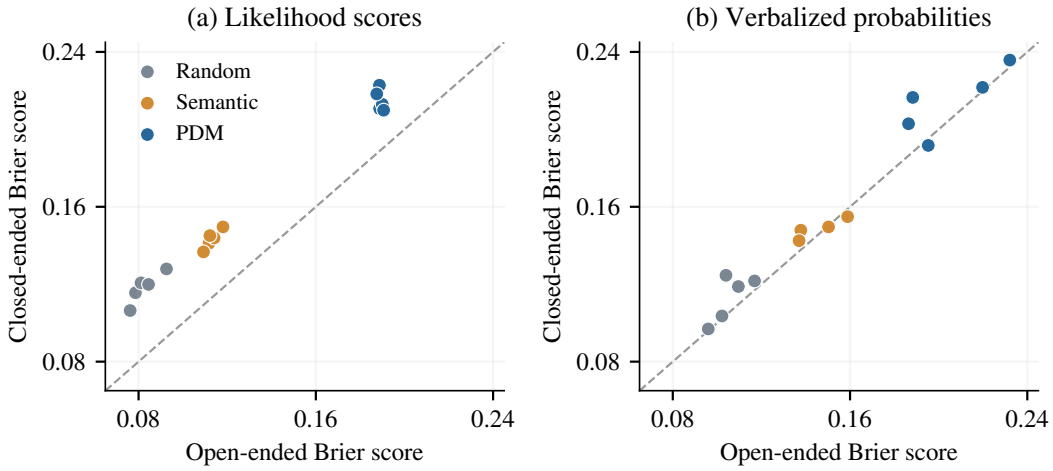


Figure 6: Independent versus joint elicitation, with one point per model/construction combination and a diagonal marking equal Brier error.

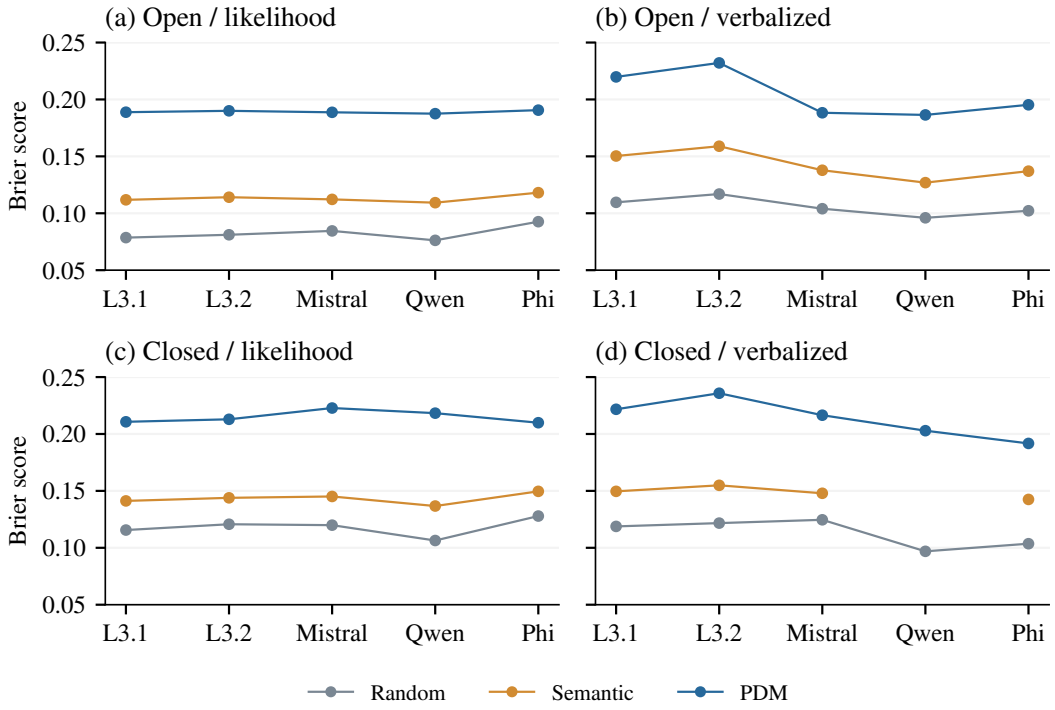


Figure 7: Forecasting comparisons by model, plotted as Brier error. PDM yields the largest error for every model and elicitation condition, showing the difficulty is not specific to any single forecaster.

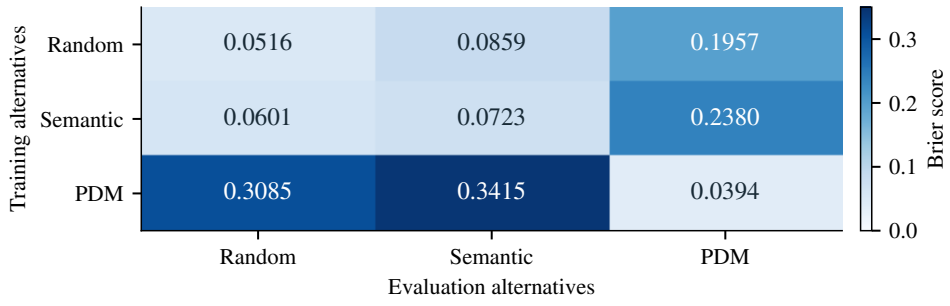


Figure 8: LLaMA-3.2-11B MC discrimination, with training constructions in rows and evaluation constructions in columns. PDM training reduces its own Brier error but transfers poorly to other constructions.

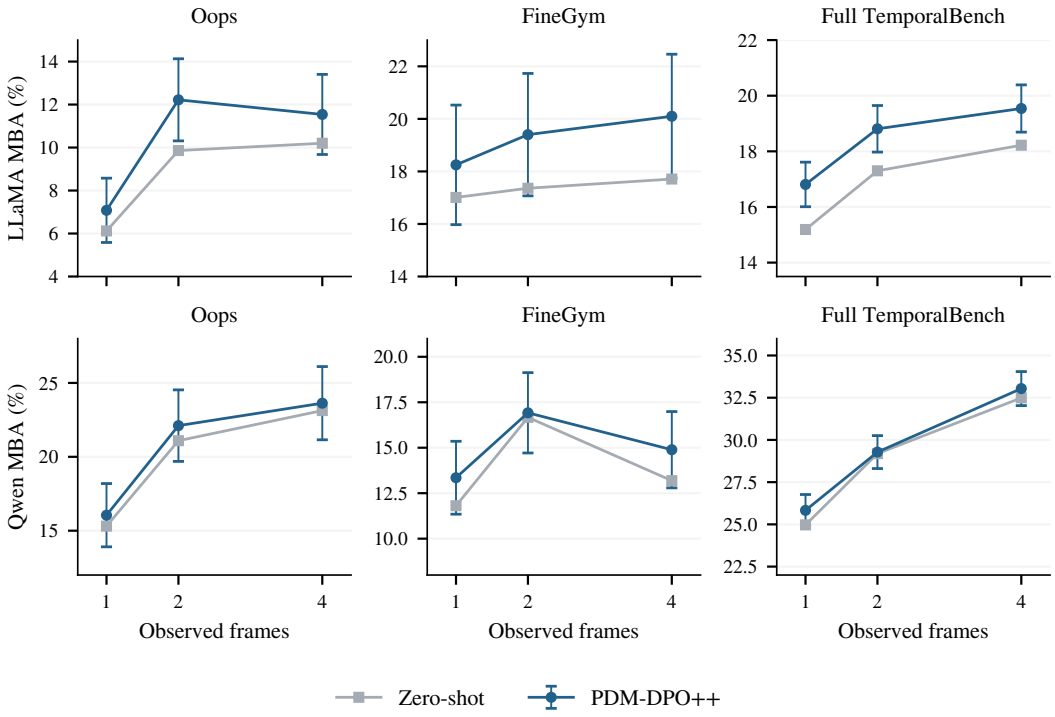


Figure 9: MBA versus observed frames on Oops, FineGym, and full TemporalBench. The top row is LLaMA-3.2-11B and the bottom row is Qwen-2.5-VL-32B. Panels use separate accuracy ranges. PDM-DPO++ exceeds zero-shot at every frame count for both backbones. Frame selection is described in Appendix C.2.

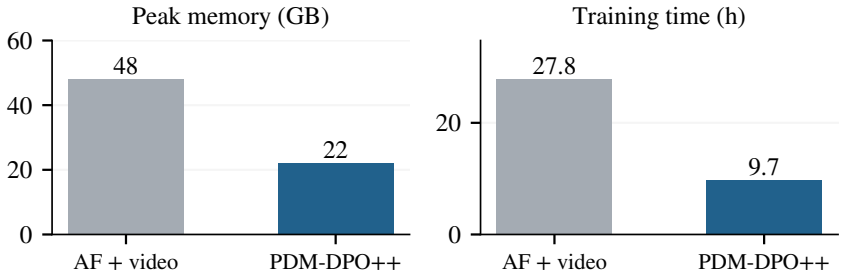


Figure 10: Training cost on LLaMA. AF + video trains LoRA in both language and vision layers, while PDM-DPO++ uses narrations to update language layers. Repeated measurements are unavailable, so no error bars are shown.

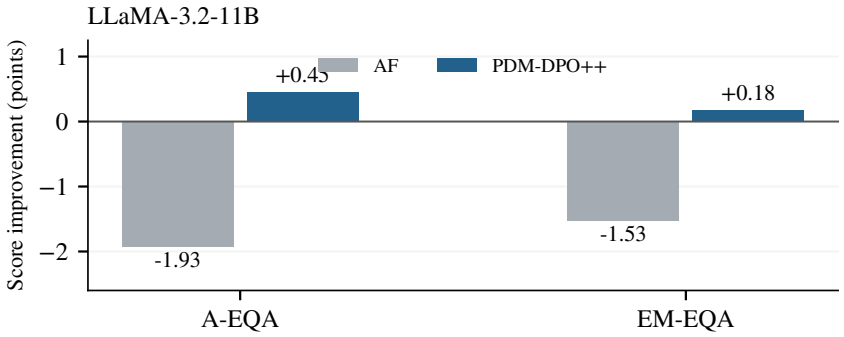


Figure 11: LLaMA OpenEQA score changes from zero-shot for AF and PDM-DPO++.