NEXTWM: NON-EXTENSIVE JOINT EMBEDDING PREDICTIVE ARCHITECTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

World models that plan in a learned latent space rely on a distributional regularizer to prevent representation collapse, and the target of that regularizer shapes the geometry in which the planner measures progress toward a goal. Current latent prediction world models fix this target to a Gaussian distribution $\mathcal{N}(0, 1)$. We ask what happens when that choice is relaxed. NextWM (Non-extensive World Model) replaces the Gaussian with a heavy-tailed q-Gaussian from Tsallis statistics, which recovers the Gaussian as a special case, while keeping the architecture, prediction loss, characteristic-function estimator and planner unchanged. Across four manipulation and navigation environments, NextWM predicts future latents more accurately than a trained world model with the Gaussian target on every task, yet its planning success still need further investigation. Our results shows the potential benefits of non-extensive targets for latent prediction world models, and motivates future work on how the geometry of the latent space affects planning.

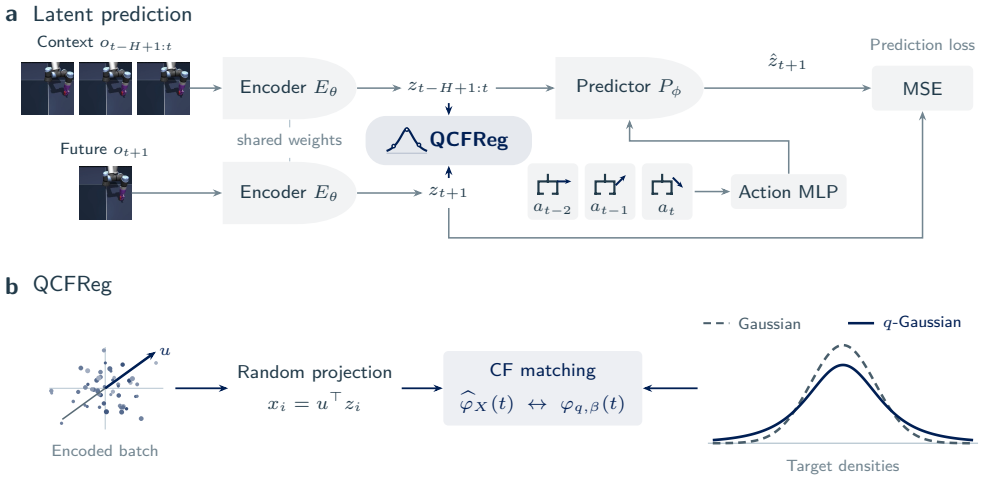


Figure 1: **NextWM overview. (a)** A shared encoder and action-conditioned predictor learn through latent prediction loss and QCFReg. **(b)** QCFReg compares empirical and target characteristic functions.

1 INTRODUCTION

A world model lets an agent imagine before it acts. Given the current observation and a sequence of candidate actions, it predicts what will happen, and a planner searches for the actions whose imagined outcome lies closest to a goal. Joint-embedding predictive architectures make this imagination cheap by predicting in a learned latent space rather than in pixel level (Bardes et al., 2024; Assran et al., 2025). The price of that efficiency is that the latent space itself must be learned, and it must be kept from collapsing to a trivial solution in which every observation looks the same.

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.

Recent work on joint-embedding world models address collapse with a distributional regularizer, LeWM (Maes et al., 2026) being the reference design: random one-dimensional projections of the latent batch are pushed towards a standard Gaussian while the encoder and predictor are trained end to end. This design is simple and stable, but it quietly makes the Gaussian part of the model. The same latent space that the regularizer shapes is the space in which the planner measures distance to the goal, so the choice of target distribution is not a neutral implementation detail; it decides which geometries the representation is allowed to adopt.

A Gaussian target is a strong assumption. A goodness-of-fit objective penalizes every departure from its reference, including heavy tails that might encode rare but informative states. The target law therefore does more than prevent collapse: it decides which structure the representation is allowed to keep. Nothing in this argument shows that control tasks need heavy tails; it does show that the target law is a modeling choice worth examining rather than a fixed default.

We introduce NextWM, which replaces the Gaussian target with a q-Gaussian drawn from the non-extensive statistics of Tsallis (Tsallis, 1988). A single entropic index controls how far the target departs from the Gaussian, which is recovered exactly in the limit. The change is deliberately minimal: the architecture, prediction loss, characteristic-function estimator and planner are all inherited from the reference design, so any difference in behavior must come from the representation that the new target induces. Based on our current knowledge, this family is distinct from the sparse ℓ_p targets explored by LpWM (Kuang et al., 2026) and there is no other works addressing this aspect of the distribution target.

The heavy-tailed target changes the representation in a measurable and, for prediction, consistently favorable way. Against a trained world model that keeps the Gaussian target, frozen-checkpoint rollouts show that NextWM forecasts its own future latents more accurately on all four tasks, at every horizon tested. In planning, it achieves the highest success on Push-T and remains competitive on the remaining tasks, with differences on the order of the evaluation variability (Figure 2). Taken together, these results isolate a question that a single success rate would hide: prediction accuracy and goal reaching can improve at different rates, because the planner depends not only on how well the model forecasts but on how its latent metric ranks candidate futures. Making that dependence visible is, we argue, the more useful outcome of this study.

Our contributions are:

- **QCFReg**, a sliced characteristic-function regularizer with a q-Gaussian target that generalizes the Gaussian regularizer of Maes et al. (2026) while leaving the rest of the world model unchanged (Section 3).
- **A four-environment planning evaluation** against a trained world model with the Gaussian target under a shared planner, revealing a task-dependent effect of the target distribution on goal reaching (Section 4).
- **A frozen-checkpoint rollout-accuracy probe** that scores multi-step latent prediction on paired validation windows with variance-normalized error and skill relative to persistence, showing that NextWM predicts more accurately on every task and exposing the gap between predictive accuracy and planning quality (Section 4.5).

The paper is organized as follows: Section 2 reviews related work. Section 3 describes the NextWM methodology. Section 4 presents the main experiments and results. Section 5 discusses limitations, and Section 6 concludes with the main results and recommnedation of future research.

2 Related Work

World models for planning. Learning a model of the environment and planning inside it has a long history in model-based reinforcement learning. Recurrent latent models showed that policies can be trained largely in imagination (Ha & Schmidhuber, 2018), and PlaNet and the Dreamer family scaled this idea to pixel observations by pairing a learned latent dynamics model with reconstruction or reward objectives (Hafner et al., 2019; 2023). TD-MPC instead couples a task-oriented latent model with model predictive control (Hansen et al., 2022). A more recent line of work plans towards visual goals without rewards: DINO-WM predicts in the feature space of a frozen pretrained encoder (Zhou et al., 2024), and PLDM learns latent dynamics from reward-free offline data (Sobal

et al., 2025). These methods typically search over action sequences with the cross-entropy method (Rubinstein, 1999), scoring candidates by a distance in latent space. That shared design is what makes the geometry of the representation directly relevant to control, and it is the setting we inherit.

Joint-embedding predictive architectures and collapse. Joint-embedding methods learn by predicting one view of the data from another in representation space, which avoids modeling pixel-level detail but admits trivial constant solutions (LeCun, 2022). Early approaches prevented collapse with architectural asymmetries such as target networks and stop-gradients (Grill et al., 2020; Assran et al., 2023; Bardes et al., 2024), or with explicit statistical penalties on the variance and covariance of embeddings (Zbontar et al., 2021; Bardes et al., 2022). V-JEPA 2 extends video joint-embedding models to action-conditioned prediction and robot planning (Assran et al., 2025). LeJEPA replaces these heuristics with SIGReg, which matches random projections of the embeddings to an isotropic Gaussian through a characteristic-function test in the spirit of Epps & Pulley (1983) (Balestriero & LeCun, 2025), and LeWM brings this regularizer to end-to-end world models trained from pixels (Maes et al., 2026). LpWM keeps the sketched regularizer but targets sparse generalized-Gaussian marginals (Kuang et al., 2026). Our work sits in this same lineage and asks a complementary question: rather than making the target sparser, we make it heavier-tailed while leaving everything else fixed.

Heavy tails and Tsallis statistics in learning. Heavy-tailed distributions have repeatedly proven useful where Gaussian assumptions are too restrictive. t-SNE uses a Student-t kernel in the embedding space to relieve crowding (van der Maaten & Hinton, 2008); Student-t priors and likelihoods make variational autoencoders more robust and better suited to heavy-tailed data (Takahashi et al., 2018; Kim et al., 2024); and deformed exponentials from Tsallis statistics underpin robust classification losses (Ding & Vishwanathan, 2010; Amid et al., 2019). The q-Gaussian arises as the maximizer of Tsallis entropy under a variance constraint (Tsallis, 1988) and coincides with a Student-t for $1 < q < 3$. To our knowledge, it has not previously been used as the target of a collapse-preventing regularizer in a world model. We use it for its single, interpretable index that interpolates continuously from the Gaussian, which makes the target law an explicit and testable modeling choice.

3 METHOD

We study how a distributional regularizer affects the latent space of an action-conditioned world model. The model is trained to predict in latent space and then used for model predictive control (MPC). Heavy-tailed targets are a modeling choice in this study; we do not assume that contact dynamics necessarily induce heavy-tailed representations.

3.1 PRELIMINARIES: ACTION-CONDITIONED JEPAS

An action-conditioned joint-embedding predictive architecture (JEPA) consists of an encoder E_θ and a predictor P_ϕ. The encoder maps an observation to $z_t = E_\theta(o_t)$; the predictor estimates $\hat{z}_{t+1} = P_\phi(z_{t-H+1:t}, a_{t-H+1:t})$ from H context observations and their actions. Following the training configuration of Maes et al. (2026), context and target observations share the encoder and both carry gradients. There is no exponential-moving-average target network or stop-gradient. A distributional regularizer discourages collapse alongside the prediction loss.

LeWM uses SIGReg (Sketched Isotropic Gaussian Regularizer) to compare random one-dimensional projections of the latents with a standard Gaussian target. This target enters the statistic as a null hypothesis.

3.2 NON-EXTENSIVE ENTROPY AND THE q-GAUSSIAN

The target family is motivated by Tsallis entropy (Tsallis, 1988). For a density $p(x)$,

$$S_q(p) = \frac{1}{q-1}\left(1 - \int p(x)^q \, \mathrm{d}x\right), \tag{1}$$

where q is the entropic index and the limit $q \to 1$ recovers Shannon entropy. We use the following one-dimensional q-Gaussian as the target for each random projection:

$$\begin{aligned} p_q(x) &\propto \exp_q\left(-\beta x^2\right), \\ \exp_q(x) &= [1 + (1-q)x]_+^{1/(1-q)}, \end{aligned} \tag{2}$$

For $1 < q < 3$, this density is a Student-t with $\nu = (3-q)/(q-1)$ degrees of freedom and tails proportional to $|x|^{-2/(q-1)}$. The variance is finite for $q < 5/3$; setting $\beta = 1/(5-3q)$ gives unit variance. The evaluated world-model targets lie in this finite-variance range; the fixed-density-scale implementation also supports heavier targets. The Gaussian is recovered as $q \to 1$; $q = 2$ gives the Cauchy outside our training range. These are one-dimensional identities: the joint synthetic distribution is an elliptical multivariate Student-t whose projections have this target law (Appendix F).

3.3 QCFREG: SLICED q-GAUSSIAN CHARACTERISTIC-FUNCTION REGULARIZATION

Our proposed regularizer, **QCFReg**, matches random one-dimensional projections of the encoded batch to a prescribed q-Gaussian through its characteristic function (CF). NextWM combines QCFReg with LeWM's unchanged architecture, prediction loss and regularizer weight. The modification to SIGReg is the reference CF, parameterized by a fixed shape q and density scale β (Figure 1); no sample likelihood or explicit Tsallis-entropy term is optimized.

Prediction loss. The transition dynamics are trained with a mean squared error in latent space,

$$\mathcal{L}_{\text{pred}} = \mathbb{E}\left[\tfrac{1}{d}\|\hat{z}_{t+1} - z_{t+1}\|_2^2\right], \tag{3}$$

where d is the latent dimension and z_{t+1} carries gradient.

Projected empirical distribution. Let $Z \in \mathbb{R}^{B\times T\times d}$ contain encoded observations from B windows at $T = H + 1$ frame positions. At each optimizer step, draw P directions $u_p = g_p/\|g_p\|_2$, with $g_p \sim \mathcal{N}(0, I_d)$, and set $x_{i\tau p} = u_p^\top z_{i\tau}$. Directions are shared across frame positions within the step and redrawn at the next step. For each frame position and direction, form the empirical CF

$$\widehat{\varphi}_{\tau p}(t) = \frac{1}{B}\sum_{i=1}^{B} e^{\mathrm{i}tx_{i\tau p}} = c_{\tau p}(t) + \mathrm{i}s_{\tau p}(t). \tag{4}$$

The batch average is taken over windows separately at each frame position, rather than pooling context and target frames into one empirical distribution. A centered symmetric target has a real CF, so QCFReg penalizes both the real residual $c - \varphi_{q,\beta}$ and the imaginary component s.

Target CF with explicit scale. For $1 < q < 3$, write $\nu = (3-q)/(q-1)$ and $s_0 = [(3-q)\beta]^{-1/2}$. The target is a Student-t with degrees of freedom ν and scale s_0, whose CF is

$$\varphi_{q,\beta}(t) = \frac{K_{\nu/2}(r)\, r^{\nu/2}}{\Gamma(\nu/2)\, 2^{\nu/2-1}}, \qquad r = \sqrt{\nu}\, s_0|t|, \qquad \varphi_{q,\beta}(0) = 1. \tag{5}$$

Here K_a is the modified Bessel function of the second kind. At $q = 1$ the implementation uses the exact Gaussian CF $\exp[-t^2/(4\beta)]$; $(q, \beta) = (1, 1/2)$ therefore recovers SIGReg, including its loss and latent gradients with matched projection draws. The earlier unit-variance convention is the special case $\beta = 1/(5-3q)$, for which $r = |t|\sqrt{\nu - 2}$. Writing the scale explicitly avoids confusing that convention with the main $(1.5, 1/2)$ target.

Matching objective and numerical estimator. Using the same Gaussian frequency window $w(t) = e^{-t^2/2}$ as SIGReg, the population-over-directions form is

$$\mathcal{R}_q = \frac{2B}{T}\sum_{\tau=1}^{T}\mathbb{E}_u \int_0^{t_{\max}} |\widehat{\varphi}_{\tau u}(t) - \varphi_{q,\beta}(t)|^2 w(t)\,\mathrm{d}t. \tag{6}$$

The factor two follows the shared implementation's symmetric-frequency convention. Let $t_k = k\Delta$, $k = 0, \dots, K-1$, where $\Delta = t_{\max}/(K-1)$. The implemented estimator is

$$\widehat{\mathcal{R}_q} = \frac{B}{TP} \sum_{\tau=1}^{T} \sum_{p=1}^{P} \sum_{k=0}^{K-1} a_k \left[(c_{\tau p}(t_k) - \varphi_{q,\beta}(t_k))^2 + s_{\tau p}(t_k)^2 \right], \tag{7}$$

where $a_k = \Delta w(t_k)$ at the two endpoints and $a_k = 2\Delta w(t_k)$ otherwise. We use $B = 128$, $P = 1024$, $K = 17$ and $t_{\max} = 3$. This finite grid and finite set of directions provide a sampled distribution-matching objective; a small loss is not proof that the full learned joint distribution equals a multivariate target.

Implementation and learning signal. The target CF is computed once at the fixed knots, using numerically stabilized Bessel evaluation, and stored as a buffer. Neither q nor β receives gradients; training differentiates through the projected latents, sine/cosine averages and the shared encoder. There is no per-batch standardization of projections. Consequently, QCFReg constrains both projected shape and scale, unlike a penalty on variance alone. Its step-time operations are the same as SIGReg's, with a different precomputed target vector; this is an implementation observation, not a measured speedup. The actual batch gradient is given below.

Objective. NextWM keeps LeWM's two-term structure, with $\lambda = 0.09$:

$$\mathcal{L}_{\text{NextWM}} = \mathcal{L}_{\text{pred}} + \lambda \mathcal{R}_q. \tag{8}$$

Role at planning time. QCFReg is used only during training. At inference, CEM retains LeWM's squared Euclidean distance between the final predicted latent and the encoded goal; it does not evaluate the q-Gaussian density or its negative log-likelihood. Any planning difference must therefore arise through the learned encoder and predictor, rather than through a new inference-time penalty.

3.4 Fixed target indices

Each run fixes $q = 1.5$ and $\beta = 1/2$ in the normalized density proportional to $[1 + (q-1)\beta x^2]_+^{-1/(q-1)}$. For $q < 1$ the support is $|x| \le [(1-q)\beta]^{-1/2}$; at $q = 1$ the target is Gaussian with variance $1/(2\beta)$ (standard at $\beta = 1/2$); for $q > 1$ it is Student-t with $\nu = (3-q)/(q-1)$ and scale $s = [(3-q)\beta]^{-1/2}$. At $q = 2$ this is Cauchy and has no finite variance. The new runs fix this density scale across environments, rather than imposing unit variance. The unit-variance special case above is the convention of the earlier reference; it is not the normalization of the $q = 1.5$ world-model runs. Fixed target characteristic functions are computed once at the integration knots and stored as buffers; no index optimizer or interpolation is used.

3.5 What changes in the actual regularizer?

The training signal is a batch-level characteristic-function discrepancy, not a scalar negative log-density. For one direction, let $x_i = u^\top z_i$, $c_k = B^{-1} \sum_i \cos(t_k x_i)$ and $s_k = B^{-1} \sum_i \sin(t_k x_i)$. With a_k denoting the nonnegative quadrature/window weights used by the implementation, its statistic is $R = B \sum_k a_k [(c_k - \varphi_k)^2 + s_k^2]$. Direct differentiation gives

$$\frac{\partial R}{\partial x_i} = 2 \sum_k a_k t_k \left[-(c_k - \varphi_k) \sin(t_k x_i) + s_k \cos(t_k x_i) \right]. \tag{9}$$

The batch-size factor cancels, and $|\partial R / \partial x_i| \le 6 \sum_k a_k |t_k|$ for every target considered here. Averaging directions and frame positions preserves boundedness. This is a latent-gradient bound; encoder parameter gradients also depend on the encoder Jacobian. Gaussian and q-Gaussian targets share this property; the target changes the residual, not the boundedness argument. The density derivatives and elementary kurtosis correspondence are supplied only as reference identities in Appendix F.1. They neither explain improved planning nor cover the fourth moment of the $q = 1.5$ target.

Separating shape from scale. For $1 \leq q < 5/3$, target variance is $v = [\beta(5 - 3q)]^{-1}$. Hence the historical $(q, \beta) = (1.25, 0.8)$ target has $v = 1$, whereas $(1.5, 0.5)$ has $v = 4$. A shape comparison must hold v fixed by choosing $\beta = 1/[v(5 - 3q)]$. This still changes the target shape, including its central concentration, not just the remote tails. Rescaling all predicted and goal latents by a multiplies the squared planning cost by a^2, preserving candidate rankings for rank-based CEM under exact arithmetic. However, during training it multiplies prediction MSE by a^2 and changes the CF residual nonlinearly on the fixed frequency grid. Thus latent scale can affect the learned predictor and the balance of loss terms even if a uniform post-hoc rescaling cannot change ideal CEM rankings.

4 EXPERIMENTS

4.1 SETUP

We evaluate Push-T, OGBench-Cube, Reacher and Two-Room using the same CEM planning configuration and a budget of 50 environment steps. Each model is evaluated on 50 dataset start/goal pairs for each of seeds 42, 43 and 44; the same pairs are selected across models within an environment and seed. Goals are 25 dataset steps ahead of the start. The new fixed-q arm trains for 155,000 optimizer steps with training seed 3072. Released LeWM checkpoints provide the reference. Appendix A gives the protocol; Appendix B gives every evaluation seed, and Appendix E gives the training diagnostics.

Table 1: **Exploratory local planning results (%).** Mean $\pm$ sample SD across three evaluation seeds of one checkpoint per task and arm, with 50 episodes per seed and a 50/25 budget/goal offset. Bold cells mark the highest mean per task, including ties, tinted green when a NextWM arm is best and red when LeWM is best; it does not indicate statistical significance. These uncertainties describe evaluation variability only. The $q = 1.25$ and $q = 1.5$ targets have variance 1 and 4 respectively. Released LeWM is re-evaluated locally but its training is unmatched.

Checkpoint	Push-T	OGBench-Cube	Reacher	Two-Room
NextWM ($q = 1.25$, unit variance)	88.7 ± 4.2	$\mathbf{71.3 \pm 9.5}$	80.0 ± 12.2	74.0 ± 3.5
NextWM ($q = 1.5$)	$\mathbf{93.3 \pm 3.1}$	64.7 ± 10.1	80.0 ± 2.0	81.3 ± 3.1
LeWM (released)	90.7 ± 7.0	69.3 ± 12.1	$\mathbf{82.7 \pm 4.2}$	$\mathbf{85.3 \pm 3.1}$

4.2 PLANNING PERFORMANCE

Figure 2 and Table 1 report the historical local evaluations. NextWM at $q = 1.5$ achieves $93.3 \pm 3.1\%$ success on Push-T, compared with $90.7 \pm 7.0\%$ for released LeWM and $88.7 \pm 4.2\%$ for the unit-variance $q = 1.25$ reference. Its mean is lower than LeWM's on OGBench-Cube ($64.7 \pm 10.1\%$ versus $69.3 \pm 12.1\%$), Reacher ($80.0 \pm 2.0\%$ versus $82.7 \pm 4.2\%$), and Two-Room ($81.3 \pm 3.1\%$ versus $85.3 \pm 3.1\%$). These are descriptive differences across three evaluation seeds, not evidence of statistical significance.

Relative to $q = 1.25$, the $q = 1.5$ arm increases mean success by 4.7 percentage points on Push-T and 7.3 on Two-Room, ties on Reacher, and decreases it by 6.7 on OGBench-Cube. The existing $q = 1.25$ result uses unit variance ($\beta = 0.8$), whereas $q = 1.5$ uses $\beta = 1/2$. Consequently, this comparison changes both tail weight and target variance and does not isolate the effect of q.

4.3 VALIDATION DIAGNOSTICS

At the final optimizer step, the $q = 1.5$ models have mean latent variance between 2.54 and 2.78, effective rank between 61.7 and 74.3 in a 192-dimensional representation, and empirical excess kurtosis between 1.53 and 2.21 (Appendix E). These diagnostics show nonzero latent spread but do not establish that the learned distribution matches the target or explain the ordering of planning scores. The prescribed $q = 1.5$ target has finite variance but no finite fourth moment; empirical kurtosis is a finite-sample diagnostic.

4.4 GEOMETRY AND PLANNING OUTCOMES

Tables 6 and 1 compare the existing validation diagnostics and planning scores. Across all four tasks, switching between the historical targets increases learned variance and decreases effective rank. Planning improves on two tasks, is unchanged on one, and declines on Cube. These checkpoint-level summaries therefore do not identify a sufficient condition for better planning. They are not paired trajectory measurements or causal mediation evidence.

4.5 FROZEN-CHECKPOINT ROLLOUT ACCURACY

We evaluate the completed $q = 1.5$ and released LeWM checkpoints without updating their parameters. For each task, we select 256 distinct dataset episodes, with one eligible anchor per episode sampled from NextWM's saved local validation-window split (split seed 3072; sampling seed 20260925). Each window supplies three observed context frames and five future frames at frameskip 5. The predictor rolls forward under the recorded actions, feeding back its own predictions, with no teacher forcing after the context. Unlike the training diagnostic averaged over all context positions, the one-step probe always uses the full three-frame context. Both checkpoints see exactly the same windows and preprocessing.

For horizon $h \in \{1, 3, 5\}$, we report prediction MSE divided by that checkpoint's mean coordinate variance over the encoded observations in the selected windows. This normalization removes uniform latent scale; it does not make representation difficulty identical across models. Figure 3 visualizes the horizon dependence. We also report prediction skill relative to a persistence forecast, $S_h = 1 - \mathrm{MSE}_h / \mathrm{MSE}_{\mathrm{persist},h}$, where persistence repeats the final context latent. Positive skill indicates improvement over that simple predictor. Paired bootstrap intervals use 2,000 resamples of the common episode anchors and hold the estimated normalization fixed; these quantify evaluation-sample variability for fixed checkpoints, not training uncertainty.

Table 2: **Frozen-checkpoint rollout accuracy.** Each task uses 256 paired validation-window anchors. The three error columns report variance-normalized latent MSE; the last column reports skill relative to persistence. Arrows indicate the preferred direction; bold cells mark the best value within each task, including ties, tinted green when a NextWM arm is best and red when LeWM is best. Horizon h counts latent steps of five environment steps each. These windows are not a guaranteed unseen-trajectory test for both checkpoints.

Task	Checkpoint	Normalized MSE ↓			Persistence skill ↑
		$h = 1$	$h = 3$	$h = 5$	$h = 5$ (%)
Push-T	NextWM ($q = 1.5$)	**0.0017**	**0.0065**	**0.0181**	**98.8**
	LeWM (released)	0.0072	0.0257	0.0736	94.8
OGBench-Cube	NextWM ($q = 1.5$)	**0.0013**	**0.0040**	**0.0121**	**99.2**
	LeWM (released)	0.0040	0.0134	0.0275	97.9
Reacher	NextWM ($q = 1.5$)	**0.0013**	**0.0021**	**0.0024**	**99.8**
	LeWM (released)	0.0071	0.0140	0.0185	98.7
Two-Room	NextWM ($q = 1.5$)	**0.0013**	**0.0045**	**0.0141**	**99.3**
	LeWM (released)	0.0613	0.0974	0.1438	92.6

NextWM has lower variance-normalized rollout error at every tested horizon in all four tasks. At $h = 5$, its NMSE is 0.0181 versus 0.0736 on Push-T, 0.0121 versus 0.0275 on Cube, 0.0024 versus 0.0185 on Reacher, and 0.0141 versus 0.1438 on Two-Room. Both checkpoints improve substantially over persistence. The prediction-error ordering therefore differs from the historical planning ordering on three tasks: better prediction in a model's own latent metric is not sufficient to establish better goal-directed planning. Different representation geometry, unmatched training, and the validation-window scope prevent attributing these differences specifically to the heavy-tailed target.

The same frozen-checkpoint probe also provides a scale-invariant geometry comparison (Table 4; Figure 6). The ratio of the 95th percentile to the median centered latent norm is 2.13–2.30 for NextWM versus 1.06–1.15 for LeWM; effective ranks are 64.4–81.0 versus 93.6–122.2. This shows

greater radial dispersion and lower effective rank in the evaluated NextWM representations, using a quantile statistic that does not require a finite fourth moment. These differences occur on tasks with both favorable and unfavorable planning outcomes and therefore are not a sufficient explanation for improved planning. The rollout probe uses recorded actions, whereas planning scores candidate action sequences; candidate-ranking fidelity remains a separate question.

An additional planning evaluation compares CEM budgets (Appendix D), always using frozen checkpoints. This test assesses practical behavior of the available models; it does not isolate the effect of the training target.

5 LIMITATIONS

Evidence and attribution. Each new model uses one training seed, so evaluation-seed variability does not quantify variability across independent training runs. Released LeWM checkpoints are evaluated under the same planning protocol but are not a matched retraining control, and the two NextWM references differ in normalization. A controlled index-and-scale ablation with multiple training seeds is needed to attribute changes specifically to tail weight. The observed Push-T gain therefore does not establish a tail-specific improvement or a general practical advantage. The current evidence does not identify when a new task or independently trained model will benefit. Validation windows may overlap training windows in time, and the released checkpoint's original split is unknown; the rollout probe therefore does not establish generalization to unseen trajectories for both methods. The elementary finite-kurtosis correspondence and scalar density derivatives concern reference laws; they are not learning guarantees and do not explain the planning results. The $q = 1.5$ target lies outside the finite-kurtosis correspondence's range.

Required controls and extensions. The priority is a controlled index-by-variance study with a matched, retrained Gaussian baseline and independent training seeds. A proposed design crosses $q \in \{1, 1.25, 1.5\}$ with target variance $v \in \{1, 4\}$, choosing $\beta = 1/[v(5 - 3q)]$ and keeping the architecture, split, compute budget, frequency grid and regularizer weight fixed. Three fresh training seeds per cell and task would require 72 trainings across the four tasks; evaluation seeds should be averaged within each checkpoint before reporting training-seed variability and paired differences. This is future work: the full-budget campaign has not been run and contributes no results to this paper.

Beyond the completed frozen-checkpoint rollout probe, a further extension should connect latent geometry to planning outcomes by measuring latent goal-distance rankings versus realized goal progress, prediction error, and robust tail and anisotropy statistics on the same planning episodes. Consistent fixed-variance improvements over the retrained Gaussian, together with better rollout or goal-ranking fidelity, would support a target-shape explanation. If a larger-variance Gaussian reproduces the gain, scale remains a sufficient explanation. Frequency-scale and loss-weight controls can then test changes in objective balance. These are proposed tests, not findings from the current checkpoints. The present evidence does not establish a task-selection rule or a general practical advantage for heavier tails.

6 CONCLUSIONS

NextWM turns the target distribution of a characteristic-function regularizer into an explicit modeling choice while leaving the world-model architecture, estimator and planner untouched. Relaxing the Gaussian to a heavy-tailed q-Gaussian has a clear and consistent effect on what the model learns: against a trained world model with the Gaussian target, NextWM predicts its own future latents more accurately on all four environments and at every horizon tested. In planning it reaches the highest success on Push-T and remains competitive elsewhere, with differences comparable to the evaluation variability; the unit-variance $q = 1.25$ reference shifts the task ordering again, and a planner-budget study shows that the ordering also depends on search compute.

The value of these results lies in what they separate. Prediction accuracy and goal reaching respond differently to the target law, so the latent metric that ranks candidate futures, and not only forecasting quality, governs planning. Because the present arms change tail weight and target variance together

and use a single training seed, we frame the planning gains as motivation rather than as an established improvement.

Future work. Matched index-and-scale experiments and independent training replication are the immediate extensions, followed by episode-level goal-ranking and geometry measurements (Section 5). Another research direction is to explore a learnable q index for the target distribution.

AI USE STATEMENT

Generative AI tools assisted with manuscript grammar review and code generation. The authors are responsible for the final content of the submission.

ETHICS STATEMENT

The reported experiments concern synthetic dynamics, simulated robot control, and released visual benchmark datasets.

REFERENCES

Ehsan Amid, Manfred K. Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on Bregman divergences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1906.03361.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2301.08243.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025. URL https://arxiv.org/abs/2511.08544.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2105.04906.

Adrien Bardes et al. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024. URL https://arxiv.org/abs/2404.08471.

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 (RSS)*, 2023. URL https://arxiv.org/abs/2303.04137.

Nan Ding and S. V. N. Vishwanathan. t-logistic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983.

Jean-Bastien Grill et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2006.07733.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://arxiv.org/abs/1803.10122.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. URL https://arxiv.org/abs/2301.04104.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2203.04955.

Juno Kim, Jaehyuk Kwon, Mincheol Cho, Hyunjong Lee, and Joong-Ho Won. t^3-variational autoencoder: Learning heavy-tailed data with student's t and power divergence. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2312.01133.

Yilun Kuang, Yash Dagade, Quentin Le Lidec, Lucas Maes, Randall Balestriero, and Yann LeCun. LpWM: A case for sparse representations in world models, 2026. URL https://arxiv.org/abs/2608.22764.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022. URL https://openreview.net/pdf?id=BZ5a1r-kVsf.

Lucas Maes et al. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. URL https://arxiv.org/abs/2603.19312.

Seohong Park et al. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.20092. arXiv preprint arXiv:2410.20092.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999.

Vlad Sobal et al. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025. URL https://arxiv.org/abs/2502.14819.

Hiroshi Takahashi, Tomoharu Iwata, Yuki Yamanaka, Masanori Yamada, and Satoshi Yagi. Student-t variational autoencoder for robust density estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics*, 52(1-2):479–487, 1988. URL https://link.springer.com/article/10.1007/BF01016429.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning (ICML)*, 2021. URL https://arxiv.org/abs/2103.03230.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024. URL https://arxiv.org/abs/2411.04983.

APPENDIX GUIDE

The appendix follows the empirical evidence: protocol and checkpoint evaluations (Appendices A–B), three-dimensional latent geometry (Appendix C), planner-budget sensitivity (Appendix D), and training diagnostics (Appendix E). Target identities support the method in Appendix F. Recorded episodes for all four tasks close the appendix (Appendix G).

A EXPERIMENTAL PROTOCOL

The historical NextWM arms share the intended architecture, data pipeline and optimizer settings, but change both index and target variance. They do not form a controlled index-only sweep. Released checkpoints are evaluated in the same local planner, but their original training and published protocols need not match ours. The code, configuration files and Slurm scripts accompany the paper.

A.1 DATA AND ARCHITECTURE

Data. Push-T (Chi et al., 2023), OGBench-Cube (single cube, expert) (Park et al., 2025), Reacher and Two-Room, using the dataset mirrors distributed for the LeWM/stable-worldmodel pipeline. Frames are 224×224 RGB; the dataset is subsampled with a frameskip of 5, and each training window holds $H = 3$ context frames and one target frame, so the action encoder sees $5\times$ the action dimension. A seeded $90/10$ split is applied to the training dataset by the data pipeline.

Architecture. LeWM's, with 18.0M parameters. The encoder is a ViT-tiny (embedding dimension $d = 192$, patch size 14) trained from scratch, shared between context and target frames. The predictor is a causal transformer with AdaLN-zero action conditioning: depth 6, 16 heads of dimension 64, MLP width 2048, dropout 0.1. Actions are embedded by an MLP into $\mathbb{R}^{192}$. There is no target network, no EMA and no stop-gradient.

A.2 TRAINING CONFIGURATION

Regularizer. Both arms use $P = 1024$ random unit projections, 17 trapezoidal knots on $[0, 3]$, frequency window $e^{-t^2/2}$ and weight $\lambda = 0.09$. The per-frame estimator and symmetric-frequency convention are specified in Section 3.3; only the fixed target CF changes.

Optimization. AdamW with learning rate 5×10^{-5} and weight decay 10^{-3}, linear warm-up followed by cosine annealing, batch size 128, bf16 mixed precision, gradient-norm clipping at 1.0. The new runs use seed 3072 and $155{,}000$ optimizer steps. Validation uses 50 held-out batches.

A.3 COMPARISON PROTOCOL

Checkpoints. Each environment has one fixed $q = 1.5$ NextWM checkpoint with $\beta = 1/2$. Existing unit-variance $q = 1.25$ checkpoints are reused without training. Released LeWM checkpoints provide the main reference under the same planner. Published scores remain separate.

Planning and evaluation. Standard CEM uses 300 candidates, 30 refinements, 30 elites and unit initial variance. The horizon is five blocks of five environment steps, with replanning after each block. Each checkpoint is evaluated on 50 matched dataset start/goal pairs per seed (42, 43, 44), with a 50-step budget and goals 25 dataset steps ahead. Pair selection and CEM share the evaluation seed; the historical retrained Push-T reference predates CEM seeding. Reported SDs measure evaluation variability, not independent-training uncertainty.

B Checkpoint Evaluation

B.1 Planning performance across evaluation seeds

Figure 2 makes the task dependence of the local comparison visible: $q = 1.5$ has the highest mean on Push-T, the unit-variance $q = 1.25$ reference on Cube, and released LeWM on Reacher and Two-Room. Table 3 exposes the evaluations behind those means. The practical finding is a change in task ordering rather than a uniform gain. The intervals summarize repeated planning evaluations of fixed checkpoints; they do not establish how that ordering would vary after independent training.

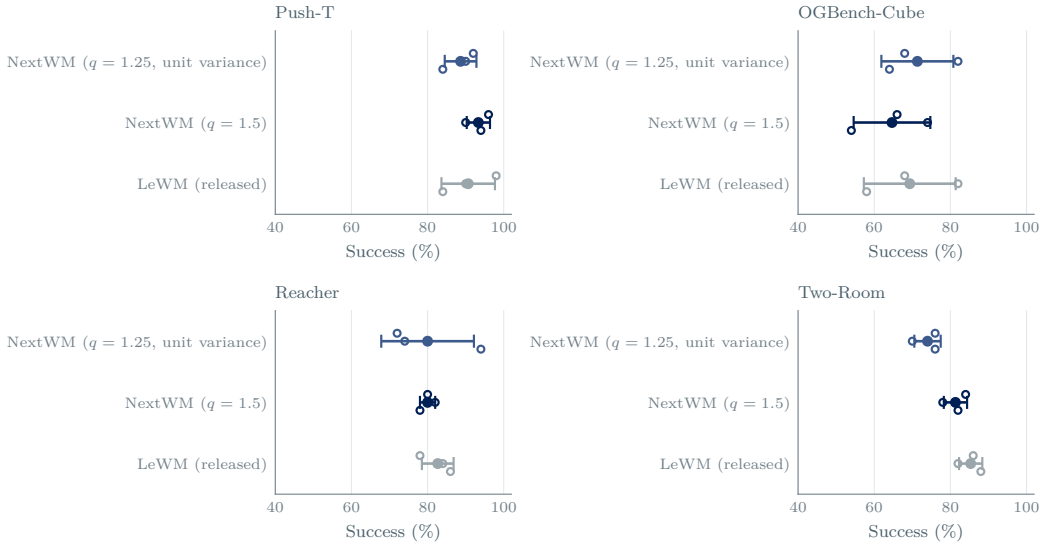


Figure 2: **Local planning performance.** Means, sample SDs and individual evaluation seeds for 50 episodes per seed. The $q = 1.25$ and $q = 1.5$ targets have variance 1 and 4. Released LeWM is evaluated with the same planner but is not a matched retraining control. Published scores are excluded.

Table 3: **Per-seed planning success (%).** All local evaluations underlying Table 1, plus the historical retrained Push-T reference. Bold cells mark the best primary-arm value per task and column, including ties, tinted green when a NextWM arm is best and red when LeWM is best. The retrained reference used unseeded CEM and is excluded from highlighting.

Environment	Model	Seed 42	Seed 43	Seed 44	Mean ± SD
Push-T	NextWM ($q = 1.25$, unit variance)	92.0	**90.0**	84.0	88.7 ± 4.2
	NextWM ($q = 1.5$)	96.0	**90.0**	**94.0**	**93.3 ± 3.1**
	LeWM (released)	**98.0**	**90.0**	84.0	90.7 ± 7.0
	LeWM (retrained)	98.0	92.0	96.0	95.3 ± 3.1
OGBench-Cube	NextWM ($q = 1.25$, unit variance)	**68.0**	**82.0**	**64.0**	**71.3 ± 9.5**
	NextWM ($q = 1.5$)	66.0	74.0	54.0	64.7 ± 10.1
	LeWM (released)	**68.0**	**82.0**	58.0	69.3 ± 12.1
Reacher	NextWM ($q = 1.25$, unit variance)	72.0	74.0	**94.0**	80.0 ± 12.2
	NextWM ($q = 1.5$)	**80.0**	82.0	78.0	80.0 ± 2.0
	LeWM (released)	78.0	**84.0**	86.0	**82.7 ± 4.2**
Two-Room	NextWM ($q = 1.25$, unit variance)	76.0	70.0	76.0	74.0 ± 3.5
	NextWM ($q = 1.5$)	84.0	78.0	82.0	81.3 ± 3.1
	LeWM (released)	**86.0**	**82.0**	**88.0**	**85.3 ± 3.1**

B.2 PREDICTION OVER MULTIPLE LATENT STEPS

Figure 3 asks whether the checkpoint ordering changes when prediction is evaluated directly under recorded actions. NextWM has lower variance-normalized error at every tested horizon in all four tasks. At five latent steps, its errors range from 0.0024 to 0.0181, compared with 0.0185 to 0.1438 for LeWM (Table 2). Both models improve over persistence. This is consistent evidence of better prediction in each model's own latent metric, but its disagreement with planning rankings shows why rollout accuracy alone cannot explain goal-reaching performance. Candidate-action ranking remains unmeasured.

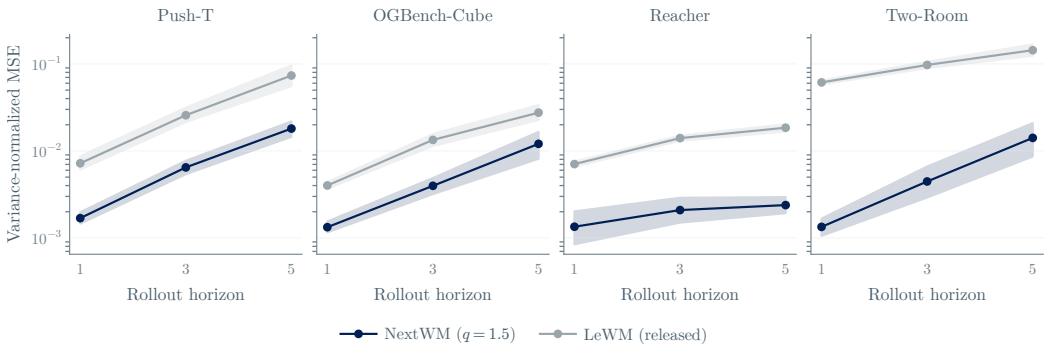


Figure 3: **Frozen-checkpoint rollout error.** Variance-normalized latent MSE at horizons 1, 3 and 5; one latent step spans five environment steps. Shading is a percentile 95% bootstrap interval over 256 paired anchors, conditional on each checkpoint and its estimated normalization. The validation windows are not a guaranteed unseen-trajectory test for both models.

C THREE-DIMENSIONAL LATENT GEOMETRY

Shape after removing uniform scale. Figure 4 compares how the encoded observations occupy a shared three-dimensional projection. LeWM forms a compact central region, whereas NextWM's 90% density envelope extends farther and is less uniform around the center in each task. Because one scalar standard deviation has been removed from each representation, these differences are not simply a global enlargement of the same projected cloud. They reflect how variance and probability mass are distributed in the selected directions; they do not identify a full-dimensional manifold or a target tail index.

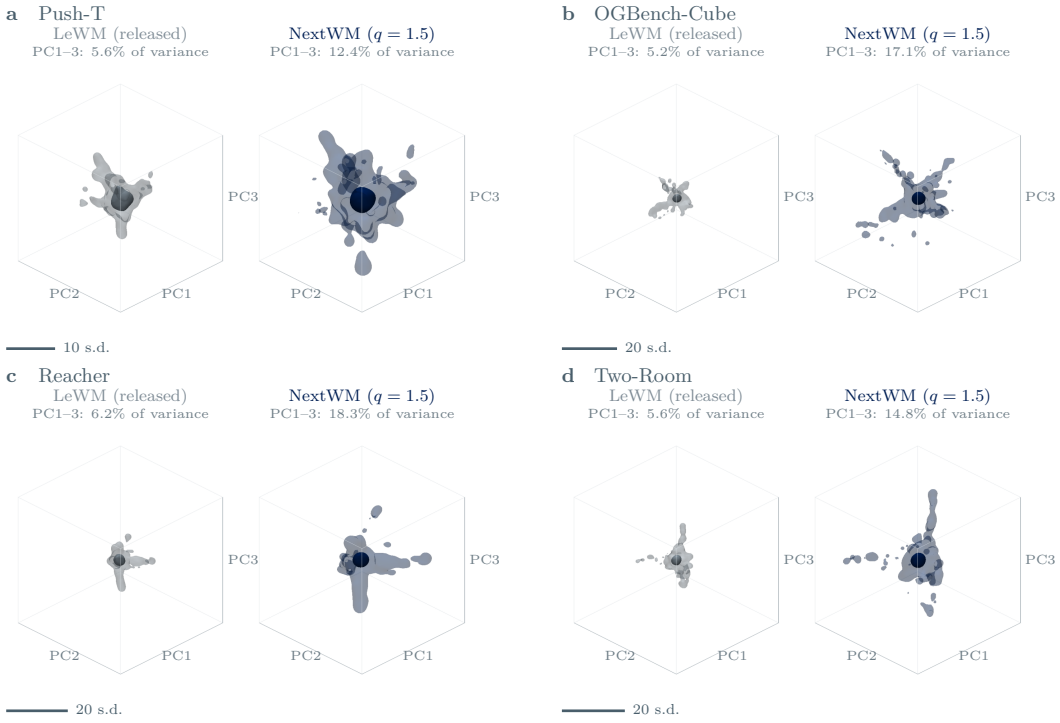


Figure 4: **Latent geometry with uniform scale removed.** Released LeWM (amber, left) and NextWM ($q = 1.5$; blue, right). Solid cores and translucent envelopes contain 50% and 90% of the estimated 3D density mass. Each task pair shares PCA axes, origin, coordinate scale and camera. Surfaces use 2,048 observed frames per model and are not confidence regions.

Restoring the learned scale. Figure 5 restores each checkpoint's scalar scale while keeping the fitted density shapes and projection fixed. The larger extent of NextWM becomes more pronounced, consistent with its higher mean coordinate variance. Reading the two views together separates visible projected shape from uniform size; neither view separates the effects of target index and target variance during training.

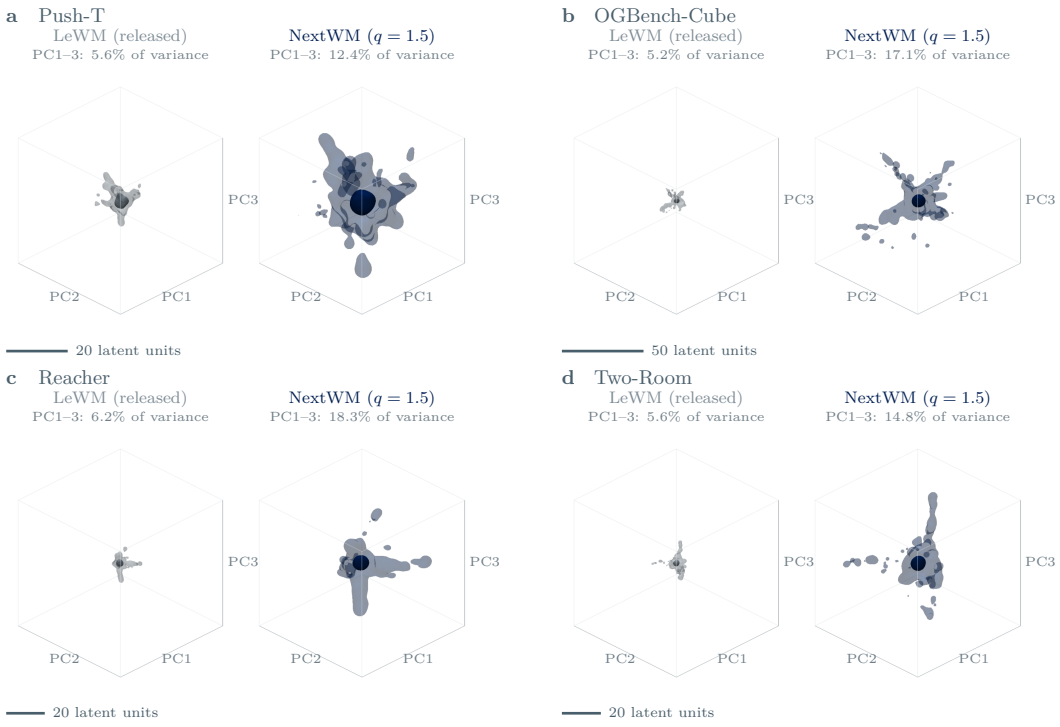


Figure 5: **Latent geometry at original scale.** The same surfaces and camera as Figure 4, with each checkpoint's learned uniform scale restored. Coordinates remain matched within each task.

Full-dimensional measurements. The summary statistics in Figure 6 and Table 4 provide the quantitative counterpart to the density views. NextWM has mean coordinate variance 2.53–2.85 versus 0.97–1.01 for LeWM, lower effective rank (64.4–81.0 versus 93.6–122.2), and greater radial dispersion (2.13–2.30 versus 1.06–1.15). Rank and radial dispersion are invariant to uniform rescaling, so a single scale factor cannot account for all observed differences.

Table 4: **Geometry on the paired rollout anchors.** Statistics pool eight frames from each of 256 anchors. Effective rank is the exponential entropy of normalized singular values. Radial dispersion is $Q_{.95}/Q_{.50}$ of centered latent norms and does not require a finite fourth moment.

Task	Checkpoint	Mean variance	Effective rank	Radial $Q_{.95}/Q_{.50}$
Push-T	NextWM ($q = 1.5$)	2.528	81.0	2.16
	LeWM (released)	1.004	113.3	1.09
OGBench-Cube	NextWM ($q = 1.5$)	2.848	76.4	2.30
	LeWM (released)	1.006	122.2	1.06
Reacher	NextWM ($q = 1.5$)	2.550	64.4	2.15
	LeWM (released)	0.987	93.6	1.12
Two-Room	NextWM ($q = 1.5$)	2.610	70.0	2.13
	LeWM (released)	0.975	112.6	1.15

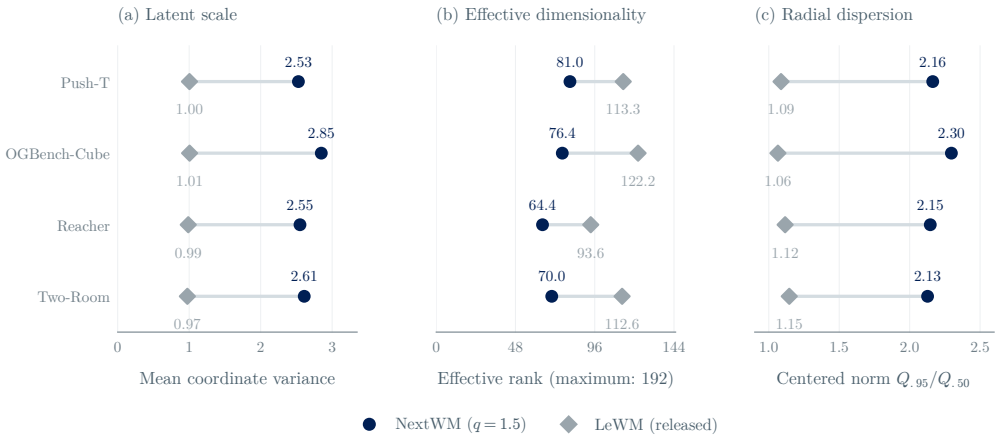


Figure 6: **Scale, dimensionality and radial dispersion.** Connected values pair checkpoints within each task. All measurements use the original 192-dimensional embeddings; no training-seed uncertainty is implied.

Construction and interpretation. Each cloud is centered and divided by the square root of its mean coordinate variance, without coordinate-wise whitening. Distance-preserving orthogonal Procrustes alignment uses paired frames; PCA is fitted to the pooled normalized pair. Three components retain only 5.2–6.2% of LeWM variance and 12.4–18.3% of NextWM variance. A Gaussian KDE with Scott bandwidth is integrated on an 88^3 grid extending four marginal kernel standard deviations beyond the observed range. Marching cubes extracts the 50%/90% highest-density surfaces; the outer 10% of estimated mass is omitted. Original-scale views rescale the same meshes. Projection and KDE smoothing affect appearance; the surfaces are neither prescribed target samples nor uncertainty regions.

Conclusion. NextWM exhibits greater radial dispersion and lower effective rank, with projected shape differences persisting after scale normalization. These changes accompany mixed planning outcomes, so they do not establish a general control advantage. Matched training runs that separate target index from variance are needed to connect latent geometry to goal-reaching.

D PLANNER-BUDGET SENSITIVITY

We compare both frozen checkpoints at three CEM compute settings: 100 candidates and 10 refinement iterations (10 elites), the standard 300 candidates and 30 iterations (30 elites), and 600 candidates and 30 iterations (60 elites). The elite fraction remains 10%; the model horizon, action-block size, 50-step environment budget and 25-step goal offset remain fixed. The low setting changes both candidate count and refinement count, so this is a compute-budget comparison rather than an isolated one-parameter ablation. Every task, checkpoint and setting uses pair-selection and CEM seeds 42/43/44, with the same 50 start/goal pairs within each seed.

Table 5: Planner-budget sensitivity: success (%) as mean ± sample SD over three evaluation seeds of each frozen checkpoint. Bold cells mark the highest mean within each task and budget, including ties, tinted green when a NextWM arm is best and red when LeWM is best. Low, standard and high correspond to 100×10, 300×30 and 600×30 candidates×iterations.

Task	Checkpoint	Low	Clean / standard	High
Push-T	NextWM ($q = 1.5$)	**84.0 ± 6.0**	**93.3 ± 3.1**	**95.3 ± 1.2**
	LeWM (released)	83.3 ± 6.4	90.7 ± 5.0	89.3 ± 2.3
OGBench-Cube	NextWM ($q = 1.5$)	63.3 ± 11.0	64.7 ± 10.1	70.0 ± 12.0
	LeWM (released)	**70.0 ± 4.0**	**69.3 ± 12.1**	**70.7 ± 5.8**
Reacher	NextWM ($q = 1.5$)	**82.0 ± 5.3**	80.0 ± 2.0	**84.0 ± 7.2**
	LeWM (released)	**82.0 ± 6.0**	**82.7 ± 4.2**	78.0 ± 6.0
Two-Room	NextWM ($q = 1.5$)	78.0 ± 5.3	81.3 ± 3.1	83.3 ± 3.1
	LeWM (released)	**83.3 ± 6.4**	**85.3 ± 3.1**	**85.3 ± 3.1**

Figure 7 shows that increasing the budget does not produce a uniform improvement or a stable ordering between checkpoints. NextWM rises from 84.0% to 95.3% on Push-T and from 63.3% to 70.0% on Cube between the low and high settings. On Reacher, the methods tie at the low budget, LeWM leads at the standard budget, and NextWM leads by 6.0 percentage points at the high budget. LeWM retains the higher mean on Two-Room at all three settings. These fixed-checkpoint results show that the comparison depends on planner compute; three evaluation seeds do not establish training-level reproducibility or a monotonic compute benefit. Total evaluation wall time is also recorded with hostname, GPU model, solver configuration and checkpoint hash. This includes environment interaction and observation processing, so it is not reported as pure planner latency. The two methods run on the same node within each task/seed comparison, and their order is alternated across seeds.

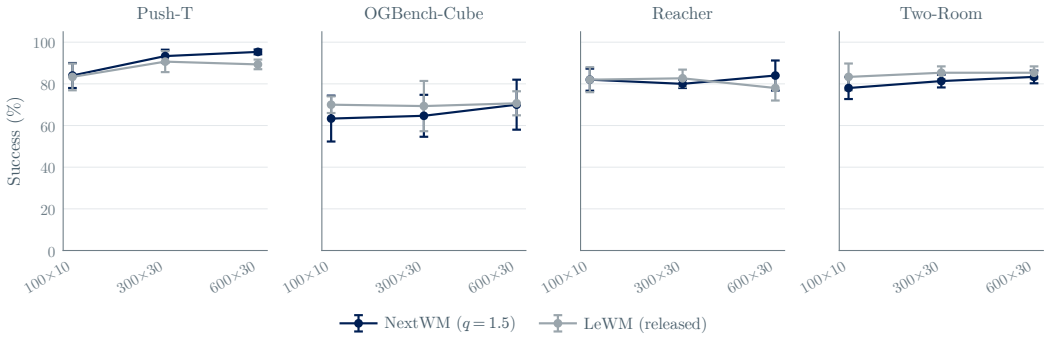


Figure 7: **Planning success versus CEM compute.** Low, standard and high budgets use 100×10, 300×30 and 600×30 candidates×iterations. Methods share evaluation pairs at every setting; intervals show sample SD across three evaluation seeds.

E TRAINING DIAGNOSTICS

The training traces test whether the observed checkpoint differences are accompanied by changes in prediction error, latent scale or dimensionality. Figure 8 summarizes logged validation aggregates, while Figures 9–12 retain the per-task trajectories. Table 6 reports the final values so that visual trends can be checked numerically. The $q = 1.5$ runs finish with larger coordinate variance and lower effective rank than the available unit-variance references on every task. These shared trends accompany different planning outcomes, so they do not by themselves identify a beneficial geometry.

Table 6: **Final validation diagnostics.** Logged aggregates at optimizer step 154,999 for one checkpoint per arm. No corresponding training trace is available for released LeWM. Empirical kurtosis is a finite-sample statistic; the $q = 1.5$ reference law has no finite fourth moment.

Environment	Model	q	Mean var.	Min var.	Eff. rank	Excess kurt.
Push-T	NextWM (unit var.)	1.2500	0.923	0.495	103.2	0.44
	NextWM	1.5000	2.663	0.825	74.3	1.73
OGBench-Cube	NextWM (unit var.)	1.2500	0.920	0.567	98.4	0.38
	NextWM	1.5000	2.778	1.563	72.6	2.21
Reacher	NextWM (unit var.)	1.2500	0.905	0.543	77.0	0.54
	NextWM	1.5000	2.540	1.200	61.7	1.53
Two-Room	NextWM (unit var.)	1.2500	0.929	0.607	97.7	0.48
	NextWM	1.5000	2.658	1.408	68.0	1.63

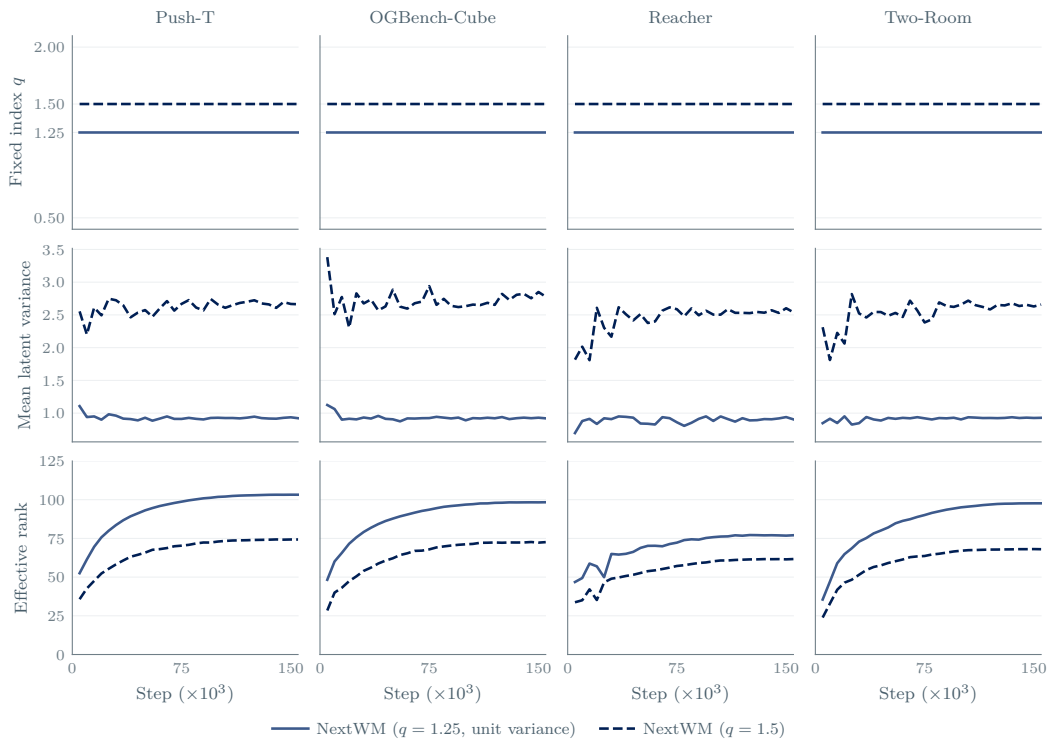


Figure 8: **Validation trajectories.** Logged epoch aggregates for completed fixed-index runs and available unit-variance references. The curves describe individual training runs, not uncertainty across training seeds.

Reading the per-task curves. Each panel follows one training run in optimizer-step order. Prediction loss describes fit in that run's latent metric; mean and minimum variance expose changes in scale and coordinate usage; effective rank summarizes dimensional concentration. Faint traces are logged values and the overlaid curves use a centered moving average. The fixed-q trace confirms the configured target index rather than an estimated tail parameter. Loss values across different latent scales should therefore be interpreted together with the geometry panels, not as a direct ranking of planning quality.

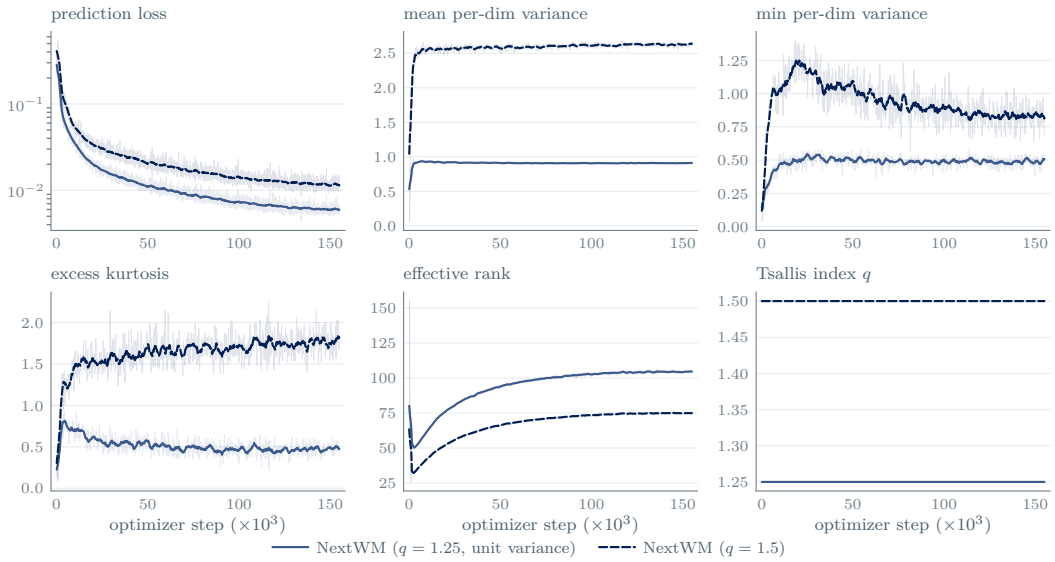


Figure 9: **Push-T training diagnostics.** Prediction loss, coordinate variance, empirical excess kurtosis, effective rank and fixed target index. Smoothing is for readability; one run per arm is shown.

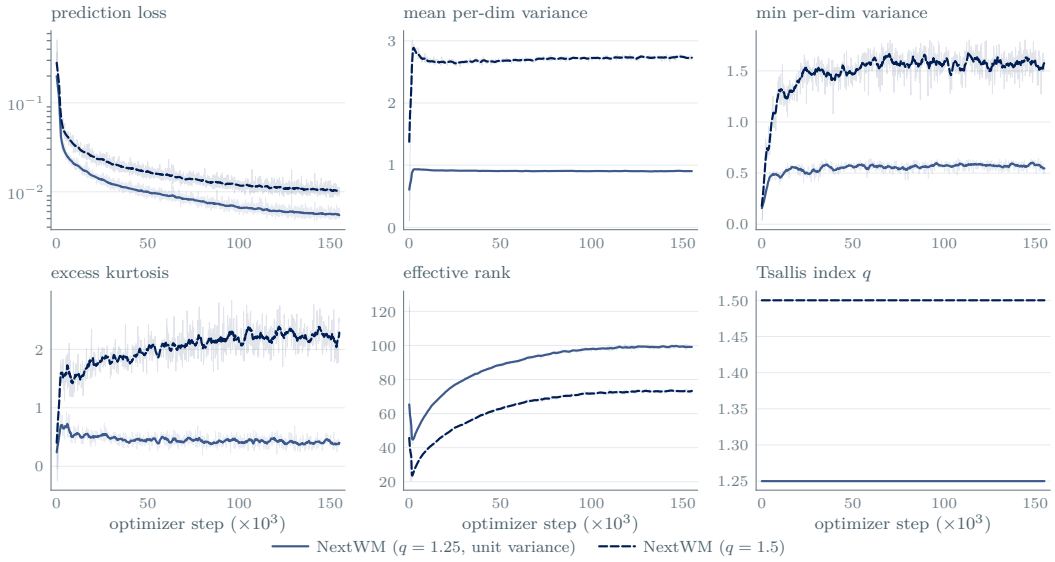


Figure 10: **OGBench-Cube training diagnostics.** Prediction loss, coordinate variance, empirical excess kurtosis, effective rank and fixed target index. Smoothing is for readability; one run per arm is shown.

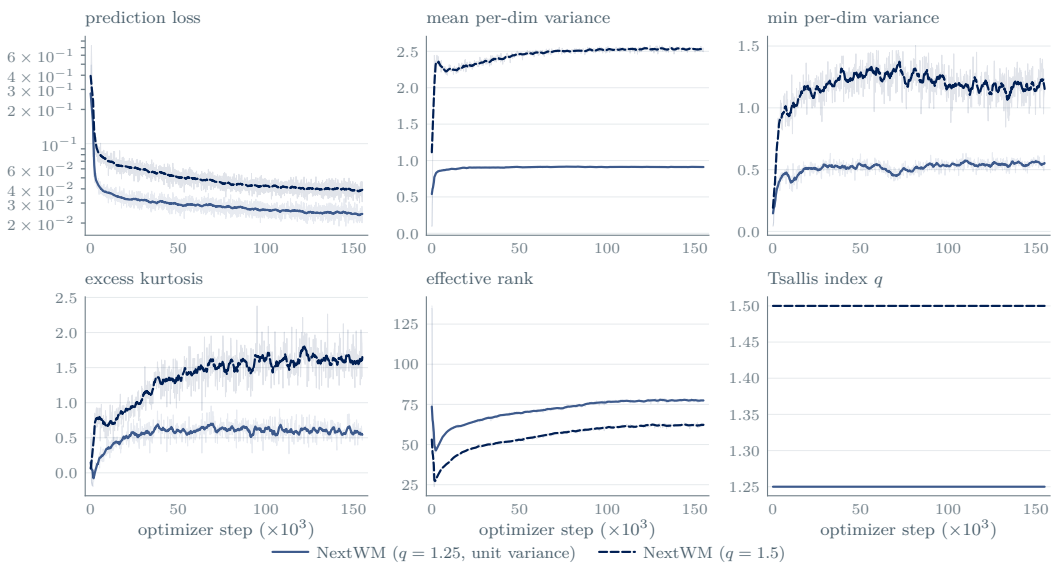


Figure 11: **Reacher training diagnostics.** Prediction loss, coordinate variance, empirical excess kurtosis, effective rank and fixed target index. Smoothing is for readability; one run per arm is shown.

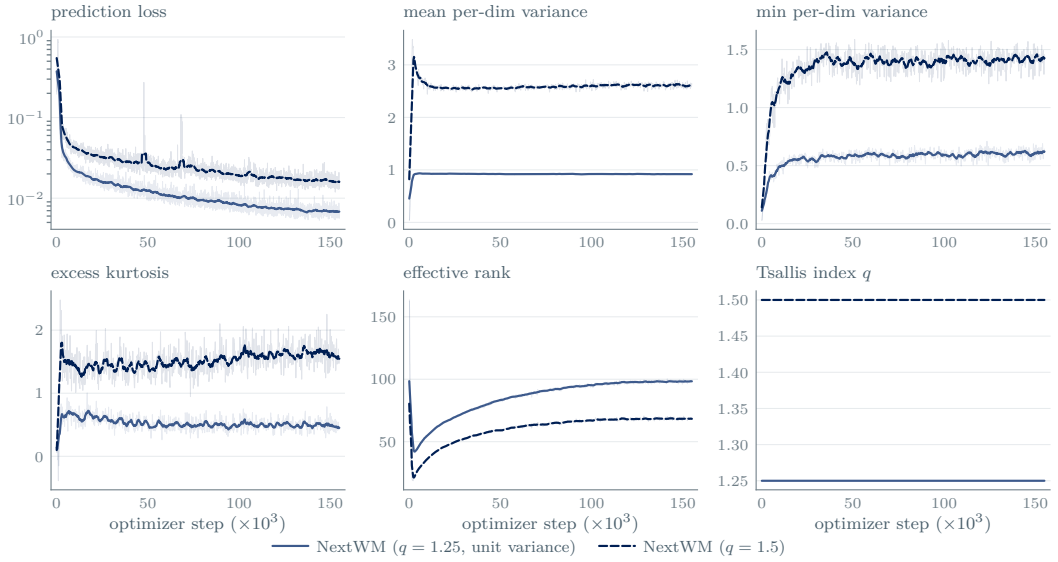


Figure 12: **Two-Room training diagnostics.** Prediction loss, coordinate variance, empirical excess kurtosis, effective rank and fixed target index. Smoothing is for readability; one run per arm is shown.

F TARGET DISTRIBUTIONS AND REFERENCE IDENTITIES

This section records the distribution identities needed to implement QCFReg and interpret the synthetic probes. They describe the target law, not a guarantee about learned geometry or planning.

Student-t parameterization and moments. For $1 < q < 3$ and $\beta > 0$,

$$p_q(x) = Z_q(\beta)^{-1}[1 + (q-1)\beta x^2]^{-1/(q-1)}, \quad Z_q(\beta) = \sqrt{\frac{\pi}{\beta(q-1)}}\frac{\Gamma(1/(q-1) - 1/2)}{\Gamma(1/(q-1))}. \tag{10}$$

Matching this density to a Student-t gives

$$\nu = \frac{3-q}{q-1}, \qquad s^2 = \frac{1}{\beta(3-q)}, \qquad \mathrm{Var}(X) = \frac{1}{\beta(5-3q)} \quad (q < 5/3). \tag{11}$$

Moments of order k exist exactly when $k < \nu$. Variance requires $q < 5/3$, while the fourth moment requires $q < 7/5$. Thus the principal $q = 1.5$ experiment has finite variance but no finite kurtosis. Choosing $\beta = 1/(5 - 3q)$ gives the unit-variance convention used in the earlier reference and synthetic probes.

Sampling and characteristic function. With independent $G \sim \mathcal{N}(0, I_d)$ and $W \sim \chi^2_\nu$, the vector $Z = sG/\sqrt{W/\nu}$ is elliptical Student-t; every unit projection has the same scalar t_ν law. This shared-scale sampler, rather than independent Student coordinates, is used in the probes. Conditioning on W yields $\varphi(t) = \mathbb{E}_W \exp[-t^2 s^2 \nu/(2W)]$ and the Bessel expression in equation 5. At unit variance, $s^2 = (\nu - 2)/\nu$ and $\sqrt{\nu}s = \sqrt{\nu - 2}$. As $q \to 1$, the CF converges to the Gaussian reference. Fixed targets are evaluated once with stabilized Bessel arithmetic, so no density fitting is performed during training.

F.1 COMPARISON WITH GENERALIZED-GAUSSIAN TARGETS

The symmetric reference $\mathrm{GG}_p(x) \propto \exp(-|x|^p)$ has stretched-exponential tails, whereas the q-Gaussian has power-law tails. LpWM also uses a rectifying link and different optimization choices, so this scalar comparison does not equate complete methods.

Proposition 1 (Finite-kurtosis correspondence)**.** *For every $\kappa \geq 0$, the unique $q \in [1, 7/5)$ satisfying $6(q-1)/(7-5q) = \kappa$ is*

$$Q(\kappa) = \frac{7\kappa + 6}{5\kappa + 6}. \tag{12}$$

The map is increasing, with $Q(0) = 1$, $Q(3) = 9/7$ and $Q(\kappa) \to 7/5$ as $\kappa \to \infty$.

Proof. Rearranging the matching equation gives $(5\kappa + 6)q = 7\kappa + 6$. Its positive denominator gives uniqueness, while $Q - 1 = 2\kappa/(5\kappa + 6)$ and $7/5 - Q = 12/[5(5\kappa + 6)]$ establish the range. Differentiation gives $Q' = 12/(5\kappa + 6)^2 > 0$; the stated values and limit follow directly. □

For GG_p, insert $\kappa_p = \Gamma(5/p)\Gamma(1/p)/\Gamma(3/p)^2 - 3$ whenever $\kappa_p \geq 0$. There is no finite-kurtosis match in this family for negative excess kurtosis. This elementary correspondence does not apply to $q = 1.5$. The companion `NextWM.lean` verifies the algebraic correspondence and selected limits; deriving the moment formulas from probability densities is outside its scope.

Density derivatives are not QCFReg gradients. The reference energies and derivatives are

$$E_p(x) = |x|^p, \quad E_p'(x) = p|x|^{p-1}\,\mathrm{sgn}(x), \tag{13}$$

$$E_q(x) = \frac{\log[1 + (q-1)\beta x^2]}{q-1}, \quad E_q'(x) = \frac{2\beta x}{1 + (q-1)\beta x^2}. \tag{14}$$

The latter derivative is bounded and decays in the tails, but neither expression is the batch-level training gradient in equation 9. Matching fourth moments therefore does not match tail decay, density penalties or the learning signal.

Numerical comparison. Table 7 and Figure 13 compare kurtosis matching with a KL-optimal Student-t fit. The latter minimizes $\mathrm{KL}(\mathrm{GG}_p \| t_{\nu,s})$ for a unit-variance GG_p, using adaptive quadrature and Nelder–Mead over $(\log \nu, \log s)$, capped at $\nu = 10^4$. Both criteria recover $q = 1$ at the Gaussian endpoint. For the Laplace reference ($p = 1$), kurtosis gives $q \approx 1.286$, whereas KL fitting gives $q \approx 1.467$ with divergence 0.009 nats. Their disagreement shows that a single moment does not characterize the whole density. The plotted ansatz $q = 1 + 2/(p + 1)$ fails even at the Gaussian endpoint and is included only as a counterexample.

Table 7: **Reference-family correspondences.** Generalized-Gaussian excess kurtosis, its matching q, and the numerically optimized Student-t fit with KL divergence.

p	κ_p	q (kurtosis match)	q (KL-best fit)	KL [nats]
0.5	22.20	1.379	1.891	0.0394
1.0	3.00	1.286	1.467	0.0093
1.5	0.76	1.155	1.195	0.0015
2.0	0.00	1.000	1.000	0.0000

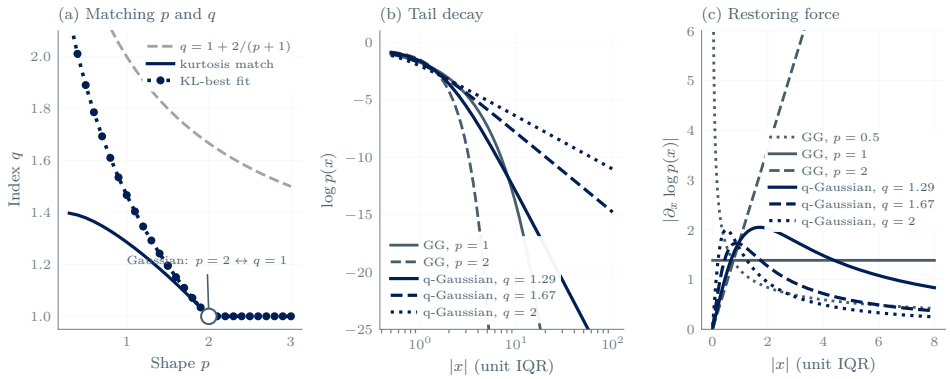


Figure 13: **Reference distributions, not learned latent densities.** (a) Kurtosis and KL correspondences. (b) Densities at unit interquartile range distinguish stretched-exponential from power-law tails. (c) Scalar negative-log-density derivatives; these are not the gradients of QCFReg.

G Recorded Episode Visualizations

These panels show the first two successful seed-42 recordings in dataset-row order for each task, using the fixed $q = 1.5$ checkpoint and historical 50/25 budget/goal offset. Columns follow actual observations from start to rollout frames and the dataset goal; terminated episodes are held at their last frame. This reproducible selection includes easy starts and illustrates behavior, not success frequency or superiority over LeWM. Aggregate results retain failures. Success follows the recorded evaluation criterion rather than pixel equality with the goal image.

Push-T. The upper sequence shows a larger change in the T-block pose, while the lower sequence starts close to the displayed goal pose and primarily changes the pusher position. These distinct starts help explain why a successful episode need not involve the same amount of object motion.



Figure 14: **Successful Push-T episodes with NextWM ($q = 1.5$).** Top: pair 0 (dataset episode 1235, start step 36); bottom: pair 1 (dataset episode 1316, start step 129). Actual observations; completed episodes are held at their final frame.

OGBench-Cube. The red block and the gripper provide separate visual cues. Block displacement is small in these selected recordings, while gripper configurations differ across the rollout and goal. The sequence makes it possible to follow object placement separately from the arm pose.



Figure 15: **Successful OGBench-Cube episodes with NextWM ($q = 1.5$).** Top: pair 2 (dataset episode 859, start step 76); bottom: pair 3 (dataset episode 892, start step 85). Actual observations; completed episodes are held at their final frame.

Reacher. In the upper sequence the arm changes configuration before returning toward the goal pose; the lower sequence begins closer to its goal and requires a smaller adjustment. The intermediate frames show how different initial configurations lead to successful endpoint configurations.

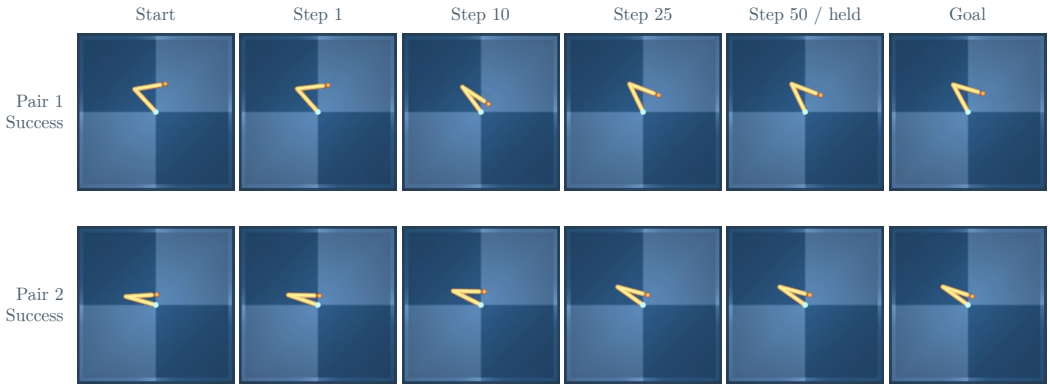


Figure 16: **Successful Reacher episodes with NextWM ($q = 1.5$).** Top: pair 1 (dataset episode 679, start step 34); bottom: pair 2 (dataset episode 859, start step 76). Actual observations; completed episodes are held at their final frame.

Two-Room. The upper sequence moves from the wall opening into the right room, whereas the lower sequence remains near the left side of the opening. Together they illustrate doorway crossing and local positioning near an obstacle.

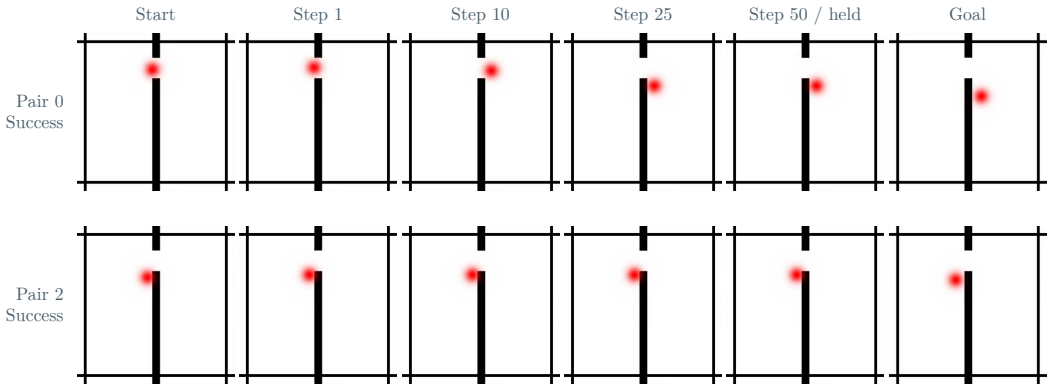


Figure 17: **Successful Two-Room episodes with NextWM ($q = 1.5$).** Top: pair 0 (dataset episode 644, start step 42); bottom: pair 2 (dataset episode 872, start step 25). Actual observations; completed episodes are held at their final frame.