WHAT MUST QUANTIZATION PRESERVE IN LATENT WORLD MODELS?

Anonymous authors
Paper under double-blind review

ABSTRACT

What must quantization preserve in latent world models to retain their planning ability? Existing work has primarily characterized sensitivity to precision configuration. To our knowledge, this is the first study to investigate latent world-model quantization from the perspective of representation preservation and calibration-target design. We analyze how quantization-induced representation errors propagate to goal-distance estimates and candidate rankings, and argue that calibration should preserve reusable latent representations rather than merely match scalar planning costs. We derive a bound relating terminal relative-representation error to the second moments of local encoder and predictor reconstruction errors, providing a theoretical basis for Module-MSE. Across LeWM, DINO-WM, and PLDM on four control tasks under symmetric per-channel W4 quantization, Module-MSE reaches 79.33% average success compared with 79.83% for full precision, while achieving the strongest overall planning-decision fidelity. It also maintains strong performance across the tested learning rates and calibration-set sizes. These results show that severe degradation under naive W4 quantization is not inevitable, establish representation preservation as a practical calibration principle, and support Module-MSE as a default choice for latent world-model quantization.

1 INTRODUCTION

World models learn action-conditioned dynamics and enable planning within a learned predictive model (Ha & Schmidhuber, 2018). Recent latent world models replace pixel-space prediction with compact representations and optimize action sequences for goal-conditioned tasks (Zhou et al., 2024; Sobal et al., 2025; Maes et al., 2026b). By decoupling learned dynamics from downstream goals, they support task reuse and zero-shot planning.

This flexibility, however, comes with substantial inference cost, as planning repeatedly rolls out many candidate action sequences over multiple prediction steps and search iterations. Post-training quantization (PTQ) offers a practical way to reduce model storage, memory traffic, and low-precision inference cost without full retraining. Quantizing a latent world model, however, differs from quantizing a one-shot feed-forward model. Predicted latent representations are reused by subsequent prediction steps and compared with goal representations, while quantization-induced score perturbations can reorder candidates, alter the CEM elite set, and change subsequent search distributions. Therefore, local weight distortion or isolated tensor reconstruction error does not directly determine whether planning behavior is preserved.

Prior work on latent world-model quantization has primarily characterized empirical sensitivity to precision configuration, including bit width, quantization granularity, planning horizon, and precision allocation across encoder and predictor modules (Fu et al., 2026; Ranganath et al., 2026). These studies show that low-bit planning can be highly sensitive to both precision and where that precision is allocated, but leave open a more fundamental question: *what information should quantization preserve during calibration?* To our knowledge, this is the first study to investigate latent world-model quantization from the perspective of representation preservation and calibration-target design.

We formulate this question as a representation-preservation problem. A natural target is the final scalar planning cost, but this cost is only a compressed readout of the underlying latent trajectory, and different latent predictions can produce the same cost. We therefore argue that quantization should preserve reusable latent representations at the points where errors are introduced, before

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.

they are reused by subsequent prediction and goal comparison. Across LeWM, DINO-WM, and PLDM on four control tasks, we compare cost-level, terminal-representation-level, and module-level calibration targets. We further derive a bound relating terminal relative-representation error to the second moments of module-level reconstruction errors, providing a theoretical basis for Module-MSE.

Our contributions are as follows:

1. We show that severe degradation under naive W4 quantization is not an inevitable consequence of 4-bit precision. With appropriate calibration, symmetric per-channel W4 retains near-full-precision planning performance across three latent world models and four control tasks, with Module-MSE reaching 79.33% average success compared with 79.83% for full precision.
2. We analyze how quantization-induced errors propagate through the planning pipeline, from module outputs to terminal representations, candidate ranking, elite selection, and subsequent CEM updates. We further derive a multi-step bound linking local reconstruction errors to terminal representation error.
3. We establish representation preservation as a calibration principle for latent world-model quantization. Module-MSE achieves the strongest overall task success and planning-decision fidelity and remains strong across learning-rate and calibration-set-size variations, supporting it as a practical default choice for latent world-model quantization.

2 Background and Related Work

2.1 Latent World Models and Predictive Planning

Learned-dynamics planning evaluates action sequences with a predictive model. PETS combines probabilistic dynamics with sampling-based MPC and CEM, while PlaNet performs CEM planning in a compact latent state learned from pixels (Chua et al., 2018; Hafner et al., 2019). Joint-embedding predictive architectures (JEPAs) instead predict target representations; I-JEPA is a representative image formulation (Assran et al., 2023). We study three latent world models used for goal-conditioned planning: DINO-WM pairs a pretrained DINOv2 encoder with an action-conditioned patch predictor (Zhou et al., 2024); PLDM learns an encoder and latent dynamics from reward-free trajectories (Sobal et al., 2025); and LeWM trains a compact JEPA end to end from raw pixels (Maes et al., 2026b).

Despite these architectural differences, all three models admit the same goal-conditioned inference abstraction. Let $M \in \{F, Q\}$ denote the full-precision or quantized model. For current and goal observations o_0, o_g and candidate action sequence $\mathbf{a}^{(i)}$, inference is

$$\begin{aligned} z_0^M &= E_M(o_0), \quad z_g^M = E_M(o_g), \\ z_{t+1}^{M,(i)} &= P_M(h_t^{M,(i)}, a_t^{(i)}), \quad t = 0, \dots, T-1, \quad J_i^M = \left\| z_T^{M,(i)} - z_g^M \right\|_2^2. \end{aligned} \tag{2.1}$$

Here $h_t^{M,(i)}$ contains the initial encoding and previous predictions, and $z_T^{M,(i)}$ is the terminal latent. No new observation enters a candidate rollout; model-specific action encoding, flattening, and score normalization are absorbed into this notation.

At CEM iteration r, planner M samples N action sequences from a Gaussian proposal (μ_r^M, σ_r^M), scores them with Equation 2.1, and lets $\mathcal{I}_r^M$ denote the indices of the K lowest-cost elites used to update the proposal. Model-predictive control executes a prefix of the planned action sequence and then replans from a new observation; real observations therefore enter between planning queries, while the predictor recurs within each query (Bharadhwaj et al., 2020).

2.2 Post-Training Quantization

Post-training quantization maps floating-point weights or activations to discrete values after training, reducing storage and inference cost. Its parameters may be shared per tensor, channel, or weight group.

Round-to-Nearest (RTN) is the simplest uniform quantizer. For signed symmetric b-bit quantization of a weight group $\mathbf{w} = (w_1, \ldots, w_n)$, let $q_{\max} = 2^{b-1} - 1$. The scale, integer code, and dequantized weight are

$$s = \frac{\max_i |w_i|}{q_{\max}}, \qquad q_i = \mathrm{clip}\Big(\mathrm{round}\Big(\frac{w_i}{s}\Big), -q_{\max}, q_{\max}\Big), \quad \widehat{w}_i = s q_i. \tag{2.2}$$

Here q_i is the integer code and $\widehat{w}_i$ its dequantized value. RTN maps each weight to the nearest level without calibration data or optimization, making it a standard PTQ baseline.

Calibration-based PTQ optimizes quantization variables using a small unlabeled set. AdaRound learns weight rounding with a layer-wise local reconstruction loss, whereas BRECQ reconstructs neural blocks to retain cross-layer dependencies (Nagel et al., 2020; Li et al., 2021). For vision transformers, PTQ4ViT introduces twin uniform quantization and Hessian-guided scale selection, while RepQ-ViT reparameterizes calibration-time scales for efficient inference (Yuan et al., 2022; Li et al., 2023). Representative low-bit LLM methods include the weight-only GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2024), and AutoRound (Cheng et al., 2024), and the weight-and-activation SmoothQuant (Xiao et al., 2023) and OmniQuant (Shao et al., 2024). These methods optimize rounding, scales, or layer/block reconstruction for general vision models or LLMs.

Existing studies of latent world-model quantization primarily examine sensitivity to precision configuration. Fu et al. (2026) study bit width, quantization granularity, planning horizon, and encoder–predictor sensitivity in DINO-WM, while Ranganath et al. (2026) investigate uniform and mixed-bit precision allocation and show that low-bit planning can depend strongly on where precision is assigned across modules. These studies characterize how precision settings affect planning performance.

3 Quantization effects on latent world model planning

3.1 Sensitivity to W4 RTN quantization

We compare each original full-precision model (FP) with its W4 symmetric per-channel RTN counterpart.

Table 1: Closed-loop success rate (%) of FP / W4 RTN. Each task contains 200 test episodes; Mean equally averages the four tasks.

Model	PushT	TwoRooms	Reacher	Cube	Mean
LeWM	85.0 / 79.5	85.0 / 83.0	95.5 / 96.5	65.0 / 68.0	82.63 / 81.75
DINO-WM	76.5 / 7.0	99.5 / 67.5	88.5 / 25.0	70.0 / 53.0	83.63 / 38.13
PLDM	48.0 / 36.0	94.0 / 93.5	96.5 / 92.5	54.5 / 57.5	73.25 / 69.88

Table 1 reveals clear model and task differences. DINO-WM is highly sensitive to W4 RTN on several tasks, especially PushT and Reacher. LeWM changes little overall, while PLDM varies by task. A few settings match or slightly exceed FP success. Overall, the effect of W4 RTN is strongly model- and task-dependent and severe in some settings.

The same W4 RTN configuration produces markedly different outcomes across models and tasks, so bit width alone cannot explain planning performance. We next follow the planning computation to analyze how quantization error propagates from latent representations to candidate scores and action selection.

3.2 From latent perturbations to action selection

We compare F and Q on the same planning query and raw candidate actions. Using the inference notation from Section 2.1, define the current, goal, and predicted latent differences for candidate i

as

$$\begin{aligned} \delta_0 &= z_0^F - z_0^Q, \quad \delta_g = z_g^F - z_g^Q, \\ \delta_{t+1}^{(i)} &= z_{t+1}^{F,(i)} - z_{t+1}^{Q,(i)} = P_F(h_t^{F,(i)}, a_t^{(i)}) - P_Q(h_t^{Q,(i)}, a_t^{(i)}). \end{aligned} \tag{3.1}$$

The two models use their own action encoders and latent histories. The first prediction reflects both δ_0 and predictor-side quantization; later calls also receive the accumulated history difference. Different action sequences induce different propagation paths and terminal perturbations $\delta_T^{(i)}$.

Using the planning cost J_i^M defined in Equation 2.1, let the score error be

$$\epsilon_i = J_i^F - J_i^Q. \tag{3.2}$$

Endpoint and goal perturbations can induce different score errors across candidates; quantization therefore does not simply shift all candidate costs by the same amount. Suppose FP prefers candidate i to j, and define its margin as $m_{ij} = J_j^F - J_i^F > 0$. Equation 3.2 gives

$$\begin{aligned} J_i^Q - J_j^Q &= -m_{ij} + (\epsilon_j - \epsilon_i), \\ J_i^Q > J_j^Q &\iff \epsilon_j - \epsilon_i > m_{ij}. \end{aligned} \tag{3.3}$$

Planning is therefore sensitive to differential rather than absolute score errors: a common offset cancels, while an error difference larger than the FP margin reverses the candidate order. If an inversion crosses the elite boundary, $\mathcal{I}_r^F \neq \mathcal{I}_r^Q$. Refitting CEM on different elites yields different proposal parameters $(\mu_{r+1}^M, \sigma_{r+1}^M)$, so even shared standard-normal samples generate different candidates. Repeated rollout and refitting can therefore amplify an initial ranking difference into different final plans and executed actions.

Thus, a latent perturbation need not cause a large absolute cost error to affect planning. It only needs to change relative scores enough to cross a candidate margin, especially at the elite boundary. The planner reads scalar costs, but the perturbations originate in representations reused for multi-step prediction and goal comparison.

4 Choosing quantization calibration targets

4.1 Three supervision locations

Section 3.2 showed how representation errors can change candidate selection. A key design choice is where to reconstruct the FP computation during calibration. Using the inference notation in Equation 2.1 and omitting the candidate index, define the terminal relative representation

$$r^M = z_T^M - z_g^M, \qquad M \in \{F, Q\}. \tag{4.1}$$

Its squared norm is the planning cost J^M defined in Equation 2.1. The possible supervision locations follow the computation

$$\text{encoder/predictor outputs} \longrightarrow r^M \longrightarrow J^M \longrightarrow \text{candidate ranking and selection.}$$

The first approach directly reconstructs the final cost; the second reconstructs the relative representation that produces that cost:

$$\mathcal{L}_{\text{Cost}} = \mathbb{E}[(J^F - J^Q)^2], \qquad \mathcal{L}_{\text{Rel}} = \mathbb{E}\left[\frac{1}{d}\|r^F - r^Q\|_2^2\right], \tag{4.2}$$

where d denotes the dimensionality of the corresponding flattened representation. Both losses allow the quantized model to roll out recursively using its own predicted history and compare only the terminal result.

The third approach directly reconstructs encoder outputs and the output of each predictor call. For $p \in \{1, 2\}$, define

$$\begin{aligned} \mathcal{L}_{E,p} &= \mathbb{E}_o\left[\frac{1}{d}\sum_{k=1}^{d} |[E_F(o) - E_Q(o)]_k|^p\right], \\ \mathcal{L}_{P,p} &= \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\frac{1}{d}\sum_{k=1}^{d} \left|[P_F(h_t^F, a_t) - P_Q(h_t^F, a_t)]_k\right|^p\right]. \end{aligned} \tag{4.3}$$

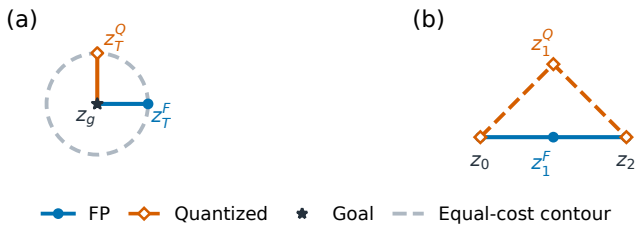


Figure 1: Degrees of freedom left by terminal supervision. (a) Distinct predictions lie on the same equal-cost contour around the goal. (b) At a fixed two-step horizon, two 2D trajectories share an endpoint and terminal relative vector but differ at the first prediction. Subscripts denote prediction steps.

We use $p = 1$ for Module-L1 and $p = 2$ for Module-MSE, applying $\mathcal{L}_{E,p}$ to encoder calibration and $\mathcal{L}_{P,p}$ to predictor calibration. Encoder reconstruction uses the same observation. Predictor reconstruction uses the same FP history and raw action, so $\mathcal{L}_{P,p}$ isolates the new error introduced by one predictor call rather than recounting earlier rollout drift.

4.2 Information limitations of terminal supervision

Consider scalar cost first. Let both models share the goal representation $z_g = (0, 0)$, with FP terminal prediction $z_T^F = (1, 0)$ and quantized terminal prediction $z_T^Q = (0, 1)$. Both predictions have squared distance 1 from the current goal, so Cost-MSE is zero for this example. If the goal representation is instead $(1, 0)$, their costs become 0 and 2. Figure 1(a) illustrates the equal-cost ambiguity for the current goal.

More generally, define the terminal relative-representation error as $e = r^F - r^Q$. Exact expansion gives

$$J^F - J^Q = 2(r^F)^\top e - \|e\|_2^2. \tag{4.4}$$

Even when $e \neq 0$, the right-hand side of Equation 4.4 can be zero. Equal distance to a goal therefore does not uniquely determine the terminal relative representation. Multiple goals in a finite training set add constraints, but direct cost reconstruction does not itself require complete preservation of each predicted representation.

Relative-MSE removes the ambiguity of equal radius but different position, yet it still compares only rollout endpoints. As a simple counterexample, consider two trajectories over a fixed horizon $T = 2$ in two dimensions. Let the FP trajectory be $(0, 0) \to (1, 0) \to (2, 0)$, the quantized trajectory $(0, 0) \to (1, 1) \to (2, 0)$, and the goal $(3, 0)$. The trajectories share the same endpoint and terminal relative vector $(-1, 0)$, so Relative-MSE is zero, even though their first predictions differ. Figure 1(b) labels the states z_0, z_1, z_2.

The two terminal losses preserve different information. Cost-MSE constrains only the distance to the goal and does not uniquely determine the terminal relative representation. Relative-MSE preserves the full terminal relative vector but allows errors at different steps to cancel at the endpoint. Neither directly constrains the encoder output or the new output introduced by each predictor call. Module-level losses supervise the locations at which these errors arise, providing a more direct route to controlling multi-step rollout error.

4.3 Module reconstruction and multi-step error propagation

The previous subsection showed that terminal comparison cannot distinguish intermediate errors. A natural alternative is to reconstruct encoder and predictor outputs where the errors are introduced. We now relate these local errors to the terminal error of a multi-step rollout. For clarity, we first present the single-state case $h_t = z_t$; the corresponding history-input formulation is given in Appendix A.1. Let $\delta_t = z_t^F - z_t^Q$ be the difference between the trajectories at step t. Module calibration compares the predictors at FP state z_t^F. Adding and subtracting $P_Q(z_t^F, a_t)$ in δ_{t+1} yields the exact

decomposition

$$\delta_{t+1} = \underbrace{P_F(z_t^F, a_t) - P_Q(z_t^F, a_t)}_{\eta_t\text{: directly controlled one-step error}} + \underbrace{P_Q(z_t^F, a_t) - P_Q(z_t^Q, a_t)}_{\text{propagated existing state error}}. \tag{4.5}$$

The first term η_t is the output difference when both predictors receive the same FP state and action; this is exactly the error optimized by predictor reconstruction. The second term arises because the inputs have already diverged. If L_Q is the maximum local gain of the quantized predictor with respect to input perturbations over the relevant rollout region, the norm of the second term is at most $L_Q\|\delta_t\|_2$. This condition can be interpreted through the largest singular value of the input Jacobian; Appendix A.2 gives the details. Hence

$$\begin{aligned} \|\delta_{t+1}\|_2 &\le L_Q\|\delta_t\|_2 + \|\eta_t\|_2, \\ \|\delta_T\|_2 &\le L_Q^T\|\delta_0\|_2 + \sum_{t=0}^{T-1} L_Q^{T-1-t}\|\eta_t\|_2. \end{aligned} \tag{4.6}$$

This expansion shows how each error source enters the endpoint. The goal encoding error is $\delta_g = z_g^F - z_g^Q$ from Equation 3.1. Since $e = r^F - r^Q = \delta_T - \delta_g$,

$$\|e\|_2 \le L_Q^T\|\delta_0\|_2 + \|\delta_g\|_2 + \sum_{t=0}^{T-1} L_Q^{T-1-t}\|\eta_t\|_2. \tag{4.7}$$

Equation 4.7 reduces the upper bound on terminal relative error to three module errors: initial-state encoding error $\|\delta_0\|_2$, goal encoding error $\|\delta_g\|_2$, and predictor reconstruction error $\|\eta_t\|_2$ at step t. Encoder reconstruction constrains both the initial state and goal, while predictor reconstruction constrains every prediction step. Reducing these errors tightens the upper bound on $\|e\|_2$.

Both sides of Equation 4.7 are nonnegative. Under the finite-gain condition above and the encoder-coverage condition detailed in Appendix A.2, squaring the inequality, applying Cauchy–Schwarz to the cross terms, and taking expectation over the calibration distribution gives

$$\mathcal{L}_{\mathrm{Rel}} \le C_E\mathcal{L}_{E,2} + C_P\mathcal{L}_{P,2}. \tag{4.8}$$

The fixed constants C_E, C_P collect propagation coefficients; the complete derivation is in Appendix A.2. Equation 4.8 bounds terminal relative-representation MSE in terms of the Module-MSE components, up to propagation constants. Equation 4.4 further connects this terminal relative-representation error to planning-cost error.

4.4 MSE AND L1 UNDER A SECOND-MOMENT ERROR BOUND

The right-hand side of Equation 4.8 contains second moments of module error. To compare MSE and L1 at the same supervision locations, let $v \in \mathbb{R}^d$ denote any of the error vectors δ_0, δ_g, or η_t in equation 4.7. Following equation 4.3, define the single-sample loss

$$\ell_p(v) = \frac{1}{d}\sum_{k=1}^{d} |v_k|^p, \qquad p \in \{1, 2\}. \tag{4.9}$$

The bound in Section 4 contains second-moment terms of the form $\mathbb{E}\|v\|_2^2 = d\,\mathbb{E}\ell_2(v)$. Module-MSE therefore enters Equation 4.8 directly. Module-L1 instead controls $\mathbb{E}\ell_1(v)$. Although every sample satisfies

$$\|v\|_2^2 \le \|v\|_1^2, \tag{4.10}$$

this does not allow a first moment $\mathbb{E}\|v\|_1$ to control the second moment required by the bound. In one dimension, for any $\lambda \ge 1$, let

$$v = \begin{cases} \lambda, & \text{with probability } 1/\lambda, \\ 0, & \text{with probability } 1 - 1/\lambda, \end{cases} \qquad \mathbb{E}|v| = 1, \quad \mathbb{E}v^2 = \lambda. \tag{4.11}$$

The mean L1 remains fixed while rare large errors make the squared error arbitrarily large. If the coordinate errors additionally satisfy $\|v\|_\infty \le b$, L1 can control the same second-moment risk indirectly:

$$\mathbb{E}\ell_2(v) \le b\,\mathbb{E}\ell_1(v). \tag{4.12}$$

MSE requires no such amplitude bound and directly penalizes large local errors quadratically, consistent with the squared-error terms in Equation 4.8.

Together with Section 3, controlling terminal relative error can reduce candidate-cost perturbations and the risk of margin-crossing ranking inversions. Appendix A.3 gives a formal sufficient condition. In summary, module-level supervision constrains the initial encoding error and the new prediction error injected at each step, while MSE directly controls the second moments in the multi-step bound. Together, these results motivate Module-MSE as a natural default for module-level calibration.

5 EXPERIMENTS

5.1 OVERALL COMPARISON OF QUANTIZATION METHODS AND CALIBRATION TARGETS

The main experiment evaluates LeWM, DINO-WM, and PLDM on PushT, TwoRooms, Reacher, and Cube, yielding twelve model–task settings. We compare FP, W4 RTN, GPTQ, and four learnable-rounding objectives: Module-MSE, Module-L1, Relative-MSE, and Cost-MSE. The four objectives are implemented with AutoRound (Cheng et al., 2024), a lightweight training-based PTQ method. Quantized models apply symmetric per-channel W4 quantization to all linear weights, while activations and the remaining parameters stay in floating point. Each condition is evaluated on 200 closed-loop episodes. Appendix B.1 provides the complete experimental settings and evaluation protocols.

We report both closed-loop success and planning-decision fidelity to the FP model. Decision fidelity is measured using candidate inversion rate (the fraction of candidate pairs whose FP ordering is reversed by quantization), elite overlap (the fraction of FP elites retained by the quantized model), and selection regret (the additional FP cost of the candidate selected by the quantized model). Appendix B.1 provides the formal definitions and evaluation protocol. FP is not a perfect task solver, so task success and FP fidelity need not agree.

Table 2: Closed-loop success rates across twelve model–task settings. Each task-level cell contains 200 episodes. Averages equally weight settings, and bold denotes the best quantized result.

Model	Task	FP	RTN	GPTQ	Module-MSE	Module-L1	Relative-MSE	Cost-MSE
LeWM	PushT	85.0%	79.5%	83.5%	**84.5%**	82.0%	80.5%	78.0%
LeWM	TwoRooms	85.0%	83.0%	82.5%	82.5%	84.0%	83.5%	**86.0%**
LeWM	Reacher	95.5%	**96.5%**	95.0%	95.5%	94.0%	93.0%	94.5%
LeWM	Cube	65.0%	**68.0%**	66.0%	65.0%	67.5%	67.0%	67.5%
LeWM	*Average*	82.63%	81.75%	81.75%	**81.88%**	**81.88%**	81.00%	81.50%
DINO-WM	PushT	76.5%	7.0%	61.0%	**76.0%**	71.5%	73.0%	22.0%
DINO-WM	TwoRooms	99.5%	67.5%	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
DINO-WM	Reacher	88.5%	25.0%	**85.5%**	84.0%	**85.5%**	82.0%	69.0%
DINO-WM	Cube	70.0%	53.0%	71.0%	71.5%	**73.0%**	68.5%	69.0%
DINO-WM	*Average*	83.63%	38.13%	79.38%	**82.88%**	82.50%	80.88%	65.00%
PLDM	PushT	48.0%	36.0%	**39.0%**	**39.0%**	36.5%	32.0%	30.0%
PLDM	TwoRooms	94.0%	93.5%	94.5%	95.0%	**96.0%**	94.5%	**96.0%**
PLDM	Reacher	96.5%	92.5%	95.5%	**99.5%**	97.0%	96.0%	96.0%
PLDM	Cube	54.5%	57.5%	58.5%	**59.5%**	58.0%	56.0%	58.0%
PLDM	*Average*	73.25%	69.88%	71.88%	**73.25%**	71.88%	69.63%	70.00%
Overall average		79.83%	63.25%	77.67%	**79.33%**	78.75%	77.17%	72.17%

Table 2 shows that GPTQ substantially restores the performance lost under RTN for DINO-WM, increasing its average success from 38.13% to 79.38%, while maintaining strong performance on LeWM and PLDM. Its competitive performance is also consistent with the usefulness of local output reconstruction objectives. Across all twelve settings, Module-MSE achieves the highest average success rate (79.33%), followed by Module-L1 (78.75%), GPTQ (77.67%), Relative-MSE (77.17%), and Cost-MSE (72.17%). Module-MSE also attains the highest mean elite overlap (89.44%), the

Table 3: Overall averages and first-place counts across twelve settings. Each cell reports the mean followed by the number of first-place settings (mean / wins). Success, overlap, and inversion are percentages; selected regret is averaged by within-setting rank. Ties count as a win for every tied method.

Method	Success ↑	Elite overlap ↑	Inversion rate ↓	Selected-regret rank ↓
RTN	63.25% / 2	68.63% / 0	14.77% / 0	5.92 / 0
GPTQ	77.67% / 3	87.94% / 3	5.52% / 4	3.17 / 1
Module-MSE	**79.33% / 6**	**89.44% / 6**	**5.00% / 5**	**1.42 / 7**
Module-L1	78.75% / 4	89.29% / 3	5.08% / 3	2.50 / 2
Relative-MSE	77.17% / 1	87.26% / 0	6.20% / 0	2.92 / 2
Cost-MSE	72.17% / 3	80.80% / 0	9.09% / 0	5.08 / 0

lowest mean inversion rate (5.00%), and the best mean selected-regret rank (1.42). Detailed per-setting values are reported in Appendix B.2.

Table 3 further shows that Module-MSE records the most first-place results for success, elite overlap, inversion rate, and selected regret, with 6, 6, 5, and 7 wins, respectively. Overall, module reconstruction performs consistently well, with Module-MSE as the strongest overall default and Module-L1 remaining competitive.

5.2 Effect of adjacent-path supervision

Relative-MSE and Cost-MSE supervise only the relationship between the rollout endpoint and the goal and do not explicitly constrain intermediate predictions. To test whether intermediate path information can improve these terminal-level objectives, we augment them with adjacent-path supervision. We evaluate this ablation on PushT for all three models using candidate trajectories with five prediction steps. Define $\Delta_t^M = z_{t+1}^M - z_t^M$ $(t = 0, \dots, 4)$ and $\Delta_g^M = z_5^M - z_g^M$, where $M \in \{F, Q\}$. Omitting fixed normalization scales, the adjacent-path objectives are

$$\begin{aligned} \mathcal{L}_{\text{Rel-Adj}} &= \frac{1}{6}\left[\sum_{t=0}^{4} \frac{\|\Delta_t^Q - \Delta_t^F\|_2^2}{d_z} + \frac{\|\Delta_g^Q - \Delta_g^F\|_2^2}{d_z}\right], \\ \mathcal{L}_{\text{Cost-Adj}} &= \frac{1}{6}\left[\sum_{t=0}^{4} \left(\|\Delta_t^Q\|_2^2 - \|\Delta_t^F\|_2^2\right)^2 + \left(\|\Delta_g^Q\|_2^2 - \|\Delta_g^F\|_2^2\right)^2\right]. \end{aligned} \tag{5.1}$$

The Relative branch matches each displacement vector; the Cost branch matches only the scalar squared distance of each segment.

Table 4: PushT results under adjacent-path supervision. Each cell reports success / elite overlap / inversion rate (%, ↑ / ↑ / ↓). Bold denotes the best quantized result within a model.

Method	LeWM	DINO-WM	PLDM
Module-MSE	**84.5 / 91.17 / 3.95**	**76.0 / 78.35 / 10.46**	**39.0 / 71.28 / 13.97**
Cost-MSE	78.0 / 87.60 / 5.31	22.0 / 33.53 / 32.26	30.0 / 64.13 / 17.76
Cost-MSE + Adj.	80.0 / 89.42 / 4.85	56.0 / 63.20 / 17.22	32.5 / 66.15 / 17.25
Relative-MSE	80.5 / 90.92 / 3.98	73.0 / 73.30 / 12.73	32.0 / 66.77 / 16.39
Relative-MSE + Adj.	82.5 / 90.68 / 4.04	70.5 / 74.45 / 12.31	32.0 / 67.58 / 16.02

Appendix B.3 provides the complete experimental settings and detailed results. As shown in Table 4, adjacent-path supervision consistently improves Cost-MSE across all three models, with the largest gains on DINO-WM, while its effect on Relative-MSE is smaller and less consistent across success and decision-fidelity metrics. These results are consistent with the analysis in Section 4: adding local segment constraints can alleviate some of the information loss of terminal-only scalar supervision. Relative-MSE already preserves the full terminal relative vector and shows smaller and less consistent gains from additional path supervision. Both adjacent-path variants remain below the Module-MSE baseline across all reported metrics.

5.3 SENSITIVITY TO LEARNING RATE AND CALIBRATION-SET SIZE

We evaluate all four calibration losses on PushT for the three models, sweeping the learning rate over $0.5\times$, $1\times$, and $2\times$ ($1\times = 1/384 \approx 2.6042 \times 10^{-3}$) and the calibration set over 1, 13, 32, and 128 trajectories. Appendix B.4 provides the complete configurations and results.

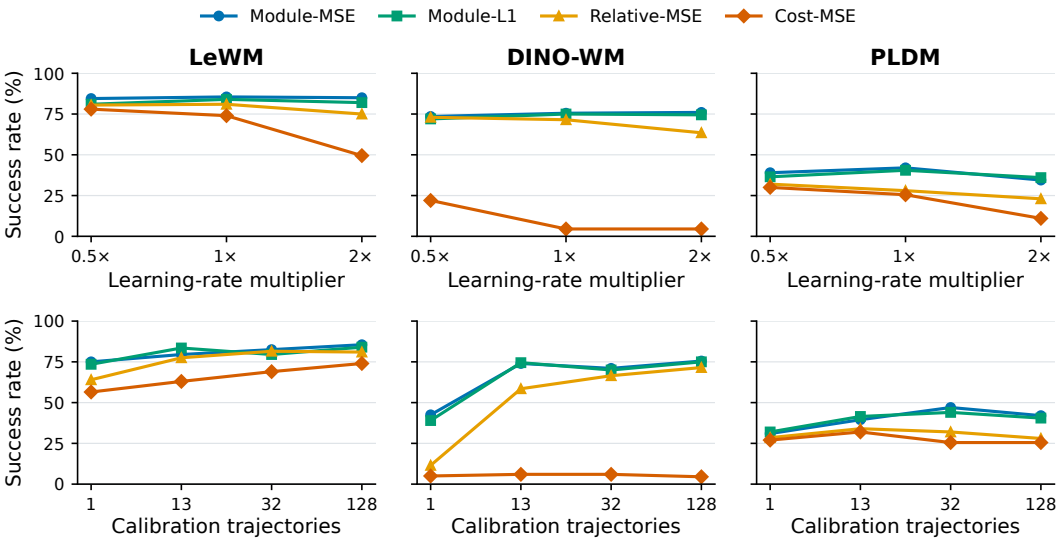


Figure 2: PushT success under learning-rate (top) and calibration-size (bottom) sweeps for three world models; each point contains 200 episodes.

Figure 2 summarizes success across both sensitivity sweeps. Across the learning-rate sweep, Module-MSE has the highest mean success for all three models, with ranges of 1.0, 2.5, and 7.5 percentage points for LeWM, DINO-WM, and PLDM. Module-L1 remains close, whereas Cost-MSE varies by 17.5–28.5 points. Module-MSE also maintains strong decision fidelity across the sweep.

Reducing the calibration set from 128 to 13 trajectories changes Module-MSE's three-model mean success from 67.67% to 64.33%. Even with one trajectory, the two module objectives retain the best success and decision fidelity; Relative-MSE degrades more sharply, while Cost-MSE remains weakest. Overall, Module-MSE maintains the strongest performance across the tested learning rates and calibration-set sizes, with Module-L1 close behind.

6 CONCLUSION

This work reframes latent world-model quantization as a representation-preservation problem. We show that matching final scalar costs alone can discard information needed for subsequent prediction and goal comparison, whereas module-level reconstruction directly preserves reusable latent representations where quantization errors are introduced. Our analysis provides a theoretical basis for Module-MSE by bounding terminal relative-representation error with module-level reconstruction errors. Across three latent world models and four control tasks, Module-MSE achieves near-full-precision W4 planning performance and the strongest overall planning-decision fidelity. These results establish representation preservation as a practical calibration principle and support Module-MSE as a default choice for latent world-model quantization. Future work should extend this analysis to activation quantization and low-precision hardware deployment.

AI USE STATEMENT

Generative AI tools were used to aid and polish the manuscript's writing and to assist with the retrieval and discovery of relevant literature. The authors independently verified all cited sources and reviewed all AI-assisted edits. The authors take responsibility for the final content of this work.

Reproducibility statement

The calibration objectives, metrics, and all reported results are specified in Section 5; the full experimental protocol is provided in Appendix B.1. The assumptions and complete derivations supporting the theoretical claims are provided in Appendix A.1–A.3.

References

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*, 2023.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Proceedings of the Conference on Learning for Dynamics and Control*, 2020.

Wenhua Cheng, Weiwei Zhang, Haihao Shen, Yiyang Cai, Xin He, Kaokao Lv, and Yi Liu. Optimize weight rounding via signed gradient descent for the quantization of LLMs. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Zhongqian Fu, Tianyi Zhao, Kai Han, Hang Zhou, Xinghao Chen, and Yunhe Wang. An empirical study of world model quantization. *arXiv preprint arXiv:2602.02110*, 2026.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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*, 2019.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021.

Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. RepQ-ViT: Scale reparameterization for post-training quantization of vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Lucas Maes, Quentin Le Lidec, Luiz Facury, Nassim Massaudi, Ayush Chaurasia, Francesco Capuano, Richard Gao, Taj Gillin, Dan Haramati, Damien Scieur, Yann LeCun, and Randall Balestriero. stable-worldmodel: A platform for reproducible world modeling research and evaluation, 2026a. URL `https://arxiv.org/abs/2605.21800`.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026b.

Markus Nagel, Rana Ali Amjad, Mart van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, 2020.

Suraj Ranganath, Anish Patnaik, and Vaishak Menon. Where bits matter in world model planning: A paired mixed-bit study for efficient spatial reasoning. *arXiv preprint arXiv:2602.11882*, 2026.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 2023.

Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *European Conference on Computer Vision*, 2022.

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.

A Additional theoretical analysis

A.1 Extension to multi-step history inputs

The main text presents the single-state case $h_t = z_t$. For models that require a multi-step history, let x_t^M be the complete history before the prediction at step t, let $P_M(x_t^M, a_t)$ produce the new latent, and let H update the history according to a model-specific fixed rule:

$$z_{t+1}^M = P_M(x_t^M, a_t), \qquad x_{t+1}^M = H(x_t^M, z_{t+1}^M). \tag{A.1}$$

H is the same for FP and the quantized model. Predicted values in the history come from previous rollout steps, not from additional observations. Let S_T extract the predicted endpoint from the final history, so $z_T^M = S_T x_T^M$. For a coordinate-selection operator, $\|S_T\|_2 \le 1$. The definition of z in the main text already includes the flattening and normalization required by the model's native score; hence

$$r^M = S_T x_T^M - z_g^M, \qquad J^M = \|r^M\|_2^2. \tag{A.2}$$

The terminal relative-representation loss is

$$\mathcal{L}_{\mathrm{Rel}} = \mathbb{E}\left[\frac{1}{d_z}\|r^F - r^Q\|_2^2\right]. \tag{A.3}$$

With history inputs, same-input predictor reconstruction remains

$$\mathcal{L}_{P,p} = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\frac{1}{d_z}\|P_F(x_t^F, a_t) - P_Q(x_t^F, a_t)\|_p^p\right], \quad p \in \{1, 2\}, \tag{A.4}$$

where d_z is the flattened dimension of one new prediction. For the shift-and-append history updates used by the studied models, the same-input update difference is nonzero only in the newly predicted slot. The single-state decomposition in the main text therefore extends directly to the history state.

A.2 How Module-MSE controls terminal error

Let $G_{M,t}(x, a) = H(x, P_M(x, a))$ denote the complete operation that produces a new latent and updates the history. Define the history error, goal encoding error, and same-input update error on the FP history as

$$\begin{aligned} &\delta_t = x_t^F - x_t^Q, && \delta_g = E_F(o_g) - E_Q(o_g), \\ &\eta_t = G_{F,t}(x_t^F, a_t) - G_{Q,t}(x_t^F, a_t). \end{aligned} \tag{A.5}$$

η_t is the error measured by the training loss on the FP history. Because H applies the same history shift in both models, $G_{F,t} - G_{Q,t}$ is nonzero only in the new prediction slot. Therefore

$$\sum_{t=0}^{T-1} \mathbb{E}\|\eta_t\|_2^2 = T d_z \, \mathcal{L}_{P,2}. \tag{A.6}$$

For a specified query distribution and finite prediction horizon, the bound uses two conditions:

1. **Finite propagation gain.** Over the relevant input region, $\|G_{Q,t}(x,a) - G_{Q,t}(y,a)\|_2 \le L_Q \|x - y\|_2$.
2. **Encoding coverage.** The reconstruction risk of the encoder controls the encoding errors of the initial history and goal.

The first condition can be understood through the predictor's local Jacobian. Let $J_{Q,t}(x,a) = \partial G_{Q,t}(x,a)/\partial x$. For any perturbation Δx, the matrix norm gives

$$\|J_{Q,t}(x,a)\Delta x\|_2 \le \sigma_{\max}(J_{Q,t}(x,a))\|\Delta x\|_2.$$

Bounding the largest singular value along the input path between x and y by L_Q yields the first condition. We consider only the input region visited by a fixed model over a finite prediction horizon and therefore use a finite local bound. We neither require $L_Q < 1$ nor claim global predictor stability.

The second condition is

$$\mathbb{E}\|\delta_0\|_2^2 + \mathbb{E}\|\delta_g\|_2^2 \le \rho_E d_z \, \mathcal{L}_{E,2}, \tag{A.7}$$

where ρ_E absorbs the initial history length and sampling weights in the encoder loss.

Proof. Add and subtract $G_{Q,t}(x_t^F, a_t)$ in δ_{t+1}:

$$\delta_{t+1} = \eta_t + G_{Q,t}(x_t^F, a_t) - G_{Q,t}(x_t^Q, a_t).$$

Applying the first condition recursively gives

$$\|\delta_T\|_2 \le L_Q^T \|\delta_0\|_2 + \sum_{t=0}^{T-1} L_Q^{T-1-t} \|\eta_t\|_2. \tag{A.8}$$

By the score definition in Appendix A.1, $e = r^F - r^Q = S_T \delta_T - \delta_g$. Together with $\|S_T\|_2 \le 1$,

$$\|e\|_2 \le L_Q^T \|\delta_0\|_2 + \|\delta_g\|_2 + \sum_{t=0}^{T-1} L_Q^{T-1-t} \|\eta_t\|_2. \tag{A.9}$$

Let $B_T = L_Q^{2T} + 1 + \sum_{k=0}^{T-1} L_Q^{2k}$. Applying the Cauchy–Schwarz inequality to the weighted sum in Equation A.9 gives, for each sample,

$$\left(L_Q^T \|\delta_0\|_2 + \|\delta_g\|_2 + \sum_{t=0}^{T-1} L_Q^{T-1-t} \|\eta_t\|_2 \right)^2 \le B_T \left(\|\delta_0\|_2^2 + \|\delta_g\|_2^2 + \sum_{t=0}^{T-1} \|\eta_t\|_2^2 \right).$$

Here B_T is the sum of squared coefficients on the left. Taking expectations and substituting equation A.9 yields

$$\mathbb{E}\|e\|_2^2 \le B_T \left(\mathbb{E}\|\delta_0\|_2^2 + \mathbb{E}\|\delta_g\|_2^2 + \sum_{t=0}^{T-1} \mathbb{E}\|\eta_t\|_2^2 \right). \tag{A.10}$$

Substituting Equations A.6 and A.7 and dividing by d_z gives

$$\mathcal{L}_{\mathrm{Rel}} \le \underbrace{B_T \rho_E}_{C_E} \mathcal{L}_{E,2} + \underbrace{B_T T}_{C_P} \mathcal{L}_{P,2}. \tag{A.11}$$

This is Equation 4.8 in the main text. The derivation does not assume independent, zero-mean, or monotonically increasing per-step errors. The triangle and Cauchy–Schwarz inequalities ignore potentially beneficial error cancellation, so the bound may be loose. The result concerns expected risk under the specified distribution. A finite calibration set only estimates this risk; held-out performance still depends on input coverage and generalization.

A.3 Representation error and candidate ordering

For candidate i, let $e_i = r_i^F - r_i^Q$. The reverse triangle inequality gives

$$\left|\sqrt{J_i^F} - \sqrt{J_i^Q}\right| = \left|\|r_i^F\|_2 - \|r_i^Q\|_2\right| \le \|e_i\|_2. \tag{A.12}$$

If FP prefers candidate i to candidate j, then

$$\sqrt{J_j^Q} - \sqrt{J_i^Q} \ge \sqrt{J_j^F} - \sqrt{J_i^F} - \|e_i\|_2 - \|e_j\|_2. \tag{A.13}$$

Therefore, if the FP distance margin satisfies

$$\sqrt{J_j^F} - \sqrt{J_i^F} > \|e_i\|_2 + \|e_j\|_2,$$

the right-hand side of equation A.13 is positive. Since the square root is strictly increasing, $J_i^Q < J_j^Q$. If this condition holds for every candidate pair that crosses the FP elite boundary, the elite set is also preserved. Tied candidates require a consistent tie-handling rule.

This is sufficient but not necessary. Small mean MSE does not guarantee small error for every candidate, but it can constrain the probability of some ordering mistakes. Let $m > 0$. For candidate pairs ordered by FP, an inversion with FP distance margin at least m requires $\|e_i\|_2 + \|e_j\|_2 \ge m$. Markov's inequality gives

$$\Pr\left(\text{inversion and } \sqrt{J_j^F} - \sqrt{J_i^F} \ge m\right) \le \frac{2\,\mathbb{E}[\|e_i\|_2^2 + \|e_j\|_2^2]}{m^2}. \tag{A.14}$$

The expectation and probability use the same candidate-pair distribution. This is an upper bound on the joint event, not a conditional inversion rate obtained by ignoring the conditioning distribution, and not a task-failure probability. It preserves the important distinction that greater fidelity to FP and higher task success are not equivalent.

B Experimental protocols and detailed results

B.1 Shared experimental protocol and metrics

Models, tasks, and planning. The main experiment covers LeWM, DINO-WM, and PLDM on PushT, TwoRooms, Reacher, and Cube, for twelve model–task settings. All LeWM models use the official release (Maes et al., 2026b), and the DINO-WM PushT model uses its official release (Zhou et al., 2024). The remaining models are trained using the default configurations of Stable World-Model (Maes et al., 2026a). The adjacent-path and sensitivity experiments use PushT with all three models. Each condition is evaluated by native closed-loop planning on 200 test episodes. Unless stated otherwise, calibration uses 128 training trajectories and 32 validation trajectories, with calibration and test episodes disjoint. The planner uses 300 candidates, 30 CEM iterations, and 30 elites. A rollout contains five prediction blocks, each corresponding to five physical actions; the system replans after 25 actions, and each episode has a budget of 50 actions.

Quantization and calibration. All quantized models apply symmetric per-channel W4 quantization to all linear weights with integer range $[-7, 7]$, while activations, convolutions, biases, normalization layers, and positional parameters remain in floating point. We compare seven conditions: FP, RTN, GPTQ, and four objectives trained in the same learnable-rounding framework—Module-MSE, Module-L1, Relative-MSE, and Cost-MSE. RTN uses no calibration data. GPTQ quantizes layers sequentially using activations from 128 training trajectories, with a block size of 128 and damping of 0.01. The four learnable-rounding methods use the same quantization parameterization and 384 optimization steps, but differ in their supervision targets and locations.

All AutoRound-based methods (Cheng et al., 2024) are trained at $0.5\times$, $1\times$, and $2\times$ learning rates. The corresponding initial learning rates are $1/768$, $1/384$, and $1/192$ (approximately 1.3021×10^{-3}, 2.6042×10^{-3}, and 5.2083×10^{-3}), respectively. Each learning rate is linearly decayed to zero over the 384 optimization steps. We choose the learning rate by the final-step candidate inversion rate on 32 validation states.

Decision-fidelity metrics. Each quantized condition is compared with FP on 200 test states, with both models scoring the same 300 candidates at every state. Let $\mathcal{P} = \{(i,j) : J_i^F < J_j^F\}$ contain candidate pairs with strict FP order, let $\mathcal{S}_F$ and $\mathcal{S}_Q$ be the K elites selected by FP and the quantized model, and let $i_F = \arg\min_i J_i^F$ and $i_Q = \arg\min_i J_i^Q$. The three fidelity metrics are

$$\begin{aligned} R_{\text{inv}} &= \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \left(\mathbf{1}[J_i^Q > J_j^Q] + \frac{1}{2}\mathbf{1}[J_i^Q = J_j^Q] \right), \\ O_{\text{elite}} &= \frac{|\mathcal{S}_F \cap \mathcal{S}_Q|}{K}, \\ R_{\text{sel}} &= J_{i_Q}^F - J_{i_F}^F. \end{aligned} \tag{B.1}$$

Candidate pairs tied under FP are excluded, and ties under the quantized model count as half an inversion. Higher elite overlap is better; the other two metrics are lower-is-better.

B.2 Overall comparison: setup and detailed results

The overall comparison evaluates the seven conditions described in Appendix B.1 in all twelve model–task settings. Closed-loop success uses 200 episodes per condition. Decision fidelity is evaluated on 200 test states per setting, with FP and every quantized model scoring the same 300 candidates at each state. Test episode identities and candidate banks are aligned within each setting.

Tables B.1–B.3 report the complete decision-fidelity results. Success rates are already given in Table 2. Overlap and inversion are percentages; selected regret is reported in the native cost units of each model–task setting and should therefore be compared only within a task. Bold indicates the best quantized result in each task and metric.

Table B.1: Detailed decision-fidelity results for LeWM.

Task	Method	Elite overlap (%) ↑	Inversion rate (%) ↓	Selected regret ↓
PushT	RTN	86.10%	6.06%	4.07229
	GPTQ	**93.50%**	**2.75%**	**0.996777**
	Module-MSE	91.17%	3.95%	1.63248
	Module-L1	91.48%	3.79%	2.43306
	Relative-MSE	90.92%	3.98%	1.77927
	Cost-MSE	87.60%	5.31%	3.84708
TwoRooms	RTN	80.08%	13.61%	11.1294
	GPTQ	94.95%	2.67%	1.97848
	Module-MSE	95.43%	**2.64%**	**1.1873**
	Module-L1	**95.53%**	2.66%	1.25887
	Relative-MSE	93.93%	4.01%	1.66361
	Cost-MSE	88.63%	7.32%	5.98562
Reacher	RTN	93.33%	4.98%	1.39875
	GPTQ	97.73%	**1.61%**	0.187712
	Module-MSE	**97.93%**	1.64%	**0.0758727**
	Module-L1	97.88%	1.74%	0.181424
	Relative-MSE	97.47%	2.15%	0.159305
	Cost-MSE	96.48%	2.92%	0.260614
Cube	RTN	84.42%	6.68%	2.64205
	GPTQ	**92.92%**	**2.96%**	0.7926
	Module-MSE	91.60%	3.41%	**0.65055**
	Module-L1	91.62%	3.48%	0.831126
	Relative-MSE	91.47%	3.83%	0.906241
	Cost-MSE	89.48%	4.69%	1.77819

Table B.2: Detailed decision-fidelity results for DINO-WM.

Task	Method	Elite overlap (%) ↑	Inversion rate (%) ↓	Selected regret ↓
PushT	RTN	16.78%	46.16%	27440.5
	GPTQ	63.83%	16.90%	4855.27
	Module-MSE	**78.35%**	**10.46%**	**1669.43**
	Module-L1	77.33%	10.70%	2172.35
	Relative-MSE	73.30%	12.73%	2933.24
	Cost-MSE	33.53%	32.26%	18356.5
TwoRooms	RTN	38.97%	23.41%	1.08622
	GPTQ	90.73%	2.75%	0.0279685
	Module-MSE	**93.07%**	**2.16%**	0.0170998
	Module-L1	92.62%	2.32%	0.0209006
	Relative-MSE	92.63%	2.48%	**0.013819**
	Cost-MSE	85.78%	4.46%	0.074933
Reacher	RTN	52.63%	14.45%	0.606662
	GPTQ	92.50%	2.32%	0.0139276
	Module-MSE	**93.45%**	**1.92%**	**0.00918045**
	Module-L1	93.25%	1.97%	0.0112634
	Relative-MSE	89.90%	3.02%	0.025059
	Cost-MSE	85.42%	4.36%	0.0521803
Cube	RTN	66.75%	11.59%	0.695867
	GPTQ	95.10%	1.61%	0.0120263
	Module-MSE	95.05%	1.48%	**0.0105492**
	Module-L1	**95.58%**	**1.39%**	0.0143456
	Relative-MSE	94.45%	1.79%	0.0172938
	Cost-MSE	89.62%	3.17%	0.066651

Table B.3: Detailed decision-fidelity results for PLDM.

Task	Method	Elite overlap (%) ↑	Inversion rate (%) ↓	Selected regret ↓
PushT	RTN	65.23%	17.49%	41.5512
	GPTQ	69.85%	14.53%	36.6468
	Module-MSE	**71.28%**	13.97%	28.5515
	Module-L1	70.95%	**13.93%**	**28.3379**
	Relative-MSE	66.77%	16.39%	35.4452
	Cost-MSE	64.13%	17.76%	52.4928
TwoRooms	RTN	78.28%	8.82%	8.83035
	GPTQ	90.52%	3.33%	5.0893
	Module-MSE	**91.98%**	**3.10%**	**2.16276**
	Module-L1	91.22%	3.26%	2.53982
	Relative-MSE	85.95%	5.43%	2.76167
	Cost-MSE	81.83%	6.79%	8.22504
Reacher	RTN	89.62%	10.39%	4.61286
	GPTQ	**94.93%**	**5.05%**	1.38869
	Module-MSE	94.77%	5.26%	1.27085
	Module-L1	94.05%	5.96%	1.61358
	Relative-MSE	93.93%	7.28%	**0.88136**
	Cost-MSE	92.87%	7.40%	2.00454
Cube	RTN	71.35%	13.64%	29.7865
	GPTQ	78.70%	9.82%	20.1499
	Module-MSE	79.22%	9.95%	19.118
	Module-L1	**80.02%**	**9.76%**	**17.6314**
	Relative-MSE	76.45%	11.35%	20.6992
	Cost-MSE	74.27%	12.61%	22.6472

B.3 Adjacent-path supervision: setup and detailed results

This experiment evaluates PushT with LeWM, DINO-WM, and PLDM. It compares Cost-MSE and Relative-MSE with their adjacent-path variants, using Module-MSE as a reference. Each adjacent-path objective averages the five consecutive rollout segments and the terminal goal segment defined in equation 5.1; no module-reconstruction term is added. All conditions follow the W4 quantization, calibration budget, learning-rate selection, planning, and fixed-candidate protocols in Appendix B.1.

Table B.4 reports all fixed-candidate fidelity results. Success rates are given in Table 4. Selected regret retains the native cost units of each model and should be compared only within a model. Bold denotes the best quantized result for each model and metric.

Table B.4: Detailed decision-fidelity results for adjacent-path supervision on PushT.

Model	Method	Elite overlap (%) ↑	Inversion rate (%) ↓	Selected regret ↓
LeWM	Module-MSE	**91.17%**	**3.95%**	**1.63248**
	Cost-MSE	87.60%	5.31%	3.84708
	Cost-MSE + Adj.	89.42%	4.85%	3.17127
	Relative-MSE	90.92%	3.98%	1.77927
	Relative-MSE + Adj.	90.68%	4.04%	2.03734
DINO-WM	Module-MSE	**78.35%**	**10.46%**	**1669.43**
	Cost-MSE	33.53%	32.26%	18356.5
	Cost-MSE + Adj.	63.20%	17.22%	4879.16
	Relative-MSE	73.30%	12.73%	2933.24
	Relative-MSE + Adj.	74.45%	12.31%	3126.25
PLDM	Module-MSE	**71.28%**	**13.97%**	**28.5515**
	Cost-MSE	64.13%	17.76%	52.4928
	Cost-MSE + Adj.	66.15%	17.25%	52.1886
	Relative-MSE	66.77%	16.39%	35.4452
	Relative-MSE + Adj.	67.58%	16.02%	39.1717

Adding adjacent-path terms to Cost-MSE improves elite overlap and inversion rate and reduces selected regret for all three models. For Relative-MSE, the adjacent variant improves overlap and inversion for DINO-WM and PLDM but not for LeWM, and selected regret increases for all three models. Module-MSE achieves the best value for every fidelity metric in each model.

B.4 Sensitivity to learning rate and calibration-set size

This experiment evaluates Module-MSE, Module-L1, Relative-MSE, and Cost-MSE on PushT with all three models. The learning-rate sweep fixes the calibration set at 128 trajectories and evaluates $0.5\times$, $1\times$, and $2\times$ the base rate. The calibration-size sweep fixes the learning rate at $1\times$ $(1/384)$ and uses 128, 32, 13, or 1 trajectory. Every condition runs for 384 optimization steps and is evaluated on 200 closed-loop episodes and 200 test states with 300 fixed candidates per state. Other settings follow Appendix B.1. Selected regret uses each model's native cost units and is compared only within a model.

Learning-rate sweep. Table B.5 gives the complete results underlying the top row of Figure 2.

Table B.5: Per-model results at three learning rates. In each cell, the first line reports success / elite overlap / inversion rate (all in percent), and the second line reports selected regret.

Loss	$0.5\times$	$1\times$	$2\times$
LeWM			
Module-MSE	84.5 / 91.17 / 3.95 1.63248	85.5 / 91.95 / 3.55 1.50452	85.0 / 91.22 / 3.80 2.29985
Module-L1	81.0 / 91.00 / 3.95 1.32224	84.0 / 90.75 / 3.79 1.35804	82.0 / 91.48 / 3.79 2.43306
Relative-MSE	80.5 / 90.92 / 3.98 1.77927	81.0 / 89.93 / 4.60 2.6221	75.0 / 85.37 / 6.66 5.31395
Cost-MSE	78.0 / 87.60 / 5.31 3.84708	74.0 / 82.88 / 7.59 9.75507	49.5 / 65.25 / 16.61 27.9864
DINO-WM			
Module-MSE	73.5 / 76.80 / 11.07 2447.05	75.5 / 77.97 / 10.69 1857.65	76.0 / 76.95 / 10.97 2409.91
Module-L1	72.0 / 77.60 / 10.79 1537.9	75.0 / 77.02 / 10.86 1983.4	74.5 / 75.35 / 11.52 2477
Relative-MSE	73.0 / 73.30 / 12.73 2933.24	71.5 / 70.68 / 14.39 3579.02	63.5 / 60.75 / 19.19 7491.59
Cost-MSE	22.0 / 33.53 / 32.26 18356.5	4.5 / 13.80 / 46.68 30137.4	4.5 / 10.88 / 49.77 28345.6
PLDM			
Module-MSE	39.0 / 71.28 / 13.97 28.5515	42.0 / 71.72 / 13.91 27.866	34.5 / 69.40 / 15.64 39.7794
Module-L1	36.5 / 70.95 / 13.93 28.3379	40.5 / 72.57 / 13.39 28.8089	36.0 / 67.70 / 16.29 39.1951
Relative-MSE	32.0 / 66.77 / 16.39 35.4452	28.0 / 59.27 / 21.30 62.2996	23.0 / 46.70 / 29.09 92.5772
Cost-MSE	30.0 / 64.13 / 17.76 52.4928	25.5 / 51.55 / 24.33 82.0173	11.0 / 25.25 / 42.02 206.345

Calibration-set-size sweep. Table B.6 gives the complete results underlying the bottom row of Figure 2.

Table B.6: Per-model results at different calibration-set sizes. In each cell, the first line reports success / elite overlap / inversion rate (all in percent), and the second line reports selected regret. Calibration percentages are relative to the full set of 128 trajectories; parentheses give the integer trajectory count.

Loss	100% (128)	25% (32)	10% (13)	1% (1)
LeWM				
Module-MSE	85.5 / 91.95 / 3.55 1.50452	82.5 / 91.18 / 3.89 1.4371	79.5 / 89.72 / 4.32 2.45891	75.0 / 86.33 / 6.09 5.35638
Module-L1	84.0 / 90.75 / 3.79 1.35804	79.5 / 90.98 / 3.90 1.86748	83.5 / 89.80 / 4.40 2.67727	73.5 / 86.10 / 6.07 4.85389
Relative-MSE	81.0 / 89.93 / 4.60 2.6221	81.5 / 88.98 / 4.92 3.58135	77.5 / 85.90 / 6.33 5.55648	64.0 / 76.38 / 10.24 15.1666
Cost-MSE	74.0 / 82.88 / 7.59 9.75507	69.0 / 81.60 / 8.17 8.06481	63.0 / 79.62 / 8.56 11.4561	56.5 / 74.87 / 10.73 18.0744
DINO-WM				
Module-MSE	75.5 / 77.97 / 10.69 1857.65	71.0 / 76.68 / 10.97 1735.08	74.0 / 72.23 / 13.24 2529.68	42.5 / 54.82 / 19.92 7314.21
Module-L1	75.0 / 77.02 / 10.86 1983.4	70.0 / 74.82 / 11.77 2302.08	74.5 / 72.48 / 13.11 2988.39	39.0 / 52.47 / 21.13 8258.36
Relative-MSE	71.5 / 70.68 / 14.39 3579.02	66.5 / 66.57 / 15.56 4166.33	58.5 / 63.53 / 17.43 5368.15	11.5 / 31.80 / 33.17 19624.6
Cost-MSE	4.5 / 13.80 / 46.68 30137.4	6.0 / 23.33 / 40.97 27900.7	6.0 / 17.37 / 43.73 29213.3	5.0 / 12.37 / 47.91 29503
PLDM				
Module-MSE	42.0 / 71.72 / 13.91 27.866	47.0 / 69.93 / 13.95 38.5553	39.5 / 70.60 / 14.01 27.6971	31.0 / 64.95 / 17.23 36.742
Module-L1	40.5 / 72.57 / 13.39 28.8089	44.0 / 70.83 / 14.31 36.717	41.5 / 69.73 / 14.71 28.7341	32.0 / 65.45 / 16.91 40.423
Relative-MSE	28.0 / 59.27 / 21.30 62.2996	32.0 / 59.32 / 20.75 58.0366	34.0 / 59.58 / 19.90 52.8098	28.5 / 59.70 / 20.28 55.7891
Cost-MSE	25.5 / 51.55 / 24.33 82.0173	25.5 / 52.22 / 23.59 91.9119	32.0 / 54.13 / 22.86 72.5171	27.0 / 58.70 / 20.74 65.7632