BETTER SAMPLES, WRONG ODDS: WHEN PREDICTION REWARDS DISTORT WORLD-MODEL PROBABILITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Prediction rewards post-train world models to produce better samples; we study when they preserve the probabilities a planner needs across futures and when they distort them. A prediction reward, which we call a point reward, compares one generated future with one observed outcome. Under such a reward a model can score below the truth's own risk only with wrong probabilities, which gives a checkable diagnostic: for any metric, the excess gain—the part of the gain that lies below the truth's own risk—lower-bounds the Wasserstein-1 distance to the truth. In PushT, where all futures can be enumerated, about 72% of the gain from point-reward training is excess gain. On RLVR-World's RT-1 video world model, point rewards lower pixel error but give a worse held-out joint energy score than an energy-score reward from the same base, data, and budget in all five seeds, by 2.1% of Base's score at 200 updates and 7.2% at an exploratory 800. The distortion reaches decisions when action rankings depend on how probability is split: in an image-conditioned MiniGrid task, it costs 10.3 points of success, and continuing with a proper score recovers most of it. Proper scoring alone is not sufficient: the update rule also matters. Under idealized GRPO standardization, energy-score and point-reward updates coincide for binary outcomes, while unbiased leave-one-out credits keep the truth a fixed point. We therefore recommend reporting a proper score beside sample metrics and training proper rewards with unbiased credits.

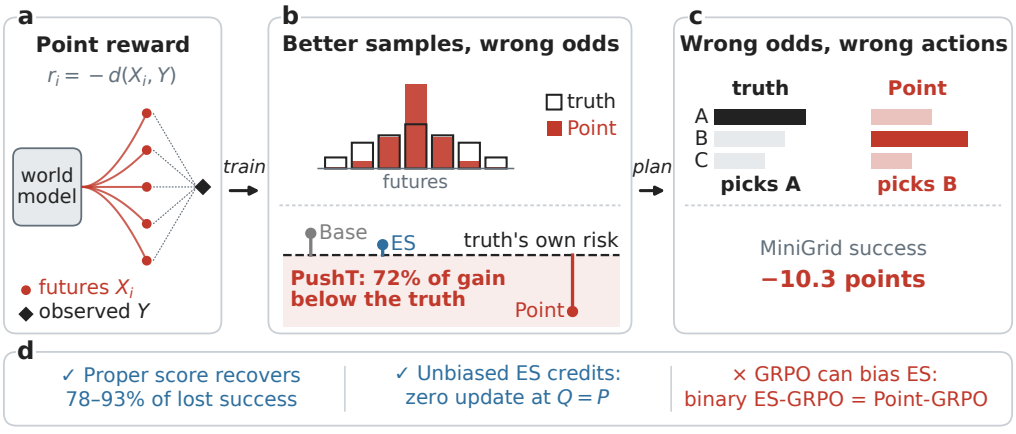


Figure 1: Overview. Elements above the dotted lines in (b) and (c) are schematic; those below are measured. (a) A point reward compares one model future with one observed outcome. (b) Point training can concentrate mass: in PushT, 72% of its point-risk gain lies below the truth's own risk, possible only with wrong probabilities (Lemma 1; Section 3.1). (c) Wrong odds change the planner's choice when action rankings depend on probability splits (Section 4.2). (d) Proper-score continuation recovers 78–93% of lost success in FrozenLake and MiniGrid; under idealized standardization, binary energy-score GRPO equals point-reward GRPO (Proposition 1).

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.

1 INTRODUCTION

World models support policy learning in imagination, learned-model search, and action-conditioned simulation (Sutton, 1990; Hafner et al., 2020; 2025; Micheli et al., 2023; Schrittwieser et al., 2020), with applications in robot learning, autonomous driving, and interactive environments (Wu et al., 2022; Hu et al., 2023; Bruce et al., 2024). In a stochastic environment, a planner needs plausible futures and also the right probabilities over them. Recent work post-trains world models with prediction rewards, following group relative policy optimization (GRPO) (Shao et al., 2024b) and reinforcement learning with verifiable rewards (Lambert et al., 2025), and reports better sample metrics and downstream control (Wu et al., 2025). A proper scoring rule gives a less favorable picture of the released RLVR-World RT-1 checkpoint. Its pixel error is lower than that of its base model, but its joint energy score over future videos is 26.5% worse and its negative log likelihood is 83% worse (Section 3.2). Its data and schedule also differ, so Section 3.2 isolates the reward in a controlled comparison.

A prediction reward, which we call a *point reward*, compares one generated future with one observed outcome. It therefore favors futures with low expected error, whether or not they appear at their true frequencies. One consequence is that a trained model can score better than the true distribution. The truth has a point risk of its own, the expected distance between two independent real outcomes, and a model can go below this value only by assigning the wrong probabilities (Lemma 1). In PushT, where every future can be enumerated, point-reward training does this, and about 72% of its gain lies below the truth's own risk (Figure 1b).

Proper scoring rules distinguish point and distributional prediction (Gneiting & Raftery, 2007; Gneiting et al., 2008). Prior work finds misleading trajectory sample metrics (Shahroudi et al., 2024), suppressed stochastic spread under pointwise errors (Zhou et al., 2026), video-world-model miscalibration (Sadeghi et al., 2026), and GRPO bias under proper rewards (Bereket & Leskovec, 2025). It has not been shown whether this difference appears when world models are post-trained with prediction rewards, whether it affects a planner, and whether a proper reward removes it.

1. **Point rewards push models below the truth's own risk.** This happens in PushT and FrozenLake, where the truth is enumerable, and direct probability metrics show that the models get worse. On RT-1, against an energy-score reward with the same base, data, and budget, point rewards give a worse held-out joint energy score in all five seeds, by 2.1% of Base's score after 200 updates and 7.2% after an exploratory 800. At a matched KL to the starting checkpoint, point rewards are also worse on two proper scores that neither reward optimizes (Sections 2.1–3).
2. **The distortion costs decisions when action rankings depend on how probability is split.** In a constructed task where every model predicts the true mean, distribution shape alone costs 17.6 points of success. Point rewards lower success by 10.3 points in a visual MiniGrid task and exact success by 1.7–2.1 points under two FrozenLake dynamics, and continuing with likelihood or the energy score recovers 78–93% of the loss. In PushT, decisions barely depend on the split, and we find no task harm (Section 4).
3. **The update rule can undo a proper reward.** Under GRPO's group standardization, energy-score updates need not vanish at the truth, and for binary outcomes they equal point-reward updates (Proposition 1). In few-outcome FrozenLake, the energy score trained with GRPO lowers success in all ten seeds, while unbiased leave-one-out credits lose about 0.1 points. On RT-1 this gap is absent at 200 updates and appears by 800 (Section 5).

2 SETUP

2.1 HOW A MODEL CAN OUTSCORE THE TRUTH

Conditional prediction. Let c contain the state or observation history and the prescribed actions. Let P_c denote the true future distribution and $Q_c = Q_\theta(\cdot \mid c)$ the model distribution. Futures under the same condition differ through environmental randomness. For a fixed distance d and independent draws $X \sim Q_c$ and $Y \sim P_c$, define

$$g_c(x) = \mathbb{E}_{Y \sim P_c}[d(x, Y)], \qquad L_{\text{point}}(Q_c) = \mathbb{E}_{X \sim Q_c}[g_c(X)]. \tag{1}$$

A *point reward* assigns $-d(X, Y)$ to one generated future, and the model may remain stochastic. This risk is linear in Q_c, so its unconstrained minimizers are supported on the minimizers of g_c and need not reproduce the probabilities in P_c (Gneiting, 2011).

The truth's own risk. The true distribution has a point risk of its own, $L_{\text{point}}(P_c) = \mathbb{E}_{Y,Y' \sim P_c}[d(Y, Y')]$ with independent Y, Y', which we call the *truth's own risk*. A model can score below it. With outcomes $\{0, 1\}$, $P = (0.7, 0.3)$, and $d(x, y) = |x - y|$, the truth's own risk is $2 \cdot 0.7 \cdot 0.3 = 0.42$, while a model that always predicts 0 scores 0.3 and assigns zero probability to an outcome that occurs 30% of the time. Lemma 1 shows that any model below the truth's own risk is wrong. Write L_0, L_Q, and L_P for the point risks of the starting model, the updated model, and the truth. For an improvement $\Delta = L_0 - L_Q > 0$, we call the part that lies below L_P the *excess gain*,

$$\Delta_{\text{excess}} = \min\{\Delta, [L_P - L_Q]_+\}, \tag{2}$$

with $[z]_+ = \max(z, 0)$. When $L_0 \geq L_P$, this is simply the reduction below L_P.

> **Lemma 1 (excess gain lower-bounds transport distance).** Let d be a metric (or a pseudometric on features) and let P_c and Q_c have finite first moments. Then $L_{\text{point}}(P_c) - L_{\text{point}}(Q_c) \leq W_1(P_c, Q_c)$, the Wasserstein-1 distance under d, and hence $\Delta_{\text{excess}} \leq \mathbb{E}_c[W_1(P_c, Q_c)]$.

Proof. By the triangle inequality, $g_c(x) - g_c(x') = \mathbb{E}_Y[d(x, Y) - d(x', Y)] \leq d(x, x')$, so g_c is 1-Lipschitz. Kantorovich–Rubinstein duality gives $\mathbb{E}_{P_c}[g_c] - \mathbb{E}_{Q_c}[g_c] \leq W_1(P_c, Q_c)$. Averaging over conditions and using $\Delta_{\text{excess}} \leq [L_P - L_Q]_+$ completes the proof. □

When d is bounded by D, $W_1 \leq D \cdot \text{TV}$ and Pinsker's inequality give $\Delta_{\text{excess}} \leq D\,\mathbb{E}_c\big[\sqrt{D_{\text{KL}}(P_c \| Q_c)/2}\big]$, so any excess gain also implies a nonzero KL to the truth. Excess gain shows that the model has moved away from the truth. It does not show that the model is worse than before, which Sections 3 and 4 test directly. Because $L_P = \mathbb{E}[d(Y, Y')]$, an unbiased estimate needs only two independent outcomes per condition, so excess gain can be computed wherever a condition can be replayed, as in PushT and FrozenLake.

KL regularization and proper scoring. With a Kullback–Leibler (KL) penalty $\beta D_{\text{KL}}(Q_c \| Q_{0,c})$, the unrestricted optimum within the support of $Q_{0,c}$ is

$$Q_c^*(x) \propto Q_{0,c}(x) \exp[-g_c(x)/\beta]. \tag{3}$$

This standard exponential tilt (Ziegler et al., 2019; GX-Chen et al., 2026) predicts the direction in which probability moves, although finite training need not reach the optimum. The energy score (ES) adds a distance between independent model draws,

$$L_{\text{ES}}(Q_c; P_c) = \mathbb{E}[d(X, Y)] - \tfrac{1}{2}\,\mathbb{E}[d(X, X')]. \tag{4}$$

For Euclidean distance and finite first moments, the true distribution minimizes this objective in the chosen feature space (Gneiting & Raftery, 2007; Gneiting et al., 2008). This makes ES a *proper reward*, and we use it as the main control. GRPO training can break this property (Section 5).

2.2 Experimental design

Table 1 lists the systems and what each one tests. Within each comparison, arms share a starting checkpoint (Base), data, budget, and decision interface. We report 95% paired bootstrap intervals over roots (start states from which futures are predicted), maps, or episodes. These intervals are conditional on the trained models and omit training variability, which we examine separately with repeated training seeds, following Agarwal et al. (2021). Arms are named by their objective (Point, MLE, ES). Point→X continues the Point checkpoint with objective X for an equal budget, and the suffixes -GRPO and -LOO name the credit rule of Section 5. Appendix Table 2 lists all names.

Prediction comparisons. In our stochastic version of PushT (Chi et al., 2023), each of 12 commands repeats the previous command with probability 0.25. Enumerating all 4,096 repeat patterns gives the conditional future distribution. A 5M-parameter autoregressive model is post-trained as Point, MLE, ES, or Point-track, a Point arm whose KL weight is tuned so that its final KL to Base matches that of ES. Each comparison is repeated with three optimization seeds (Appendix C).

Table 1: What each system tests. Where the distribution is enumerable or exact, we check probabilities directly. Where only recorded futures exist, we compare held-out scores. Task success is evaluated only where a decision interface exists.

System	Truth available	Question addressed
PushT, sticky commands	All 4,096 repeat patterns	Does point-reward training drive the model below the truth's own risk?
RT-1, controlled	One recorded future	Does the objective contrast persist in a video model?
RT-1, released	One recorded future	Is the controlled pattern consistent with released models?
FrozenLake	Exact transition kernel and task value	Do probability changes affect planning and recovery?
MiniGrid	Simulator outcomes	Does the effect reach a visual decision task?
Two-channel task	Exact; fixed true mean	Can distribution shape alone change decisions?

On RT-1 we start from RLVR-World's official `rt1-world-model-multi-step-base` checkpoint (Wu et al., 2025), trained on RT-1 data (Brohan et al., 2023), and post-train on 256 video windows. The arms cross the reward, a point distance or the energy score, with the credit rule, GRPO or leave-one-out (LOO). MAE-GRPO keeps the pixel mean absolute error (MAE) term of the released recipe's MAE+LPIPS reward (Zhang et al., 2018). LOO credits are rescaled to match GRPO's typical step size, and unscaled copies of both LOO arms check this choice (Appendix Table 6). All arms are trained for 200 updates and evaluated on 256 held-out episodes with 16 model futures per condition. MAE-GRPO has three seeds at this endpoint and the other arms five (Appendix D).

Decision comparisons. FrozenLake uses Gymnasium's slippery 8×8 environment (Towers et al., 2024) with exact 200-step planning and success evaluation. We use two dynamics, Gymnasium's slippery transitions (slip) and the same with a 25% chance of repeating the previous action (slip+hold). Models are trained on one map per dynamics, and five fresh seeds repeat training. We also rerun the whole slip chain on 16 new maps and evaluate the slip+hold models on 16 unseen maps (Appendix E). In MiniGrid Dynamic Obstacles (Chevalier-Boisvert et al., 2023), an image-conditioned model predicts obstacle maps that are rendered as RGB frames, and PointRGB is the point reward on these frames. A fixed planner ranks four routes using 64 model futures per route, and three Base seeds are each evaluated on 512 roots (Appendix F). A constructed two-channel task fixes every model's predicted mean, so only the shape of the distribution can change (Appendix G).

Metrics. Where the token-level truth P_c^τ is enumerable, $\mathrm{KL}(P \,\|\, Q)$ denotes $D_{\mathrm{KL}}(P_c^\tau \| Q_c)$ computed with unrenormalized model probabilities, so mass outside the true support counts. On RT-1, the main score is *joint ES*, the energy score of the whole predicted future video flattened into one vector, under root-mean-square pixel distance. Held-out proper scores and negative log likelihood (NLL) compare predictions without access to each conditional truth. RT-1 results, including contrasts between two post-trained arms, are percentages of Base's score. All differences are signed so that positive means worse, and task effects are reported as harm, Base minus arm success in points.

3 Point rewards improve samples and worsen probabilities

Where the truth is enumerable, point-reward training pushes the model below the truth's own risk. On RT-1, where the truth is unknown, a controlled comparison ties the trade to the reward.

3.1 PushT: the model falls below the truth's own risk

On the 64 PushT roots used to fit Base, Point takes point risk from above the truth's own risk to well below it (0.0457 to 0.0299, against 0.0413), so about 72% of its gain is excess gain; MLE and ES stay above the truth's risk (Appendix Figure A1). The lower risk comes with worse probabilities: $\mathrm{KL}(P \,\|\, Q)$ rises from 2.70 to 4.53, and the error of physical-event probabilities nearly triples (Table 4).

Two controls locate the damage. It is not explained by moving farther from Base: Point-track, whose final KL to Base matches ES's, still has higher event error than ES (by 0.0026, $[0.0018, 0.0035]$),

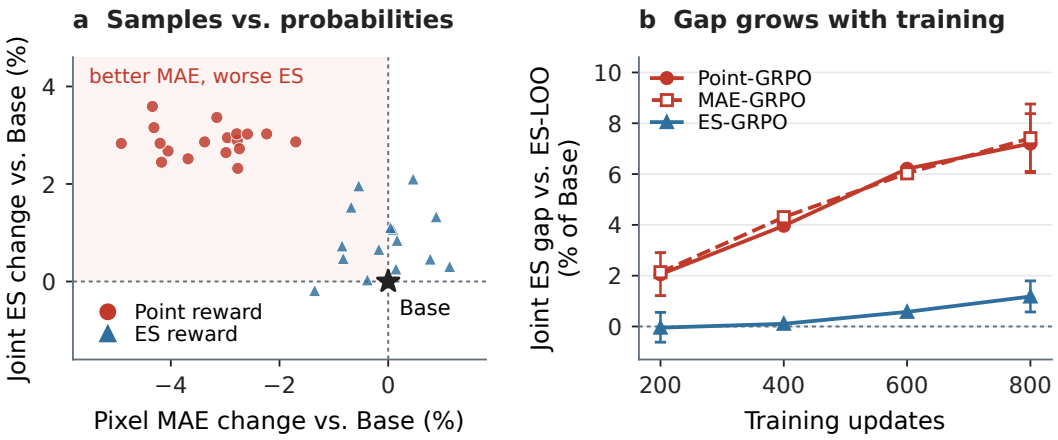


Figure 2: RT-1: point rewards trade probability fidelity for pixel accuracy. (a) At 200 updates, each marker is one arm and seed on 256 held-out episodes. Point-type arms (circles) lower MAE and worsen joint ES, and ES arms (triangles) stay near Base. (b) Joint-ES gap of each GRPO arm against ES-LOO over training, as a percentage of Base. Curves average five seeds (MAE-GRPO has three at 200 updates), and intervals are episode-paired 95% bootstrap intervals.

and this contrast replicates on two fresh Bases (Appendix C). Nor is it mass on impossible futures: Point raises its mass on the true support from 0.56 to 0.76 while roughly doubling the KL within it, so it misallocates probability among valid futures.

3.2 RT-1: THE TRADE GROWS WITH TRAINING

Released checkpoint. Evaluated in the same way as our arms, the released RLVR-World checkpoint improves pixel MAE and LPIPS relative to its base, while joint ES worsens by 26.5% and NLL by 83% (Appendix Table 7). Because it was trained with its own data and schedule, we repeat the comparison under control and change only the reward and the credit rule.

Controlled comparison. At 200 updates, point rewards trade probability fidelity for pixel accuracy: Point-GRPO is 3.6% better than ES-LOO in MAE but 2.1% worse in held-out joint ES ($[1.2, 2.9]$, all five seeds). The split follows the reward, not the credit rule: the four point-type arms, with GRPO or LOO credits and with or without scaling, form one cluster with lower MAE and worse joint ES, and the three ES arms form another near Base (Figure 2a; Appendix Table 7).

Matched divergence and unoptimized scores. Point-GRPO also moves farther from Base than ES-LOO, so part of this gap could reflect divergence alone. The two LOO arms rule this out at 200 updates. Their training KL to Base is matched (16.7 versus 16.2 nats per sequence over the last 20 updates), yet Point-LOO is worse on two proper scores that neither arm is trained on, a Gaussian kernel score (Gneiting & Raftery, 2007) ($+2.6\%$) and a temporal variogram score (Scheuerer & Hamill, 2015) ($+2.2\%$). Held-out NLL does not separate the pair (Appendix Figure A5, Table 8).

Longer training. The gap grows with training, to $+7.2\%$ ($[6.1, 8.4]$) at 800 updates, as Point-GRPO's MAE gain over Base doubles (Figure 2b). This longer run is exploratory; the training divergence of both arms grows by this horizon (Appendix Table 8).

4 THE DISTORTION CAN REACH DECISIONS

A planner is hurt only when the distortion changes which action ranks first. With the decision interface fixed and only the objective changed, point rewards lower task success in FrozenLake and MiniGrid, and continuing with a proper score recovers most of the loss (Figure 3).

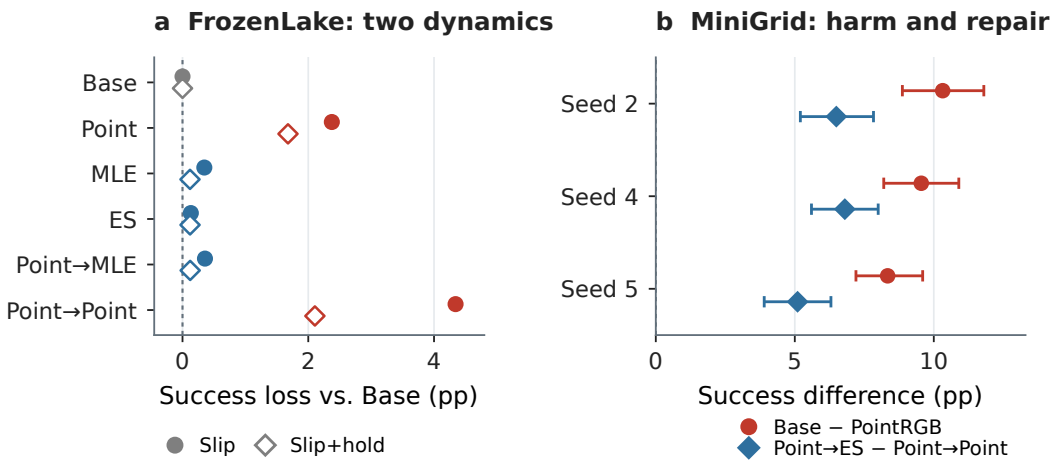


Figure 3: Decision effects and continuation controls. (a) FrozenLake exact 200-step success loss relative to Base on 16 new maps per dynamics (slip: the chain rerun on each map; slip+hold: the trained models on unseen maps). Circles are slip and open diamonds slip+hold. Positive values mean lower success, and task values are exact. (b) MiniGrid: harm is Base minus PointRGB; the continuation advantage is Point→ES minus equal-budget Point→Point. Bars are paired 95% root-bootstrap intervals. Seeds 4 and 5 share their roots.

4.1 FrozenLake: an exact chain from reward to success

In FrozenLake the transition kernel, planner, and 200-step success are exact. Each Base is within 0.1% of the truth's own risk, and Point ends 1.8–4.4% below it, so nearly all of Point's gain is excess gain.

Under slip, Point changes the planner's first action and lowers success on the training map from 91.3% to 89.2%, while MLE and ES from the same Base lose almost nothing. Continuing with likelihood (Point→MLE) recovers 87% of the loss; an equal-budget Point→Point continuation loses more. The harm holds across five fresh training seeds and when the whole chain is rerun on 16 new maps. Under slip+hold, the trained models lose 1.7 points on 16 unseen maps ($[1.0, 2.5]$), of which Point→MLE recovers 93% (Figure 3a, Appendix Table 10).

Most of this harm comes from errors in the predicted means, which are themselves a misallocation of probability across outcomes: replacing Point's means with the true ones largely restores success (Appendix E). Section 4.3 shows that shape alone can also matter.

4.2 MiniGrid: a visual decision task

On 512 unseen roots, PointRGB lowers future-frame RGB error by 17.5% while worsening token NLL and the Brier score of its success forecasts. It changes the chosen route on most roots and lowers success by 10.3 points ($[8.9, 11.8]$), whereas MLE from the same Base loses none. Continuing with a full-RGB energy score (Point→ES) recovers 91% of the loss, well beyond an equal-budget Point→Point continuation, and gives back almost all of the pixel-error gain. Two further Base seeds agree (Appendix F).

4.3 When distortion matters for decisions

Shape with a fixed mean. In a constructed two-channel task, symmetry fixes every model's predicted mean at the true value. Point still lowers exact success from 78.6% to 61.0% (Figure 4). To reduce its risk, Point moves mass to the center outcome of each channel. This understates the left channel, which succeeds at its side exits, and the planner switches to the right channel (Appendix G).

Tasks without harm. Stochastic dynamics alone are not enough. In PushT, complete-task harm is only 0.2 points ($[0, 0.6]$): the planner chooses among a few fixed candidates, and even an oracle

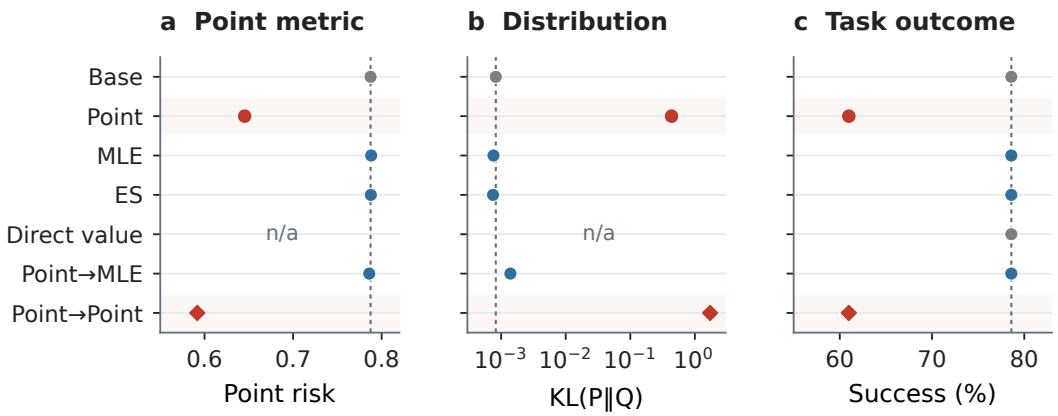


Figure 4: Distribution shape alone can change decisions. In a constructed two-channel task, symmetry fixes every model's predicted mean at the truth. Point lowers point risk (a) while raising $\mathrm{KL}(P \,\|\, Q)$ (b, log scale) and lowering exact success (c). Point→MLE restores success and Point→Point does not. Values are means over eight training runs on 64 held-out roots. The direct value baseline has no predictive distribution. The paired harm interval is in Appendix Table 13.

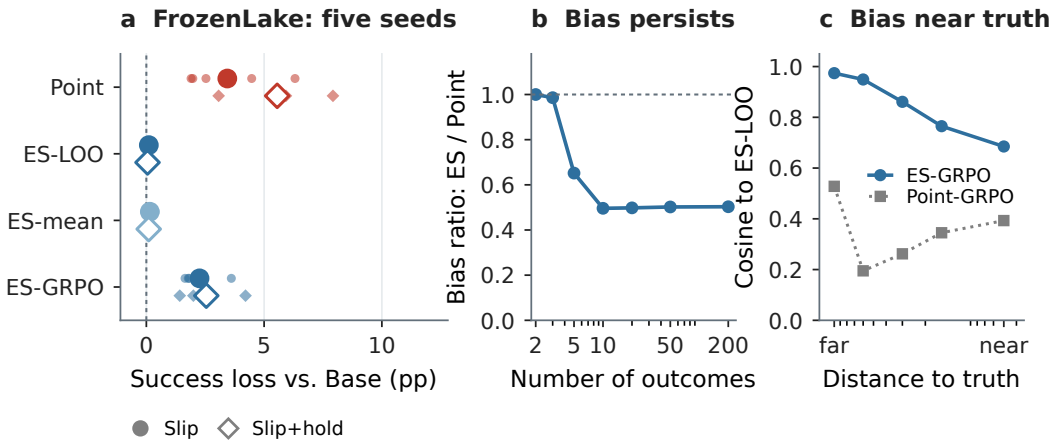


Figure 5: GRPO standardization weakens the energy-score correction. (a) FrozenLake success loss relative to Base at $K = 8$, where positive is harm. Small marks show five training seeds under each dynamics and large marks their means. Circles are slip and open diamonds slip+hold. (b) Toy expected-update norm ratio at the truth across outcome counts. (c) Alignment with the ES-LOO update along a path toward the toy truth. The RT-1 training-length contrast is shown in Figure 2b.

choice among them beats a probability-free one by under half a point. In Blackjack, MountainCar with a 25% action hold, and slippery CliffWalking, even an idealized distortion, planning with the exponentially tilted true kernel of Equation 3, barely changes returns, so we do not train models there (Appendix G). Harm requires an action ranking that depends on how probability is split.

5 The update rule can undo a proper reward

Proper-score continuation repairs much of the damage in Section 4, but a proper objective does not ensure a proper update, and GRPO's group standardization can break it. For binary outcomes, idealized standardization makes the energy-score update identical to the point-reward update. The resulting bias is largest when outcomes are few, and in a many-outcome toy it persists near the truth (Figure 5).

5.1 THE ENERGY-SCORE ESTIMATOR AND GROUP STANDARDIZATION

For $K \geq 2$ independent model futures and one independent observed outcome, the ES credit (advantage) of sample i is

$$r_i = -d(X_i, Y) + \frac{1}{K-1} \sum_{j \neq i} d(X_i, X_j). \tag{5}$$

It keeps the pairwise terms involving X_i after a sample-independent baseline is removed (Williams, 1992; Kool et al., 2019). Multiplying it by $\nabla \log Q(X_i)$ gives an unbiased estimate of the negative ES gradient. GRPO instead uses $(r_i - \bar{r})/s$, with group mean $\bar{r}$ and sample standard deviation s (Shao et al., 2024b).

> **Proposition 1 (energy-score credits under group standardization).** Under the regularity conditions in Appendix B: (a) the expected raw-credit update equals $-\nabla L_{\mathrm{ES}}$ and vanishes at $Q = P$. (b) The standardized update need not vanish there. For outcomes $\{0, 1, 2\}$, $P = (0.2, 0.5, 0.3)$, $K = 8$, and absolute distance, its expected logit update is $(-0.059, +0.135, -0.076)$. (c) For binary outcomes, idealized within-group standardization gives identical ES and point-reward updates. Constant-reward groups receive zero updates.

The pairwise term is part of the unbiased estimator and cannot be dropped as a baseline. Even subtracting only the group mean biases these coupled credits (Appendix B). How much harm this causes depends on the number of outcomes and on the distance to the truth.

5.2 FEW OUTCOMES: ES UNDER GRPO BEHAVES LIKE POINT

Proposition 1(c) is the extreme case, in which ES under GRPO is Point under GRPO. In a discretized bimodal toy (Appendix B), the bias shrinks only partly as the number of outcomes grows. At the truth, the expected ES-GRPO update is as large as Point-GRPO's for two or three outcomes, 0.65 of it for five, and about half of it from 10 to 200 outcomes (Figure 5b).

FrozenLake transitions have three or six outcomes, so this bias should matter there, and it does. At $K = 8$, ES-GRPO lowers success by 2.3 points under slip and 2.5 under slip+hold, harming all ten seeds, and like Point it drives point risk below the truth's own risk. The standard-deviation scaling is the main cause: ES-LOO and ES-mean, which subtracts the group mean but does not rescale, each lose at most 0.15 points on average. Nor is the shift a matter of step size, since with a 10–100$\times$ weaker KL penalty ES-LOO's point risk ranges from 0.08% below to 0.17% above Base's (Appendix E.2).

5.3 MANY OUTCOMES: THE BIAS APPEARS NEAR THE TRUTH

In the many-outcome toy, ES-GRPO departs from the ES-LOO update as it nears the truth. Along a path toward the toy truth, its cosine with the ES-LOO update falls from 0.97 to 0.69. At the truth, the ES-LOO update vanishes and the ES-GRPO update does not (Figure 5c).

On RT-1, ES-GRPO matches ES-LOO at 200 updates and falls behind with longer training, by 1.2% ($[0.6, 1.8]$, all five seeds) at 800 updates (Figure 2b). This is consistent with the toy but is not evidence that the RT-1 models near the truth, since every arm stays worse than Base on joint ES; ES-GRPO has also moved farther from Base (Appendix D).

6 RELATED WORK

Model objectives and decisions. A model's training objective need not match its use in control. Likelihood-trained models can plan poorly (Lambert et al., 2020), and value-aware and value-equivalent objectives train models for the decisions they support (Farahmand et al., 2017; Grimm et al., 2020). We study the reverse case, a post-training reward (Wu et al., 2025) that improves sample metrics while moving probabilities, and with them decisions, away from the truth. Proxy-reward overoptimization (Gao et al., 2023; Skalse et al., 2022) and reduced diversity under RLHF (Kirk et al., 2024) are related, but we focus on probability fidelity among valid futures.

Calibration and proper scoring. Calibration methods (Guo et al., 2017; Lakshminarayanan et al., 2017; Kuleshov et al., 2018) separate accuracy from probability correctness, and generative precision–recall (Kynkäänniemi et al., 2019) separates sample fidelity from support coverage; we ask instead how a post-training reward moves probability among valid futures. CaliBench measures miscalibration in video world models (Sadeghi et al., 2026). Energy scores evaluate and train distributional models (Shahroudi et al., 2024; De Bortoli et al., 2025); sample-access proper losses also apply to discrete generators and language generation (Kimpara et al., 2023; Shao et al., 2024a). We use these tools rather than propose them.

GRPO and sharpening. GRPO miscalibrates binary probability reports under log rewards and shows a similar bias under Brier rewards (Bereket & Leskovec, 2025). Fan et al. (2026) analyze finite-batch sampling and parameter coupling, and Dr. GRPO removes standard-deviation normalization (Liu et al., 2025). We show that for ES credits over generated futures, pairwise coupling makes even mean-only centering biased and binary ES updates equal to point-reward ones.

7 Discussion and conclusion

A point reward can prefer a distribution other than the truth even under unbiased optimization. Where the truth is enumerable, trained models fall below the truth's own risk, and on RT-1 the effect appears in proper scores at matched divergence. The distortion lowers task success when decisions depend on how probability is split, and continuing with a proper score recovers most of the loss. This correction depends on the update rule. For binary outcomes, Proposition 1 shows that idealized group standardization removes it exactly. In FrozenLake, whose transitions have three or six outcomes, ES-GRPO lowers success in all ten seeds, while unbiased credits lose about 0.1 points on average. On RT-1, ES-GRPO matches ES-LOO at 200 updates and trails it by 1.2% at 800.

Implications for evaluation and training. We recommend reporting a proper score next to each sample metric, together with a same-budget likelihood or ES control. When a condition can be replayed, excess gain is also cheap to report and flags gains that the truth itself could not achieve. For score-function ES training, the raw credits of Equation 5 should be kept or replaced by another unbiased estimator. Mixing a proper score or likelihood into a point reward trades retained sample gain against task harm along a smooth frontier (Appendix H).

Scope and limitations. Our decision evidence comes from systems where task value can be computed exactly or replayed. Extending it to physical robots requires a planner that uses the model's probabilities. In MiniGrid, two other variants of the setup did not separate the objectives clearly, and the reported one trains only the prediction head (Appendix F). On RT-1, we compare objectives rather than improve on Base. No post-trained arm, including MLE ($+2.6\%$), beats Base on joint ES, and held-out likelihood separates the arms less clearly than sample-based proper scores do. The 800-update length was set after the 200-update study, so we treat those results as exploratory. In PushT, the conclusions hold on the roots used to fit Base. On 16 poorly fitted roots, Point stays above the truth's own risk and improves several probability metrics (Appendix C). Finally, excess gain works in one direction only, and a value of zero does not show that the probabilities are correct.

AI use statement

Large Language Models (LLMs) were used in this project as general-purpose assistant tools. Specifically, we used Claude Code with Claude Opus 5 to help sort and refactor code for readability and debugging during the research process, and used OpenAI ChatGPT-5.6 Sol to polish the writing for clarity and effective communication of our ideas. No part of the model design, experimental results, or scientific conclusions depended on LLMs.

Reproducibility statement

Appendix B provides the estimator proof and counterexample. Appendices C–G give experimental details and controls in the order of the main text, and Appendix H gives a supplementary analysis. Code, configurations, and per-seed results are provided as anonymized supplementary material.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

Michael Bereket and Jure Leskovec. Uncalibrated reasoning: GRPO induces overconfidence for stochastic outcomes. *arXiv preprint arXiv:2508.11800*, 2025.

Anthony Brohan et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023.

Jake Bruce et al. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Maxime Chevalier-Boisvert et al. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023.

Valentin De Bortoli, Alexandre Galashov, J Swaroop Guntupalli, Guangyao Zhou, Kevin Patrick Murphy, Arthur Gretton, and Arnaud Doucet. Distributional diffusion models with scoring rules. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in RL with verifiable rewards. *arXiv preprint arXiv:2601.15609*, 2026.

Amir-massoud Farahmand, André M. S. Barreto, and Daniel N. Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, 2017.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2023.

Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011. doi: 10.1198/jasa.2011.r10138.

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.

Tilmann Gneiting, Larissa I. Stanberry, Eric P. Grimit, Leonhard Held, and Nicholas A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *TEST*, 17(2):211–235, 2008. doi: 10.1007/s11749-008-0114-x.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. URL `https://papers.nips.cc/paper/2020/hash/3bb585ea00014b0e3ebe4c6dd165a358-Abstract.html`.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Anthony GX-Chen, Jatin Prakash, Jeff Guo, Rob Fergus, and Rajesh Ranganath. KL-regularized reinforcement learning for generative modelling is designed to mode collapse. In *International Conference on Learning Representations*, 2026.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Anthony Hu et al. GAIA-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Dhamma Kimpara, Rafael Frongillo, and Bo Waggoner. Proper losses for discrete generative models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17015–17040, 2023. URL https://proceedings.mlr.press/v202/kimpara23a.html.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *Deep Reinforcement Learning Meets Structured Prediction, ICLR 2019 Workshop*, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, 2018. URL https://proceedings.mlr.press/v80/kuleshov18a.html.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, 2020.

Nathan Lambert et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2411.15124.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *Conference on Language Modeling*, 2025.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Jonathan Sadeghi, Jenny Seidenschwarz, Jesse Allardice, Sirish Srinivasan, Benjamin Graham, and Jeffrey Hawke. CaliBench: Are the stochastic dynamics of video world models physically calibrated? *Transactions on Machine Learning Research*, 2026.

Michael Scheuerer and Thomas M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015. doi: 10.1175/MWR-D-14-00269.1.

Julian Schrittwieser et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

Novin Shahroudi, Mihkel Lepson, and Meelis Kull. Evaluation of trajectory distribution predictions with energy score. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Chenze Shao, Fandong Meng, Yijin Liu, and Jie Zhou. Language generation with strictly proper scoring rules. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 44474–44488, 2024a. URL https://proceedings.mlr.press/v235/shao24c.html.

Zhihong Shao et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/3d719fee332caa23d5038b8a90e81796-Paper-Conference.pdf.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, 1990.

Mark Towers et al. Gymnasium: A standard interface for reinforcement learning environments, 2024. URL https://arxiv.org/abs/2407.17032v1.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. RLVR-World: Training world models with reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. Daydreamer: World models for physical robot learning. In *Conference on Robot Learning*, 2022. URL https://proceedings.mlr.press/v205/wu23c/wu23c.pdf. CoRL 2022; proceedings published in 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00068.

Shirui Zhou, Shiteng Zheng, Junzhe Ding, Rui Jiang, and Junfang Tian. A strictly proper scoring-rule theory for calibrating stochastic car-following models, 2026. URL https://arxiv.org/abs/2609.04988.

Daniel M. Ziegler et al. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A NAMES, CONVENTIONS, AND CROSS-SYSTEM SUMMARY

This appendix collects what the main text uses throughout: arm names (Table 2), how uncertainty is reported, and a one-table summary of the main contrasts (Table 3). Appendices B–G then follow the order of the main text, and Appendix H holds an analysis that the main claims do not rely on.

Table 2: Names used throughout. Arms are named by their training objective; where the credit rule varies, it is added as a suffix.

Name	Meaning
Base	Starting checkpoint shared by all arms of a comparison
Point, MLE, ES	Post-training with a point reward (Section 2.1), maximum likelihood, or the energy score (Equation 4); PointRGB is Point on rendered RGB frames in MiniGrid
Point-track	Point with its KL weight tuned so that its final KL to Base matches ES's (PushT)
Point→X	The Point checkpoint trained further with objective X for an equal budget; Point→MLE and Point→ES are *proper-score continuations*
·-GRPO, ·-LOO	Credits standardized within each group as in GRPO, or unbiased leave-one-out credits (Equation 5); (u) marks LOO without step-size scaling
ES-mean	ES credits with the group mean subtracted but no standard-deviation scaling
Point-type arms	Point-GRPO, MAE-GRPO, Point-LOO, and Point-LOO (u) on RT-1
Root	A start state from which futures are predicted and, in decision tasks, an action is chosen; the unit of bootstrap resampling
Slip, slip+hold	The two FrozenLake dynamics: Gymnasium's slippery transitions, and the same with a 25% chance of repeating the previous action

Statistical conventions. PushT intervals resample roots within strata after averaging the three post-training seeds and condition on the trained models and generated evaluation sets. FrozenLake values are exact finite-horizon computations under the true kernel. RT-1 intervals are paired episode bootstraps over the 256 held-out episodes, conditional on the trained models; seeds are averaged before resampling, and the seed-to-seed consistency is reported separately. Training and test roots never overlap; MiniGrid Base seeds 4 and 5 share one test-root bank.

Cross-system summary. Table 3 collects the prediction and task contrasts. Comparisons use different evidence units across systems, so the effect sizes should be read within each row. A dash denotes a quantity not reported for that setting.

Table 3: Cross-system summary of the main contrasts. Task harm is Base minus Point success in percentage points; repair is the share recovered by MLE or energy-score continuation. Excess-gain shares are approximate point estimates relative to the truth's own risk; they are not confidence bounds. The slip task values refer to the training map; the slip-plus-hold values refer to unseen maps. RT-1 controlled values are relative to Base at 200 updates; L2 arms use five seeds and the MAE arm three.

System (role)	Point metric	Probability metric	Excess gain	Task harm	Recovery
RT-1, released (observational)	MAE -7.2%	joint ES $+26.5\%$, NLL $+83\%$	—	—	—
RT-1, controlled (5/3 seeds)	MAE -2.9 to -4.5%	joint ES $+2.7$ to $+3.1\%$	—	—	—
PushT (controlled)	risk -35%	$\mathrm{KL}(P \,\|\, Q)$ $2.70 \to 4.53$	$\approx 72\%$	not detected	—
FrozenLake slip	risk -1.8 to -2.1%	$\mathrm{KL}(P \,\|\, Q) \approx 0 \to 0.017$	$\approx 100\%$	2.1	87%
FrozenLake slip+hold	risk -4.1 to -4.4%	$\mathrm{KL}(P \,\|\, Q) \to 0.060$	$\approx 100\%$	1.7	93%
MiniGrid (visual)	RGB MSE -17.5%	NLL $1.98 \to 10.0$	—	10.3	91%
Two-channel (constructed)	risk -18%	$\mathrm{KL}(P \,\|\, Q) \approx 0 \to 0.44$	—	17.6	100%

B PROPOSITION 1: PROOF AND NUMERICAL COUNTEREXAMPLE

This appendix proves Proposition 1 (Section 5.1), shows why mean-only centering is also biased, states the LOO credits used in the experiments, and gives the toy computations behind Sections 5.2 and 5.3.

A group contains K independent model futures $X_1, \dots, X_K \sim Q$ and one independent observation $Y \sim P$. Sample i receives the energy-score credit r_i in Equation 5, which keeps only the terms of the group energy score that involve X_i. The unbiased estimator uses $r_i \nabla \log Q(X_i)$; GRPO uses $(r_i - \bar{r})/s$, with group mean $\bar{r}$ and standard deviation s (Shao et al., 2024b).

(a) Direct energy-score credits are unbiased. For the energy score,

$$\begin{aligned}\mathbb{E}\big[r_i \nabla \log Q(X_i)\big] &= -\mathbb{E}_{X,Y}\big[d(X,Y)\nabla \log Q(X)\big] + \mathbb{E}_{X,X'}\big[d(X,X')\nabla \log Q(X)\big] \\ &= -\nabla L_{\mathrm{ES}}(Q;P),\end{aligned}$$

by independence of the draws and symmetry of X, X', which cancels the factor $1/2$ when differentiating the pairwise term. With finite first moments (Section 2.1) and a differentiable interior parameterization at $Q = P$, propriety gives $\nabla L_{\mathrm{ES}}(P;P) = 0$ (Gneiting & Raftery, 2007; Gneiting et al., 2008). Dropping the pairwise term instead gives the negative point-risk gradient, which need not vanish at the truth.

(b) GRPO standardization is biased at the truth. GRPO replaces the reward r_i with $(r_i - \bar{r})/s$. The standardization depends on all K samples, so the standardized quantity that multiplies $\nabla \log Q(X_i)$ is no longer conditionally unbiased. Numerical counterexample: three outcomes with $d(x,y) = |x-y|$, $P = (0.2, 0.5, 0.3)$, $K = 8$, evaluating the expected standardized-reward update over all groups at $Q = P$ gives, per logit, $(-0.059, +0.135, -0.076)$: mass flows toward the middle outcome, whose neighbors inflate its pairwise term relative to the group mean. The expectation is computed by exact enumeration over groups (`toy/three_outcomes.py`).

(c) Two outcomes make ES-GRPO and Point-GRPO identical. Let the two outcomes be $\{0, 1\}$, let $n_{\neg Y}$ be the number of samples different from the observed outcome Y, and write the point reward as $p_i = -|x_i - Y| \in \{0, -1\}$. The ES reward is $r_i = p_i + \frac{1}{K-1}\sum_{j \neq i} |x_i - x_j|$, and direct computation gives

$$r_i = a\, p_i + b, \qquad a = \tfrac{2n_{\neg Y} - 1}{K-1} > 0, \quad b = \tfrac{n_{\neg Y}}{K-1},$$

whenever both outcomes appear in the group (if only one appears, both rewards are constant and both standardizations give zero). GRPO's standardization is invariant to positive affine transformations of a group's rewards, so the standardized ES and point rewards are equal in every group, and so are the updates.

Mean-only centering is not unbiased for coupled credits. Let $A = \mathbb{E}_{X,Y}[d(X,Y)\nabla \log Q(X)]$ and $B = \mathbb{E}_{X,X'}[d(X,X')\nabla \log Q(X)]$. Since the other credits also depend on X_i through their pairwise terms,

$$\mathbb{E}[(r_i - \bar{r})\nabla \log Q(X_i)] = -\frac{K-1}{K} A + \frac{K-2}{K} B.$$

At $Q = P$, $A = B$, leaving $-A/K$, generally nonzero. Thus the smaller empirical harm of mean-only centering does not establish an unbiased ES update.

The implemented LOO credits. In the -LOO arms of PushT, RT-1, and FrozenLake, sample i receives $r_i - b_i$, where b_i is the same credit computed from the other $K - 1$ samples ($K \geq 3$):

$$b_i = -\frac{1}{K-1}\sum_{j \neq i} d(X_j, Y) + \frac{1}{(K-1)(K-2)} \sum_{\substack{j \neq l \\ j,l \neq i}} d(X_j, X_l).$$

Since b_i does not involve X_i, $\mathbb{E}[b_i \nabla \log Q(X_i)] = 0$ and the update in (a) is unchanged in expectation. Point-LOO uses the same construction without the pairwise terms. On RT-1, the resulting credits are further multiplied by a fixed constant (Table 6), which rescales the expected update without changing its direction or its zero at $Q = P$.

Toy computations for Sections 5.2 and 5.3. The toy discretizes a bimodal distribution on $[0, 1]$, and expected updates are computed by enumeration or Monte Carlo over groups. At the truth, the norm of the expected ES-GRPO update relative to Point-GRPO's is 1.000, 0.986, 0.652, 0.496, 0.498, 0.502, and 0.503 for $n = 2, 3, 5, 10, 20, 50, 200$ outcomes (`toy/support_check.py`). The ratio stays near one half rather than vanishing, because the standardized ES credit keeps sample-dependent variation in the pairwise term. Along a geometric path to the truth, with path parameter $1.0, 0.6, 0.3, 0.15, 0.05$, the cosine between the ES-GRPO and ES-LOO updates is 0.974, 0.949, 0.861, 0.765, and 0.685, against 0.528 to 0.195 for Point-GRPO; at the truth, the ES-GRPO update is nonzero while the ES-LOO update vanishes (`toy/distance_check.py`).

C PushT: details and full results

This appendix supports Section 3.1 and the PushT part of Section 4.3. It gives the full endpoint table, the setup, the per-seed matched-KL results on all three Bases, and the boundaries: decision diagnostics and roots never used for fitting.

Table 4: PushT at the 1,000-update endpoint, averaged over three optimization seeds. Distance improvement is relative to Base. ED/2 is half the energy distance ($\times 10^{-3}$). KL to Base measures how far each arm moved from the starting checkpoint. Point+MLE adds a likelihood term on true futures to the Point objective.

Model	Dist. improv.	Event RMSE	$\mathrm{KL}(P \,\|\, Q)$	ED/2	KL to Base
Base	—	0.031	2.70	0.61	0
MLE	3%	0.016	2.50	0.18	0.24
Point	**35%**	0.083	4.53	5.32	0.81
ES	5%	0.018	2.68	0.22	0.10
Point-track	23%	0.054	2.82	2.22	0.10
Point+MLE	34%	0.082	4.00	5.09	0.73

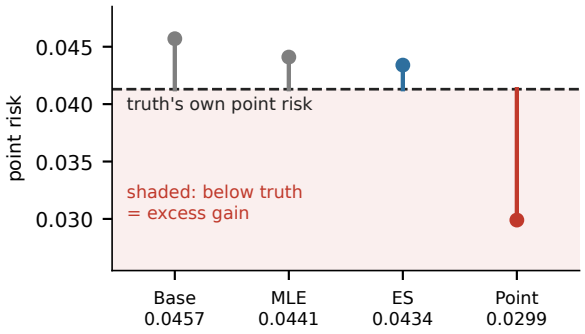


Figure A1: PushT point risk on the 64 fitted roots. The dashed line marks the truth's own risk. About 72% of Point's gain lies below it (shaded), while MLE and ES stay above it.

C.1 Conditional futures, model, and fitting

Each condition fixes the recorded physical state, previous executed command, and 12 proposed commands. At each command, independent sticky randomness with probability 0.25 replaces the proposal with the previous executed command. All 4,096 binary masks are replayed under the simulator, with mask probability $0.25^s 0.75^{12-s}$ for s sticky steps. Identical continuous trajectories and, separately, identical token sequences are merged by summing probabilities; masks and duplicate futures are not independent experimental units. The fitted panel contains 32 reset and 32 decision roots, each with four fixed candidates; 16 further roots, not used to fit Base, form a separate panel. All 80 root reset seeds are distinct. Uncertainty resamples roots, retaining their four correlated candidates.

The common StateAR model has 5,094,400 parameters: six pre-normalization Transformer layers, width 256, eight heads, and feed-forward width 1,024. It autoregressively predicts 60 tokens (agent x/y, block x/y, block angle at each of 12 commands) from a 38-value condition vector. Each field uses the same fixed 512-category codec relative to the initial pose; angle displacement is wrapped, and out-of-support values raise an error rather than clipping. No post-training arm refits the codec. The original base B was warm-started and fitted for 80,000 updates in total; bases 310 and 311 start from random weights and are fitted for 120,000 updates. All arms within each base start from the same checkpoint; the three bases are reported separately because their fitting histories differ.

C.2 Rewards, estimation, optimization, and KL matching

For a state $s = (x_a, y_a, x_b, y_b, \theta_b)$ the fixed geometry is $\phi(s) = (x_a/512,\, y_a/512,\, x_b/512,\, y_b/512,\, \sin\theta_b,\, \cos\theta_b)$, and the distance d is the Euclidean distance between flattened 12-step feature trajectories divided by $\sqrt{72}$. Point minimizes $\mathbb{E}[d(X, Y)]$; ES minimizes Equation 4. Each update visits all 256 conditions in microbatches of 32, using 32 independent model futures and 16 oracle futures per condition; MLE uses the same oracle draws for token likelihood training. Point and ES use leave-one-out score-function credits, the ES credit including the model–model distance term; neither uses group standardization. Rewards divide distances by a fixed scale of 0.0155 (the exact value is in the code). All post-training arms start with fresh AdamW state, learning rate 10^{-6}, zero weight decay, and gradient-norm clipping at 1; Point and ES use sequence-KL coefficient 0.1. We report results at 1,000 updates and also evaluate at 100 and 300; optimization seeds 20, 21, and 22 repeat post-training from each fixed base; the base itself is fitted once.

The Point-track controller changes only its KL coefficient using the paired ES run's measured sequence KL: every 10 updates it applies a log-coefficient gain of 0.2, clips relative feedback to $[-0.5, 0.5]$, constrains the coefficient to $[10^{-4}, 10^{4}]$, and floors the target KL at 0.005. A separate 256-draw evaluation measures the final mean $D_{\mathrm{KL}}(Q\|\mathrm{base})$, with a target gap of at most 20% to ES in each seed. The match holds for the panel mean at the final checkpoint, not for every root or every step: the original-base gaps are 1.2%, 4.9%, and 8.8%, and the six fresh-base gaps are at most 5.7%.

$\mathrm{KL}(P \,\|\, Q) = D_{\mathrm{KL}}(P^{\tau}\|Q)$ uses the merged finite oracle token support and exact autoregressive sequence probabilities; it is not continuous-state density KL. Point risk and energy distance use 256 model and 256 oracle draws per condition with off-diagonal self-distance terms. Four physical-event probabilities concern block translation above 10 or 30 pixels and absolute rotation above 5 or 15 degrees; event MSE subtracts the model-draw sampling-variance term before averaging over events and conditions, and event RMSE is the square root of that aggregate. For base B, intervals use 5,000 paired root-bootstrap draws that keep the 32 reset and 32 decision roots in fixed proportion; seeds are averaged before root resampling, so intervals condition on the fitted base, the optimization runs, and the generated evaluation banks. They do not cover new training seeds or model families.

Truth's own risk. The PushT value of the truth's own risk in Section 3.1 uses the existing 256-draw model and oracle evaluation banks. With the same oracle draws, it equals twice the difference between ES and half the energy distance. The reported excess-gain fraction is computed from panel-averaged point-risk estimates, rather than an exact sum over all oracle trajectories; a ratio of panel averages need not equal the average of per-condition fractions.

C.3 Matched-control results by seed

Table 5 reports each seed at 1,000 updates on the fitted panel. Each base's ES and Point-track rows are paired by seed; Point is included for reference. The original base B has $\mathrm{KL}(P \,\|\, Q)$ 2.699; the independently fitted bases 310 and 311 start at 0.138 and 0.137.

Table 5: Matched-control results by seed. Lower event RMSE and $\mathrm{KL}(P \,\|\, Q)$ are better; mean KL to Base records how far each arm moved from the starting checkpoint.

Base	Seed	Arm	Event RMSE	$\mathrm{KL}(P \,\|\, Q)$	Mean $\mathrm{KL}(Q\|\text{base})$
B	20	Point	0.08334	4.52335	0.80834
B	20	ES	0.01739	2.68828	0.09743
B	20	Point-track	0.05400	2.81393	0.09626
B	21	Point	0.08346	4.54747	0.81044
B	21	ES	0.01758	2.67219	0.09989
B	21	Point-track	0.05369	2.80875	0.09504
B	22	Point	0.08357	4.51745	0.80817
B	22	ES	0.01817	2.69035	0.09596
B	22	Point-track	0.05472	2.82270	0.10441
310	20	Point	0.07901	0.99692	0.40945
310	20	ES	0.01544	0.13631	0.01127
310	20	Point-track	0.02797	0.14769	0.01063
310	21	Point	0.07907	0.99230	0.40921
310	21	ES	0.01553	0.13464	0.01126
310	21	Point-track	0.02851	0.14835	0.01120
310	22	Point	0.07895	0.99019	0.40907
310	22	ES	0.01510	0.13578	0.01156
310	22	Point-track	0.02837	0.14799	0.01142
311	20	Point	0.07954	1.01478	0.41553
311	20	ES	0.01697	0.13480	0.01056
311	20	Point-track	0.02969	0.14779	0.01090
311	21	Point	0.07943	1.00642	0.41293
311	21	ES	0.01673	0.13458	0.01104
311	21	Point-track	0.02965	0.14753	0.01080
311	22	Point	0.07953	1.01689	0.41668
311	22	ES	0.01658	0.13668	0.01235
311	22	Point-track	0.03084	0.14934	0.01194

Averaged over the three seeds, Point-track minus ES event MSE is $+5.6 \times 10^{-4}$ (95% paired-root interval $[3.6, 7.8] \times 10^{-4}$) on base 310 and $+6.2 \times 10^{-4}$ ($[4.2, 8.4] \times 10^{-4}$) on base 311, and the $\mathrm{KL}(P \,\|\, Q)$ differences are $+0.012$ and $+0.013$. These replicate the contrast on the same states with independently initialized bases; they do not test new states. In the decision analysis of Figure A3, each root offers four fixed candidate actions; the planner picks the one with the highest target coverage estimated from model samples, and coverage regret is the true expected coverage of the best candidate minus that of the chosen one. Coverage-value error is the absolute error of the estimated coverage values.

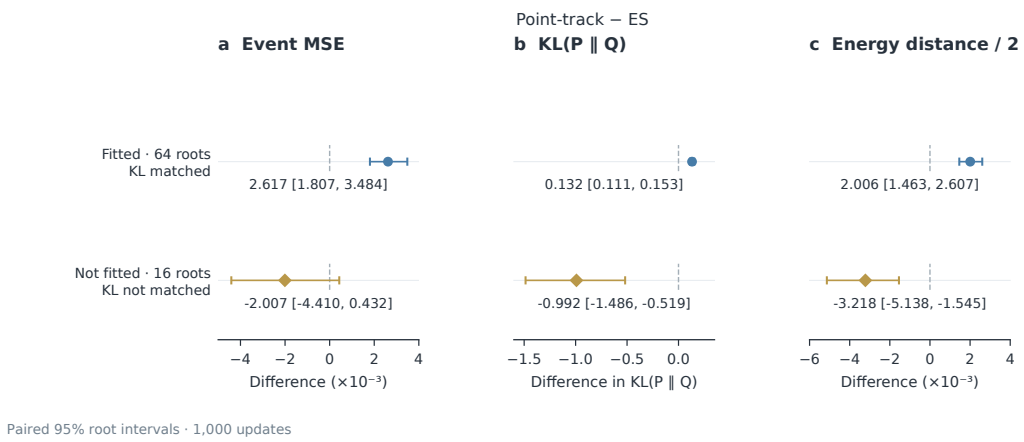


Figure A2: Applicability boundary of the matched PushT contrast. Point-track minus ES at 1,000 updates for event MSE, $\mathrm{KL}(P \,\|\, Q)$, and half the energy distance. Markers average three seeds; bars are paired 95% root-resampling intervals. On the 64 fitted roots the endpoint-KL match holds. On 16 roots not used to fit the base, the base fit is poor and the KL match fails. There Base is far above the truth's point risk (0.153 against 0.035), Point stays above it (0.139), and Point improves several probability metrics, consistent with the excess-gain reading of Section 2.1.

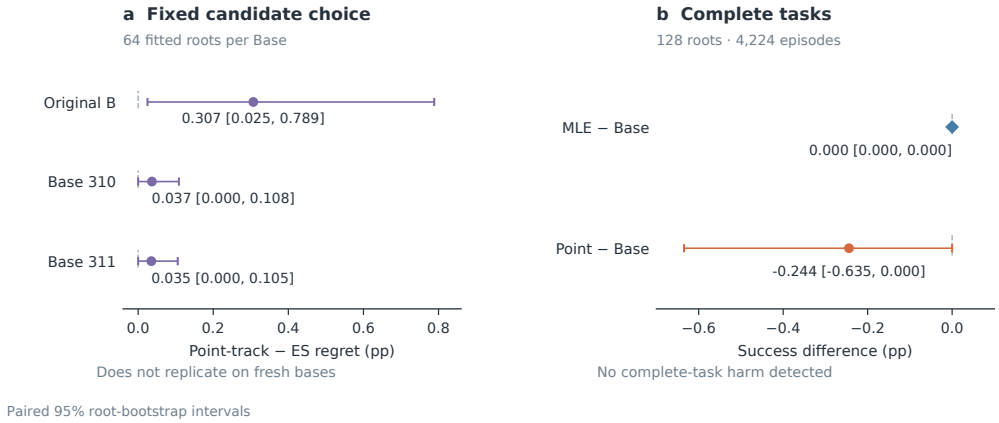


Figure A3: Localized PushT decisions and the complete-task null. (a) Point-track minus ES coverage regret on 64 fitted roots for the original base and independently fitted bases 310 and 311. (b) Success differences relative to Base on 128 roots and 4,224 complete episodes: complete-task harm is not established. Bars are paired 95% root-bootstrap intervals. The two panels use different decision metrics.

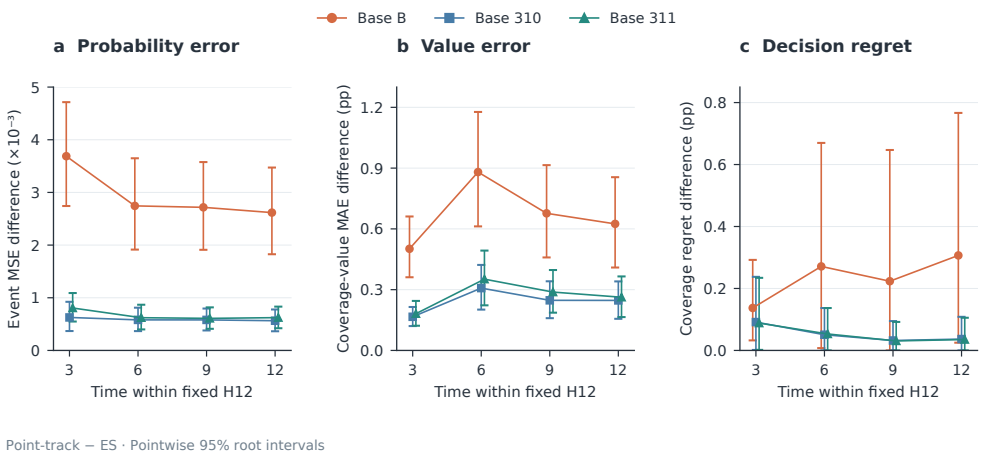


Figure A4: Diagnosis across time steps within the same 12-step trajectories. Point-track minus ES probability error, coverage-value error, and decision regret at steps 3, 6, 9, and 12 for the original base B and fresh bases 310 and 311. Bars are pointwise 95% root intervals. The comparison spans steps within one horizon; separately trained horizons are not compared. Choices become more stable over time although probability error persists.

C.4 Other PushT decision interfaces

The PushT model predicts 5D poses, which limits how its probabilities can enter a complete-task decision. Three alternative interfaces do not change this. Value heads fitted on true 12-step futures predict task value poorly, adding the true hidden velocities to predicted poses does not give usable rollouts, and with a public policy or cached values, choosing by the predicted distribution is no better than choosing the physically valid medoid. Across time steps, choices also become more stable while probability error persists (Figure A4). Together these account for the small PushT task effect in Section 4.3.

C.5 Roots never used for fitting

On the 16 PushT roots never used for base fitting, the base's $\mathrm{KL}(P \,\|\, Q)$ is about 318 and the Point-track/ES KL match fails in every seed (gaps of 25–38%), so the fitted-panel conclusions do not carry over to them. This is consistent with the poor initial fit (Figure A2).

D RT-1 controlled experiment

This appendix supports Sections 3.2 and 5.3. It gives the setup and arms, the full 200-update table, the matched-divergence analysis, the 800-update runs, and the evaluation of the released checkpoint.

D.1 Setup

Base: official `thuml/rt1-world-model-multi-step-base`; data: 256 training windows and 32 development windows. Training: batch 4 conditions $\times$ 8 futures, AdamW learning rate 5×10^{-6}, KL coefficient 0.001, clip 1. We evaluate at 200 updates on 256 held-out episodes with $K = 16$. The main contrasts are ES-GRPO − ES-LOO and Point-GRPO − ES-LOO on joint ES and NLL; we also report ES-GRPO − Point-GRPO, the scaled-versus-unscaled pairs, every arm against Base, and sample metrics.

Table 6: The seven arms of the RT-1 controlled experiment. All share base, data, schedule, optimizer, and KL; only the reward and the advantage computation differ. LOO: unbiased leave-one-out credits (Equation 5 for ES); the main text uses the names in the first column. *Seeds 20–24 from 400 updates.

Arm	Code name	Reward	Credit	Seeds
Point-GRPO	`point_grpo`	full-video L2	GRPO (group mean/std)	20–24
ES-GRPO	`es_grpo`	energy score	GRPO	20–24
Point-LOO	`point_l2_scaled`	L2	LOO, $\times 72$	20–24
ES-LOO	`energy_l2_scaled`	energy score	LOO, $\times 124$	20–24
Point-LOO (unscaled)	`point_l2`	L2	LOO, unscaled	20–24
ES-LOO (unscaled)	`energy_l2`	energy score	LOO, unscaled	20–24
MAE-GRPO	`mae_grpo`	pixel MAE	GRPO	20–22*

D.2 Arms

D.3 Batched sampling and evaluation reuse

Training rollouts place the 4×8 samples of each step in one batch, which changes only the random-number stream; all sampling arms use this code. All arms, Base, and the released checkpoint are scored by the same evaluation code; MLE, which does not sample during training, is unaffected by the batching. The scaling constants 72 (Point) and 124 (ES) are the credit standard deviations measured in short test runs.

D.4 MLE on RT-1 behaves like the point-type arms

The MLE arm worsens joint ES by 2.6% while improving MAE by 2.7% (Table 7), unlike MLE in every enumerable system. We do not use MLE as a probability-correct reference on RT-1. This is plausibly because MLE fits the single recorded future per condition (256 training windows) more directly than the sampling-based objectives do.

Table 7: RT-1 controlled experiment at 200 updates, 256 held-out episodes, $K = 16$. All columns: change relative to Base (%). Values are seed averages; intervals for the contrasts are in the text. RLVR is the released checkpoint under the same evaluation (observational).

Arm	joint ES	NLL	MAE	LPIPS
Point-GRPO	$+3.0$	$+1.0$	-3.2	-1.7
ES-GRPO	$+0.9$	$+0.8$	-0.9	-0.2
Point-LOO	$+2.7$	$+0.8$	-3.1	-1.5
ES-LOO	$+0.9$	$+0.8$	$+0.3$	$+0.5$
Point-LOO (unscaled)	$+2.8$	$+0.9$	-2.9	-1.5
ES-LOO (unscaled)	$+0.7$	$+0.9$	$+0.3$	$+0.4$
MAE-GRPO	$+3.1$	$+1.2$	-4.5	-1.9
MLE	$+2.6$	$+0.6$	-2.7	-1.8
RLVR (released)	$+26.5$	$+83.0$	-7.2	-3.7

D.5 Matched divergence and unoptimized scores

Table 8 supports the matched-divergence paragraph of Section 3.2. The training KL is the on-policy Monte Carlo estimate logged at every update: for each training condition, the mean over the 8 sampled futures of $\log Q(X) - \log \mathrm{Base}(X)$ over the whole token sequence. We average it over each seed's last 20 updates before the stated step and then over seeds; it is measured on training conditions, not on the held-out split. The Gaussian kernel score uses $k(x, y) = \exp[-d(x, y)^2/(2(\alpha\sigma)^2)]$ on the full-video distance d with fixed scale σ, averaged over $\alpha \in \{0.5, 1, 2\}$; the temporal variogram score compares the predicted and observed absolute frame-to-frame changes with an unbiased estimator. No arm is trained on either score. Contrasts use the same 256 held-out episodes, $K = 16$, and episode-paired bootstrap as Section 3.2. The table is computed from the saved evaluations by `figures/scripts/rt1_divergence_check.py`.

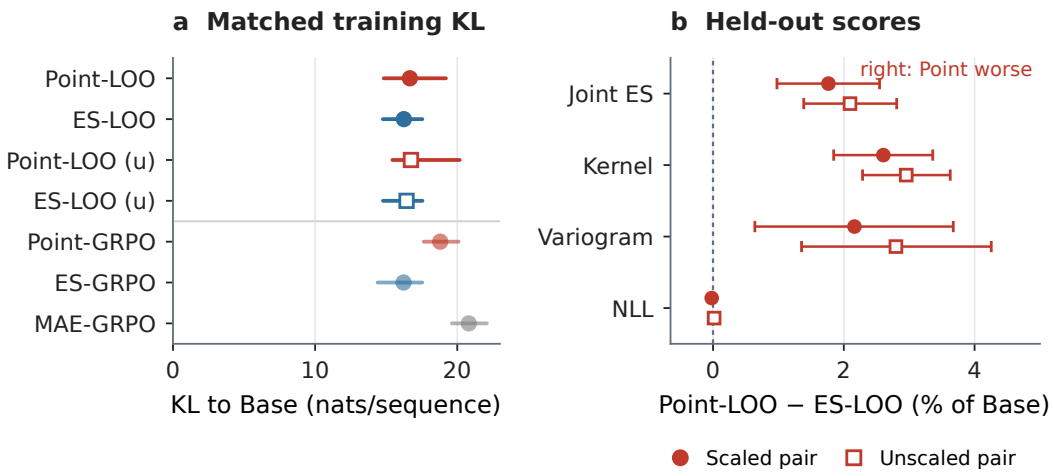


Figure A5: RT-1 at matched divergence. (a) Training KL to Base over the last 20 updates before the 200-update endpoint (dots are seed means; lines span seeds). Adjacent Point-LOO and ES-LOO pairs are matched; (u) marks the unscaled variants. (b) Point-LOO minus ES-LOO on held-out scores (joint ES is optimized by ES-LOO; the kernel and variogram scores by neither arm), relative to Base, with episode-paired 95% bootstrap intervals: point rewards are worse on two proper scores that neither arm optimizes, while NLL does not separate them (Appendix Table 8).

Table 8: RT-1 divergence and unoptimized proper scores. Top: training KL to Base (nats per sequence; mean over seeds, range in brackets; five seeds unless noted). Bottom: 200-update contrasts relative to Base (%), positive means the first arm is worse; 95% episode-paired bootstrap intervals; last column counts seeds with the same sign as the mean for joint ES / kernel / variogram. Unscaled pair: Point-LOO − ES-LOO without step-size scaling.

Arm	KL at 200	KL at 800
Point-GRPO	18.8 [17.6, 20.1]	83.0 [72.2, 93.7]
MAE-GRPO	20.8 [19.6, 22.1]	83.1 [75.3, 89.0]
Point-LOO	16.7 [14.8, 19.2]	—
Point-LOO (unscaled)	16.7 [15.4, 20.2]	—
ES-GRPO	16.2 [14.4, 17.5]	64.9 [53.8, 73.2]
ES-LOO	16.2 [14.8, 17.5]	57.4 [53.2, 61.7]
ES-LOO (unscaled)	16.4 [14.8, 17.6]	—

Contrast	joint ES	kernel score	variogram	NLL	seeds
Point-LOO − ES-LOO	+1.8 [1.0, 2.5]	+2.6 [1.8, 3.4]	+2.2 [0.6, 3.7]	−0.02 [−0.06, 0.02]	5/5/4
Unscaled pair	+2.1 [1.4, 2.8]	+3.0 [2.3, 3.6]	+2.8 [1.4, 4.3]	+0.02 [−0.02, 0.06]	5/5/5
Point-GRPO − ES-GRPO	+2.1 [1.4, 2.7]	+2.9 [2.3, 3.6]	+1.5 [0.05, 2.9]	+0.25 [0.21, 0.29]	5/5/4

D.6 Training to 800 updates

All seven arms' seed-20 runs continue to 800 updates. For Point-GRPO, ES-GRPO, ES-LOO, and MAE-GRPO, seeds 21–24 continue from their 200-update state with the same data order as an uninterrupted run, and MAE-GRPO seeds 23–24 train from scratch. Table 9 gives the resulting contrasts (`figures/scripts/rt1_dose_check.py`). Point-type rewards pull away from ES-LOO steadily, while ES-GRPO separates only late. Relative to Base, the MAE gain at 800 updates is 6.8% for Point-GRPO and 7.5% for MAE-GRPO.

Table 9: RT-1 contrasts against ES-LOO over training (% of Base's score; positive means the first arm is worse on joint ES and NLL, and negative MAE means lower pixel error). Five seeds; 95% episode-paired bootstrap intervals at 800 updates. At 800 updates every entry has the sign of its mean in all five seeds, except ES-GRPO's NLL (four of five).

	joint ES			at 800 updates	
Contrast	400	600	800	NLL	MAE
Point-GRPO − ES-LOO	$+4.0$	$+6.2$	$+7.2$ [6.1, 8.4]	$+2.1$	-7.9
MAE-GRPO − ES-LOO	$+4.3$	$+6.0$	$+7.4$ [6.1, 8.8]	$+2.2$	-8.6
ES-GRPO − ES-LOO	$+0.1$	$+0.6$	$+1.2$ [0.6, 1.8]	$+0.5$	-2.0

D.7 Released checkpoint

The released-checkpoint numbers in Section 3.2 use the controlled experiment's evaluation settings (256 episodes, $K = 16$, top-k 0, temperature 1) applied to the released RLVR-World checkpoint and its base, so that they are comparable with the controlled arms. An evaluation with the official decoding settings (top-k 100) gave MAE -6%, LPIPS -3%, joint ES $+26\%$, and NLL nearly doubled—the same pattern. Teacher-forced NLL evaluates the original model probabilities.

E FrozenLake: training, seeds, and the GRPO study

This appendix supports Sections 4.1 and 5.2.

E.1 Training and independent training seeds

Both task chains use Gymnasium's 8×8 slippery FrozenLake with exact 200-step finite-horizon planning and success evaluation. Slip has three reachable modes; its Base estimates a shared symmetric slip parameter from 1,024 environment transitions per nonterminal state-action, with 64 independent validation transitions, and initializes a trainable $68 \to 128 \to 64 \to 3$ network from that estimate. Slip+hold adds the 25% action hold; its Base estimates global slip and hold rates from 256 environment transitions per nonabsorbing augmented state-action, with 32 validation transitions, and initializes a $324 \to 128 \to 64 \to 6$ network. Both Bases are close to the true kernel in KL and in task value.

Under each dynamics, Point, MLE, and ES start from the same Base with the same added environment transitions, minibatch stream, capacity, optimizer, and 200-step planner. Point is the expected Euclidean distance between a generated and an observed next-state feature; MLE is negative observed-outcome log likelihood; ES subtracts half the expected distance between two generated futures. Reachable modes are summed exactly in the small output space, reducing estimator variance without supplying true probabilities to training. All arms include $0.5\, D_{\mathrm{KL}}(Q \| \mathrm{Base})$ and run 1,200 Adam updates with batch size 512 and learning rate 0.001. Point→MLE and Point→Point continue the same final Point checkpoint for another 1,200 updates on a shared fresh environment-sample budget. The true kernel is used only to check Base and for the final exact evaluation, never in training. Under each dynamics, models are trained on one map, and five fresh training seeds repeat each full chain on that map (Table 10). The two 16-map panels answer different questions. Under slip, the whole chain, including Base fitting, is rerun on each of 16 new maps, so the panel tests replication across maps. Under slip+hold, the trained checkpoints are evaluated on 16 unseen maps without retraining, so the panel tests transfer. Replacing Point's predicted means with the true means brings success close to Base under both dynamics; on 16 further maps the remaining loss is below 0.2 points. On the slip 16-map panel, Point costs at least one point of success on 10 of the 16 maps, and 8 of these also meet every other step of the chain: lower point risk, a changed policy, distorted probabilities, and recovery by Point→MLE.

Table 10: Training-map complete-task success differences (percentage points) for the five fresh training seeds. Harm is Base−Point; recovery is Point→MLE minus Point.

Dynamics	Seed	Harm	Point→MLE recovery
Slip	0	6.31	5.94
Slip	1	2.22	1.84
Slip	2	4.45	3.48
Slip	3	1.90	1.66
Slip	4	2.53	2.26
Slip+hold	0	4.93	4.60
Slip+hold	1	8.06	7.43
Slip+hold	2	5.68	5.46
Slip+hold	3	6.18	5.82
Slip+hold	4	3.01	2.44

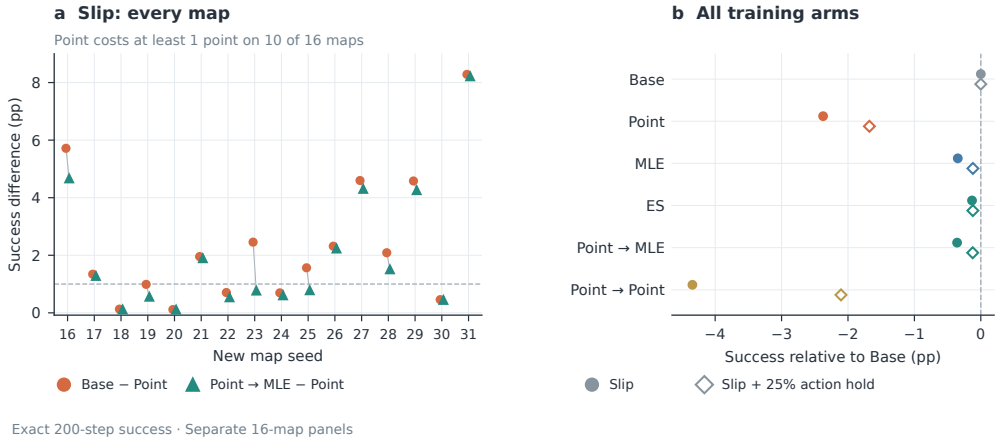


Figure A6: FrozenLake harm and repair, companion panels to Figure 3. (a) Exact success loss (Base−Point) and recovery (Point→MLE minus Point) on all 16 new slip maps, each with its own Base; the dashed line marks one percentage point, which 10 of 16 maps reach. (b) Mean exact success relative to Base for all six arms under slip and slip+hold, evaluated on separate 16-map panels. Values are exact 200-step success probabilities, so no per-arm intervals are shown. Both dynamics belong to one task family.

E.2 The GRPO propriety study: full results

This study runs on the fresh seeds' training maps rather than the unseen-map panels of Section 4.1, and it reuses the Bases, post-training samples, minibatch streams, optimizer, and KL of the mixture sweeps, so only the reward-gradient estimate changes. Each context draws K model futures and observes one environment outcome. The arms are exact Point (the full-distribution expectation), ES-LOO, ES-mean, and ES-GRPO, each under both dynamics with five seeds.

ES-GRPO harms every one of the ten seeds, while mean-only centering is nearly harmless. The harm is stable across $K = 4$, 8, and 16 (1.8 to 2.5 points), and it is excess gain in the sense of Section 2.1: ES-GRPO pushes point risk below Base's, which matches the truth's own risk within 0.1%.

KL-penalty control. Weakening the KL coefficient of ES-LOO by $10\times$ and $100\times$ ($\beta = 0.05, 0.005$) puts exact point risk between 0.08% below and 0.17% above Base's across the 20 runs (5 fall below it) and costs 0.2–0.6 points of success on average (worst seed 2.0). These KL controls do not reproduce ES-GRPO's shift of 1.2–1.4% below Base.

Table 11: FrozenLake GRPO study, $K = 8$: complete-task harm, Base minus arm success (points; five seeds, range in brackets; positive is harm), and exact point risk relative to Base (%).

Arm	slip harm	slip+hold harm	point risk (slip / slip+hold)
Point (exact)	$3.4\ [1.9, 6.3]$	$5.6\ [3.1, 7.9]$	-1.9% / -4.3%
ES-LOO	0.1	0.1	0 / 0
ES-mean	0.1	0.1	-0.1% / -0.3%
ES-GRPO	$2.3\ [1.6, 3.6]$	$2.5\ [1.4, 4.2]$	-1.4% / -1.2%

Repair from a damaged checkpoint. Continuing the Point checkpoints of the main chains with the same budgets (harm in points, slip/slip+hold): ES-LOO reduces harm from $3.4/5.6$ to $2.1/2.4$; ES-GRPO deepens it to $6.0/8.4$; Point-GRPO deepens it to $27.7/30.6$.

F MiniGrid: setup and backbone choice

This appendix supports Section 4.2. It gives the full seed-2 results, the setup and decision interface, the two further Base seeds, and why the backbone is fixed.

Table 12: MiniGrid, Base seed 2, 512 unseen roots and 65,536 environment episodes. Continuation arms start from the PointRGB checkpoint with equal extra budgets. Seeds 4 and 5 are in Figure 3b.

Arm	Success	Brier	Token NLL	RGB MSE ($\times 10^{-3}$)
Base	29.3%	0.122	1.98	2.92
PointRGB	18.9%	0.153	10.04	**2.41**
MLE from Base	29.6%	0.123	1.96	2.91
PointRGB→energy score	28.3%	0.124	2.10	2.91
PointRGB→PointRGB	21.8%	0.158	12.26	**2.42**

The model is a CNN-conditioned autoregressive predictor of four obstacle positions, followed by a fixed renderer that turns each predicted map into an exact 56×56×3 RGB frame. Its input is the official partial RGB observation. The main Base is training seed 2. Post-training keeps the CNN backbone fixed and updates only the head. Training roots are 10000–11599, validation roots 11600–11999, and test roots 60000–60511. PointRGB and MLE both train for two epochs (batch 128, AdamW learning rate 3×10^{-4}, weight decay 10^{-4}, gradient clipping at 1) and use the final checkpoint; continuation from PointRGB uses the same extra two-epoch budget for energy score and PointRGB, with identical example order within each paired comparison.

PointRGB uses four straight-through Gumbel map draws at temperature 1. For each valid future step 1–20, the sampled next map is rendered at the teacher-forced true next agent pose and scored by normalized squared error over the entire RGB frame; the pose defines the supervised loss and is not supplied as an unknown future input at deployment. MLE scores the future map-token sequence. The repair objective is the normalized full-frame Euclidean energy score, estimated with four independent generated RGB frames and all six distinct generated pairs; renderer weight identities were checked against explicit RGB arrays on 4,096 sampled frame pairs. The population energy score is proper in the evaluated RGB space (Gneiting & Raftery, 2007), but straight-through gradients and finite training do not guarantee reaching its population optimum, and these losses are per-frame conditional objectives rather than a likelihood or energy score for the joint future video.

The decision interface offers four fixed 20-action routes; after each route the environment episode continues with in-place turns until goal, collision, or the 256-step limit, with no extra route chosen. Every arm draws 64 model futures per route, applies the same deterministic projection and task scorer, and selects one route. PointRGB changes the chosen route on 311 of the 512 seed-2 roots. On each of 512 new roots, four routes each have 32 independently seeded full-task environment continuations, yielding 65,536 unique official episodes in a shared outcome bank; collision, success, and timeout remain in the denominator, and the root is the independent resampling unit with 20,000 paired draws.

Two further Base seeds (4 and 5) each run the same five arms with the same fixed backbone, training roots, and head-only recipe, and both share one new root bank (roots 70000–70511, again with four routes and 32 environment continuations per root and route). Intervals use 20,000 paired bootstrap draws over the 512 roots, and the three seeds are reported separately. Seed 4: Base 28.0% and PointRGB 18.4% success, harm 9.6 points $[8.2, 10.9]$, and ES continuation beats equal-budget Point continuation by 6.8 points $[5.6, 8.0]$. Seed 5: Base 28.4% and PointRGB 20.1%, harm 8.3 points $[7.2, 9.6]$, and ES continuation ahead by 5.1 points $[3.9, 6.3]$.

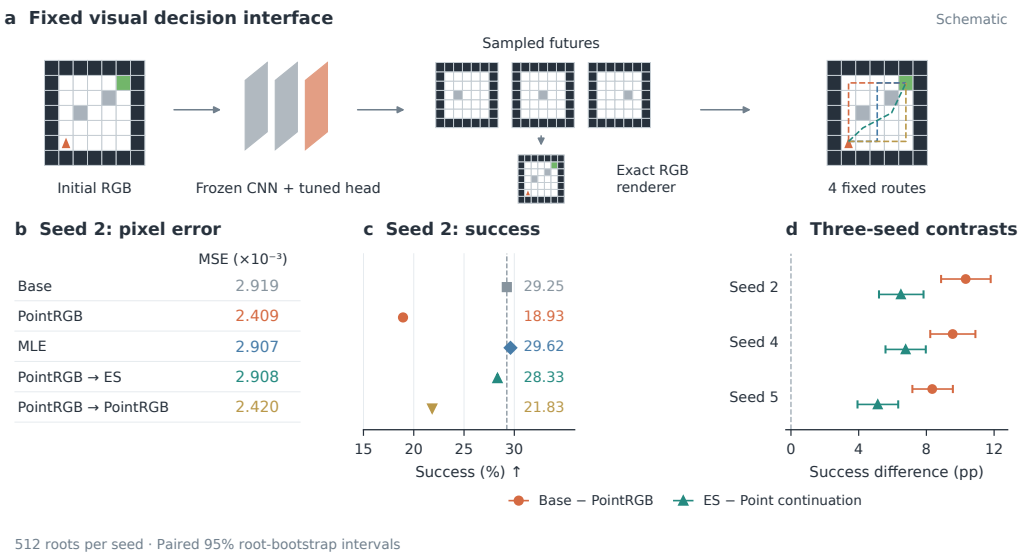


Figure A7: MiniGrid companion panels to Figure 3b. (a) Schematic of the fixed visual prediction and four-route decision interface. (b–c) All five seed-2 arms: PointRGB lowers RGB error while reducing success; energy-score (ES) continuation recovers most of the success gap while surrendering the pixel-error gain. Dashed line: Base success. (d) Base−PointRGB harm and ES-continuation advantage over equal-budget Point continuation for seeds 2, 4, and 5. Bars are paired 95% root-bootstrap intervals (20,000 draws; 512 roots per seed). Seeds 4 and 5 share one root bank.

Why the backbone is fixed. Two other variants of this setup do not separate the objectives clearly. With Base seed 0, PointRGB causes harm, but MLE from the same Base does not clearly do better, and temperature calibration improves NLL without restoring success. With Base seed 1 and a trainable backbone, neither MLE nor energy-score continuation separates the objectives clearly. Swapping modules between models traces this to the backbone, so all reported runs keep it fixed and train only the prediction head.

G When distortion matters: constructed and standard tasks

This appendix supports Section 4.3. The constructed task shows that distribution shape alone can change a decision; the standard tasks show that stochastic dynamics alone do not.

Constructed two-channel task. *Environment.* A particle launched from the origin enters a left or right channel chosen by the planner and then follows one of three 12-step trajectories, fixed at launch, whose lateral exit offset is ma with $m \in \{-1, 0, +1\}$. Each root draws $c_L \sim U[0.18, 0.24]$, $c_R \sim U[0.58, 0.64]$, and $a \sim U[0.9, 1.1]$; in channel k, the center outcome $m = 0$ has probability c_k and each side has $(1 - c_k)/2$. The left channel succeeds at its side exits ($|m| = 1$) and the right channel at its center exit ($m = 0$), so the true success values are $1 - c_L$ and c_R, and the left channel is optimal at every root.

Model and planner. The observation is a 32×32 image with one texture patch per channel, whose pixel density is c_k, plus a bar encoding a. A linear layer maps the mean intensity of each patch to a

center probability q_k, and each side outcome receives $(1 - q_k)/2$; every model's mean exit offset is therefore zero, equal to the truth. The planner picks the channel with the higher predicted success, $1 - q_L$ or q_R.

Why Point harms. With $d(x, y) = |x-y|$, the point risk in channel k is $a(1-c_k q_k)$, which decreases in q_k, so Point moves mass to the center outcome in both channels. Once $q_L > 1 - q_R$, the planner switches to the right channel. Point switches at all 64 test roots, so its loss equals the mean gap between the two true values, $\mathbb{E}[(1 - c_L) - c_R] = 17.6$ points.

Training and evaluation. Training roots 20000–27999 (two sampled outcomes per root and channel) and validation roots 28000–28999 are disjoint from the 64 test roots 1000–1063. Base is trained by likelihood for 1,500 AdamW updates (batch 256, learning rate 10^{-3}) and reaches $\mathrm{KL}(P \,\|\, Q) \leq 0.01$ and the optimal channel on at least 95% of validation roots. Point, MLE, ES, and a direct success-value model each take 600 updates from Base, and Point→MLE and Point→Point take 1,200 from Point (learning rate 2×10^{-3}). Point and ES use exact expectations over the three outcomes; MLE and the direct model use the sampled outcome or its success. Eight training streams resample outcomes and minibatch order. Success is computed exactly under the true probabilities. The direct success-value model matches Base, so the task does not show that a full distribution is required for good decisions; it shows that distorting the shape alone can cost a planner.

Table 13: Constructed two-channel task: 64 test roots, averaged over eight training streams. Every model predicts the true mean, so the Base−Point loss of 17.6 points (paired 64-root interval $[17.1, 18.1]$) comes from distribution shape.

Model	True point risk	Mean $\mathrm{KL}(P \,\|\, Q)$	Complete success
Base	0.788	0.0008	78.6%
Point	**0.645**	0.435	61.0%
MLE	0.788	0.0008	78.6%
Energy score	0.788	**0.0007**	78.6%
Direct success value	—	—	78.6%
Point→MLE	0.786	0.0014	78.6%
Point→Point	**0.592**	1.72	61.0%

Standard stochastic tasks without harm. A point reward can hurt a planner only if distorting the probabilities changes which action is best. We check this in three standard tasks without training any model: we plan once with the true kernel P and once with a fixed exponential tilt Q of it toward low point risk (Equation 3), and evaluate both plans exactly or by simulation under P. *Blackjack* (Gymnasium `Blackjack-v1`, infinite deck, Hamming point distance, tilt strength 2): the tilt lowers point risk from 0.572 to 0.526 and changes the first action with probability 4.0%, yet costs only 0.0009 in expected return per hand and 0.15 points of win rate. *MountainCar* (`MountainCar-v0` with a 25% hold of the previous action; 16 start states, 128 paired episodes each): all 4,096 episodes reach the goal under both plans. *CliffWalking* (`CliffWalking-v1` with `is_slippery=True`, 200-step budget): the plans differ at 1,334 state–time entries but share the first action, and the exact expected returns differ by 7×10^{-6} and 200-step completion by 0.0005 points. Because even the tilted true kernel barely changes returns, we do not train models on these tasks. Stochastic dynamics alone therefore do not create the hazard; it needs a decision whose ranking depends on how probability is split.

H Supplementary analysis: objective mixtures

The main claims do not rely on this analysis. It asks how much of a point reward's gain an objective can keep without task harm when a proper score or likelihood is mixed in.

FrozenLake. For each dynamics we reuse the five fresh-seed Bases, post-training samples, and minibatch streams of Table 10 and change only the objective: $\mathbb{E}[d(X, Y)] - w \cdot \frac{1}{2}\mathbb{E}[d(X, X')]$ with $w \in \{0, 0.25, 0.5, 0.75, 0.9, 1\}$ and KL coefficient 0.5; pure Point with KL coefficient $\beta \in \{0.1, 2, 8, 32\}$; and Point plus λ·NLL with $\lambda \in \{0.03, 0.1, 0.3, 1\}$. Success is the exact 200-step

value on each seed's training map. The $w = 0$, $w = 1$, and MLE arms reproduce the Point, ES, and MLE success values of Table 10 to within 0.42 points; the difference comes from thread-level floating-point nondeterminism. With $\beta = 0.1$, Point overshoots far beyond the default run and loses 54 and 59 points of success under the two dynamics.

MiniGrid. From each of Base seeds 2, 4, and 5 we train head-only continuations with the recipe and example order of the PointRGB arm: a full-RGB Euclidean objective $\mathbb{E}\|X - Y\| - w \cdot \frac{1}{2}\mathbb{E}\|X - X'\|$ with $w \in \{0, 0.25, 0.5, 0.75, 1\}$ (seed 2) or $\{0, 0.5, 1\}$ (seeds 4 and 5), and PointRGB plus $\lambda \cdot$ future-token NLL with $\lambda \in \{3, 6, 10, 30\} \times 10^{-4}$. Each seed is evaluated on the roots of its main result. A small likelihood term is enough to avoid harm: with $\lambda = 10^{-3}$, the success loss relative to Base is at most 0.31 points in the three seeds, with every interval covering zero, while about a fifth (20–22%) of PointRGB's RGB-MSE improvement is retained. Direct energy-score training from Base has harm between -0.5 and $+0.7$ points across seeds, so it neither reliably helps nor reliably hurts. In the supplementary code, the data are in `results/frontier/` and the plotting script is `figures/scripts/make_fig_frontier.py`.

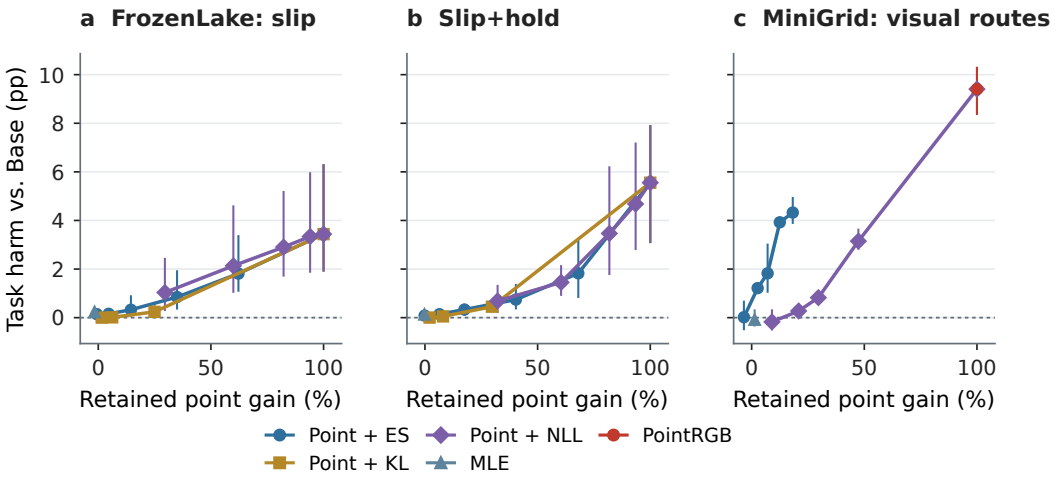


Figure A8: Task harm against retained point gain for three objective families. The horizontal axis is each arm's point-risk improvement over Base as a percentage of pure Point's; in MiniGrid it is the RGB-MSE improvement as a percentage of PointRGB's, and the ES series starts from a Euclidean Point loss. Markers are means over seeds, and bars span the seed range. In (c), the Euclidean mixtures with $w = 0.25$ and 0.75 were run for Base seed 2 only; the other MiniGrid markers average three seeds. In (a) and (b), Point with KL coefficient 0.1 loses 54 and 59 points and lies beyond the axes.