ADAPTIVE LATENT CAPACITY FOR WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Adaptive LeWorldModel (ALeWM), a world model based on a joint-embedding predictive architecture (JEPA) that learns to concentrate predictive information in compact prefixes of a wide latent representation. To encourage this ordering, ALeWM learns a sequence-conditioned distribution over prefix lengths and trains the predictor to estimate the full next embedding from a sampled input prefix. As standard anti-collapse objectives encourage variation across latent coordinates and do not organize them by predictive importance, we also introduce MixSIGReg. MixSIGReg regularizes the masked embeddings against a prior-weighted mixture with Gaussian active prefixes and zeros in the remaining coordinates. As a result, the ALeWM objective encourages early coordinates to retain information useful for prediction and recursive planning. Our analysis shows that the mixture distribution used by MixSIGReg assigns higher variance to earlier coordinate blocks and lower variance to later ones. In addition, we show that, under specified assumptions, prediction error is minimized by placing the information most useful for prediction in earlier blocks. Empirically, we study the behavior of ALeWM in a controlled dynamical system with known state variables and in goal-conditioned visual control. We show that ALeWM consistently achieves higher mean success rates than tuned fixed-width LeWM, with lower planning capacity on average.

1 INTRODUCTION

World models allow agents to anticipate the consequences of actions and use imagined futures to guide behavior (Ha & Schmidhuber, 2018; Hafner et al., 2025). Recent research directions include world action models, which jointly generate future observations and actions for robot control (NVIDIA et al., 2025; Kim et al., 2026; Li et al., 2026a; Ye et al., 2026), and explicit 3D world generation and reconstruction of dynamic scenes (HunyuanWorld Team et al., 2025; Yang et al., 2025; Zhang et al., 2026a). Joint-embedding predictive architectures (JEPAs) offer a different route: they predict future embeddings without reconstructing observations (LeCun, 2022; Assran et al., 2025). LeWorldModel (LeWM) makes this approach accessible through a compact encoder and action-conditioned predictor trained jointly from scratch (Maes et al., 2026). Its two-term objective combines next-embedding prediction with SIGReg, which discourages collapse by encouraging Gaussian-distributed embeddings (Balestriero & LeCun, 2025). This reward-free, end-to-end formulation provides a practical foundation for studying how learned representations support planning.

In latent planning, the learned representation is the state carried through every imagined trajectory. Recent studies of LeWM highlight the importance of how latent states are organized and used for control (Li et al., 2026b; Nguyen et al., 2026b). We focus on a complementary question: can a world model retain a wide embedding for learning while concentrating predictive information in a short prefix, consisting of its first few coordinates? Such an ordering would let one trained model support different latent-state budgets without fitting a separate representation for each width. Compact prefixes could offer several benefits: helping planning focus on control-relevant information while limiting the influence of irrelevant visual details, allowing different tasks or episodes to use prefix lengths suited to their information needs, and leaving the remaining coordinates available for features useful to other downstream tasks. Ordered representations already provide this flexibility in reconstruction and retrieval (Rippel et al., 2014; Kusupati et al., 2022) using fixed, input-independent weighting over prefix lengths. For a world model, the additional challenge is to make these prefixes useful through repeated action-conditioned predictions and goal comparisons during planning.

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.

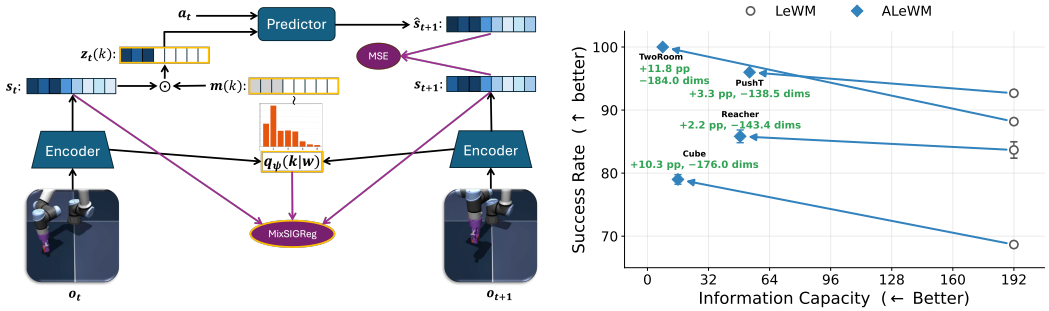


Figure 1: ALeWM overview. **Left:** A shared encoder maps consecutive observations to latent states, and a learned network selects the active prefix capacity k for masking the state $\mathbf{s}_t$. The predictor estimates the full next latent state from the masked embedding and action, while the prediction loss and MixSIGReg objective jointly train the representation. In orange, our modifications to the LeWM framework. **Right:** Gains of ALeWM over fixed-width LeWM using ViT-Tiny models with latent dimension 192. Moving left and up indicates lower average capacity and higher test success rate.

Standard anti-collapse objectives do not impose this ordering. SIGReg encourages an isotropic Gaussian embedding distribution, which treats all coordinates symmetrically. VICReg discourages low coordinate variance and cross-coordinate covariance (Bardes et al., 2022). These objectives encourage variation across the representation, but do not assign greater predictive importance to earlier coordinates. One alternative is to reduce the embedding width, but empirically we noticed that this can degrade planning performance. A possible explanation is a tension between representation and prediction: narrow embeddings may restrict the representation of useful scene information, while wider embeddings may distribute predictive information across many coordinates, potentially complicating the dynamics that the predictor must learn. We therefore seek to retain a wide embedding during learning and train a compact prefix to predict the full next embedding, encouraging early coordinates to capture information useful for prediction and planning.

Here, we introduce *Adaptive LeWorldModel* (ALeWM), a novel method that learns how much of a wide embedding to retain for prediction and planning. Using nested dropout as inspiration (Rippel et al., 2014), we define a capacity network that predicts a distribution over prefix lengths for each observation sequence. During training, the predictor receives a sampled prefix of each history embedding and must predict the full next embedding. This gives the early coordinates a clear role: retain information that helps predict the wider representation even when only a short prefix is available. To accommodate this unequal use of coordinates, we introduce *MixSIGReg*, which regularizes masked embeddings against a prior-weighted mixture of Gaussian prefixes and zero suffixes. At deployment, the capacity network selects a prefix length from the initial and goal observations, fixing it throughout the episode. Planning then uses this prefix for recursive prediction and goal comparison.

We analyze ALeWM to explain why early coordinates are favored. The MixSIGReg reference assigns lower variance to later coordinate blocks, reflecting how often they remain active. Our prediction analysis further shows that, under stated assumptions, placing the most useful information first minimizes expected prediction error. Empirically, we study ALeWM in a controlled dynamical system and four common simulated visual control tasks. In the controlled system, an eight-dimensional ALeWM concentrates linearly decodable information about four state variables in its first four coordinates. On the visual control tasks, selected ALeWM models achieve higher mean success rates than tuned fixed-width LeWM, with gains of up to approximately 7 percentage points (PP) in success rate and an average reduction of 56% in planning capacity.

To summarize, we make the following novel contributions: (1) We introduce ALeWM with MixSIGReg to concentrate predictive information in compact prefixes while retaining a wide latent representation in JEPA-based architectures; (2) We characterize the mixture target's decreasing variance across coordinate blocks and establish conditions for ordering predictive information by its contribution to reducing prediction error; (3) We demonstrate concentration of linearly decodable state information in a controlled dynamical system and improved planning success with substantially lower planning capacity on average in goal-conditioned visual control.

2 Related Work

Joint-embedding world models. Joint-embedding predictive architectures (JEPAs) learn representations by predicting related inputs in embedding space rather than reconstructing observations (LeCun, 2022). I-JEPA applies this principle to masked image prediction (Assran et al., 2023), while V-JEPA extends feature prediction to video (Bardes et al., 2024). For planning, DINO-WM and V-JEPA 2-AC learn action-conditioned dynamics over pretrained visual features (Zhou et al., 2025; Assran et al., 2025), whereas PLDM, LeWM and Sensorimotor World Models (SMWM) jointly learn visual representations and dynamics from reward-free data (Sobal et al., 2026; Maes et al., 2026; Ivashkov et al., 2026). However, predictability alone does not ensure relevance for control: representations may retain persistent distractors (Sobal et al., 2022), and low prediction error on the data distribution need not control error on states reached by a planner (You et al., 2026). Latent-recovery analyses also rely on specific assumptions, including dimension matching and isotropic Gaussian Ornstein–Uhlenbeck dynamics for population LeJEPA (Klindt et al., 2026), or controlled linear-Gaussian dynamics, invertible observations, spectral separation and sufficient conditional action variation (Zhang et al., 2026b). Such results do not establish semantic coordinate ordering for learned prefixes. We build on LeWM's end-to-end setting and LeJEPA's SIGReg machinery (Balestriero & LeCun, 2025) to study representation capacity and geometry.

Regularizing predictive embeddings. Anti-collapse regularizers differ in the properties they impose on the embedding distribution. Barlow Twins reduces cross-coordinate redundancy (Zbontar et al., 2021), W-MSE whitens features (Ermolov et al., 2021), and VICReg combines invariance with a coordinate-wise variance floor and an off-diagonal covariance penalty (Bardes et al., 2022). SIGReg matches random projections to a standard Gaussian (Balestriero & LeCun, 2025; Zimmermann et al., 2025), while VISReg separates scale control from sliced-Wasserstein distribution-shape matching (Wu et al., 2026). Other approaches impose subspace constraints or sparse targets: Sub-JEPA regularizes fixed random subspaces (Zhao et al., 2026), and Rectified LpJEPA introduces sparse mixed targets (Kuang et al., 2026b). The concurrent study LpWM uses a rectified generalized-Gaussian target to learn non-negative sparse predictive representations (Kuang et al., 2026a). LpWM sparsifies a fixed-width code, unlike our construction, which defines nested prefixes.

Ordered representations. Learning ordered representations has been studied in several contexts, including visual tokenization (Bachmann et al., 2025; Yan et al., 2025; Yang et al., 2026; Lu et al., 2026) and representation learning, as depicted next. Nested dropout trains only a sampled prefix, exposing earlier coordinates more often (Rippel et al., 2014). Matryoshka Representation Learning and Information-Ordered Bottlenecks similarly train representations to remain useful across prescribed prefix widths (Kusupati et al., 2022; Ho et al., 2025). MIC adds prefix–residual correlation, variance and uniformity penalties to Matryoshka training (Nguyen et al., 2026a). Learning the retained capacity also has precedents. Variational Nested Dropout learns input-conditioned ordered masks (Cui et al., 2023), while SparC-IB learns input-dependent categorical prefix lengths for supervised prediction (Samaddar et al., 2023). Both use decoder networks to train and organize the latent representation in non-temporal tasks: one reconstructs the input, and the other predicts a label. We instead learn a decoder-free sequence-conditioned distribution over state-coordinate prefix lengths through action-conditioned prediction of future embeddings, and match the aggregate distribution of masked embeddings using characteristic functions. Lastly, in work concurrent with this study, Liu et al. (2026) proposed using nested dropout for action-token ordering and reconstruction in vision-language-action models for robot control.

3 Method

To motivate our method, we begin with the LeWM formulation (Maes et al., 2026), which combines two objectives: (1) a prediction loss between the predicted next-step embedding and the encoder's embedding of the corresponding observation; and (2) SIGReg regularization (Balestriero & LeCun, 2025), which encourages the aggregate embedding distribution to match an isotropic standard Gaussian (see Appendix A for a full description). These objectives do not impose a coordinate ordering, allowing predictive information to be distributed throughout the latent representation. Empirically, we observed that increasing the embedding dimension can often reduce planning performance. One possible explanation is that predicting a larger latent can make accurate prediction more difficult. Since during rollout the learned predictor is applied recursively, prediction errors can propagate and

affect the planner's evaluation of candidate actions. Reducing the embedding dimension can mitigate this issue, but requires careful tuning and restricts the expressivity of the latent representation.

Hence, we seek to retain the expressivity of a wide embedding while encouraging the information needed for prediction to concentrate in compact prefixes, resulting in reduced difficulty of prediction. Inspired by nested dropout (Rippel et al., 2014), which was proposed in the context of autoencoders, we learn a sequence-conditioned distribution over a finite set of admissible prefix capacities. During training, the predictor receives a sampled prefix of the latent embedding vectors and predicts the full next embedding, encouraging early coordinates to retain information useful across multiple capacities. The prefix size is sampled from a distribution produced by a capacity network which is jointly *learned* with the rest of the network parameters, namely, the encoder and the predictor. At deployment, the capacity network selects a fixed prefix size for each episode based on the known initial and goal frames. Our construction displays several favorable properties: (1) As the capacity network is learned during training, the prefix size can be set automatically without an exhaustive search after training; (2) different episodes can use different capacities without training separate models; and (3) alternative prefix sizes can be examined for each episode after training. A graphical description of our method is presented in Figure 1. We describe our construction in detail next.

3.1 Controlling the Information Capacity

In what follows, scalars are denoted in lower-case (e.g., x), vectors in bold lower-case (e.g., $\mathbf{x}$), and matrices in bold upper-case (e.g., $\mathbf{X}$). Assume we are given a training sequence of F observations $\mathbf{w} = (\mathbf{o}_1, \ldots, \mathbf{o}_F)$, and a set of associated actions $(\mathbf{a}_1, \ldots, \mathbf{a}_{F-1})$. We define the set of admissible capacities as an ordered set $\mathcal{K} = \{k_1 < \cdots < k_C\} \subseteq \{1, \ldots, d\}$ with $k_C = d$, the maximal latent embedding dimension, and $k_c \in \mathbb{N} \setminus \{0\} \;\; \forall c \in \{1, \ldots, C\}$. For instance, if we fix $C = d$, then $\{k_1 = 1 < k_2 = 2 < \cdots < k_C = d\}$. Alternatively, if we fix $C = d/2$, a possible capacity set is the set of even numbers $\{k_1 = 2 < k_2 = 4 < \cdots < k_C = d\}$, assuming that d is even. In our experiments, the capacity set is fixed for each d (Appendix E specifies the choices per experiment).

Given $k \in \mathcal{K}$, define the prefix mask by $\mathbf{m}(k)$ with coordinate values $[\mathbf{m}(k)]^j = \mathbb{1}[j \leq k], \;\; \forall j \in \{1, \ldots, d\}$. The latent embedding $\mathbf{s}_t \in \mathbb{R}^d$ and the masked-latent embedding $\mathbf{z}_t(k) \in \mathbb{R}^d$ for a time-step t are obtained according to:

$$\mathbf{s}_t = \mathrm{enc}_\theta(\mathbf{o}_t), \qquad \mathbf{z}_t(k) = \mathbf{m}(k) \odot \mathbf{s}_t = \mathbf{M}_k \mathbf{s}_t, \qquad z_t^j(k) = \begin{cases} s_t^j, & j \leq k, \\ 0, & j > k, \end{cases} \tag{1}$$

where $\mathbf{M}_k = \mathrm{diag}(\mathbf{m}(k)) \in \{0, 1\}^{d \times d}$ and $\mathrm{enc}_\theta(\cdot)$ is an encoder network. Next, we define a *learned* capacity selector network $q_\psi(k \mid \mathbf{w})$ and a *fixed* capacity prior distribution $\pi_0(k; \alpha)$ whose role will become clear in the MixSIGReg regularizer. Both distributions lie in Δ^{C-1}. We construct the capacity prior from polynomial weights over reverse capacity ranks. For degree $\alpha \in \mathbb{R}$, treated as a hyperparameter, define

$$w_\alpha(k_c) = (C - c + 1)^\alpha, \qquad \pi_0(k_c; \alpha) = \frac{w_\alpha(k_c)}{\sum_{\ell=1}^{C} w_\alpha(k_\ell)}. \tag{2}$$

Thus, $\alpha > 0$ favors smaller capacities, $\alpha = 0$ gives a uniform prior, and $\alpha < 0$ favors larger capacities. In this study, our construction depends only on the ordering of the capacity support, not on the numerical spacing between its elements. We leave for future work further investigations of other weighting schemes for the capacities. To learn the capacity-network parameters, we initialize the network such that its output is equal to prior probabilities $\boldsymbol{\pi}_0(\cdot, \alpha)$ and use the straight-through Gumbel–Softmax estimator: we draw $k \sim q_\psi(\cdot \mid \mathbf{w})$, use the corresponding hard prefix in the forward pass, and differentiate through its soft relaxation (Jang et al., 2017; Maddison et al., 2017) (full construction in Appendix B). We intentionally condition the capacity on a sequence of observations to constrain the capacity to be the same for the entire sequence while allowing the model to choose different capacities for different sequences. Intuitively, for the same sequence we expect that the same information capacity will store all the relevant details, yet it may vary between sequences.

3.2 Training Objective and Planning

As in LeWM, our objective consists of prediction loss for the next state given a window of previous states, and a regularization term that prevents embedding collapse. For clarity, we present next-step prediction; the objective extends naturally to sequences of frames and actions with frame skips.

Prediction loss. Let $\mathbf{s}_t = \mathrm{enc}_\theta(\mathbf{o}_t)$ and $\mathbf{s}_{t+1} = \mathrm{enc}_\theta(\mathbf{o}_{t+1})$ be the encoding of the current and next states, respectively. In addition, let $\mathbf{a}_t$ be the current (ground-truth) action and let $k \sim q_\psi(\cdot \mid \mathbf{w})$ be the sampled capacity with its corresponding mask $\mathbf{M}_k$. The model predicts the next state as follows:

$$\mathbf{z}_t(k) = \mathbf{M}_k \mathbf{s}_t, \qquad \hat{\mathbf{s}}_{t+1} = \mathrm{pred}_\phi(\mathbf{z}_t(k), \mathbf{a}_t), \tag{3}$$

where $\mathrm{pred}_\phi(\cdot,\cdot)$ is a predictor network. We define the prediction target to be the full next encoder embedding $\mathbf{s}_{t+1}$. Thus, the prediction loss is given by

$$\mathcal{L}_{\mathrm{pred}} = \frac{1}{B(F-1)} \sum_{b=1}^{B} \sum_{t=1}^{F-1} ||\hat{\mathbf{s}}_{b,t+1} - \mathbf{s}_{b,t+1}||_2^2 \tag{4}$$

where B is the batch size, and the subscript b denotes the index of an example in a batch. Note that in the objective both the input and target embeddings are learned jointly without relying on a fixed external teacher. The intuition behind predicting the full next embedding is to encourage the prefix to retain information useful for predicting the wider representation, while preserving flexibility during representation learning. Since earlier coordinates are retained at least as often as later ones, the model is encouraged to concentrate predictive information near the beginning of the embedding.

MixSIGReg. Applying the prediction loss alone can lead to representation collapse. Common regularization schemes like SIGReg (Balestriero & LeCun, 2025; Maes et al., 2026) and VICReg (Bardes et al., 2022) encourage non-degenerate variance uniformly across all coordinates. Applying this constraint to the masked embedding may not be appropriate in our case, as we require later coordinates to be active less often and to be zero when masked. Indeed, in Section 4.3, we empirically observe that the SIGReg loss does not enforce a small prefix size for the masked embedding. To account for this, we define a reference distribution for each capacity k, with a standard Gaussian on the active coordinates and zeros on the remaining coordinates: $p_0(\mathbf{z}_t \mid k) = \mathcal{N}(\mathbf{z}_t^{1:k}; 0, \mathbf{I}_k) \otimes \delta_0(\mathbf{z}_t^{k+1:d})$, where δ_0 is the Dirac distribution centered at zero, $\mathbf{z}_t^{1:k}$ denotes the first k coordinates of $\mathbf{z}_t$, and $\mathbf{z}_t^{k+1:d}$ denotes its remaining $d-k$ coordinates.

Let $\pi_0(k;\alpha)$ be the polynomial capacity prior in Eq. 2. The marginal reference distribution of $\mathbf{z}_t$ takes the form of a mixture of Gaussian distributions:

$$p_0(\mathbf{z}_t) = \sum_{k \in \mathcal{K}} \pi_0(k;\alpha) \left[\mathcal{N}(\mathbf{z}_t^{1:k}; 0, \mathbf{I}_k) \otimes \delta_0(\mathbf{z}_t^{k+1:d})\right]. \tag{5}$$

Having defined the reference distribution, we follow the SIGReg recipe by projecting embeddings onto unit-norm directions and minimizing a univariate Epps–Pulley (Epps & Pulley, 1983) test statistic adapted to this reference distribution. Specifically, we draw P projection vectors $\{\mathbf{u}^{(p)}\}_{p=1}^P$, with each $\mathbf{u}^{(p)} \in \mathbb{S}^{d-1}$. Under this reference law, $\mathbf{u}^{(p)\top}\mathbf{z}_t \mid K = k \sim \mathcal{N}\big(0, \mathbf{u}^{(p)\top}\mathbf{M}_k\mathbf{u}^{(p)}\big)$. Therefore, for a particular direction p the target characteristic function at an evaluation point τ is:

$$\phi_0^{(p)}(\tau) = \sum_{k \in \mathcal{K}} \pi_0(k;\alpha) \exp\left(-\frac{\tau^2}{2} \left\|\mathbf{M}_k \mathbf{u}^{(p)}\right\|_2^2\right). \tag{6}$$

Likewise, we estimate the characteristic function at frame position t by marginalizing over prefix capacities using the learned capacity distribution $q_\psi(k \mid \mathbf{w}_b)$ and applying a Monte Carlo average over batch observations:

$$\hat{\phi}_t^{(p)}(\tau) = \frac{1}{B} \sum_{b=1}^{B} \sum_{k \in \mathcal{K}} q_\psi(k \mid \mathbf{w}_b) \exp\Big(i\tau \mathbf{u}^{(p)\top}\mathbf{M}_k \mathbf{s}_{b,t}\Big). \tag{7}$$

The resulting loss, MixSIGReg, is obtained by averaging over projections and frame positions:

$$\mathcal{L}_{\mathrm{MixSIG}} = \frac{B}{PF} \sum_{p=1}^{P} \sum_{t=1}^{F} \int w(\tau) \left|\hat{\phi}_t^{(p)}(\tau) - \phi_0^{(p)}(\tau)\right|^2 d\tau. \tag{8}$$

Here $w(\tau) = \exp(-\tau^2/2)$ is the Gaussian frequency window and the integral is approximated using the quadrature specified in Appendix E.2. Overall, MixSIGReg combines collapse prevention with a preference for compact representations. Its reference distribution assigns more frequent use to earlier coordinates while requiring active prefixes to retain variation.

To summarize, the complete training objective of ALeWM is:

$$\mathcal{L} = \mathcal{L}_{\mathrm{pred}} + \lambda \mathcal{L}_{\mathrm{MixSIG}} \tag{9}$$

Fixed-prefix rollout. As in LeWM, ALeWM performs planning using model predictive control (MPC). Unlike LeWM, however, a single model supports planning with multiple prefix capacities. Given an episode's initial observation $\mathbf{o}_1$ and goal observation $\mathbf{o}_g$, we select the modal supported capacity $k_r = \arg\max_{k \in \mathcal{K}} q_\psi(k \mid \mathbf{w})$ where $\mathbf{w} = (\mathbf{o}_1, \mathbf{o}_g)$, and hold it fixed throughout the episode. Candidate action sequences are optimized so that the predicted terminal latent state at planning horizon H approaches the goal embedding in the active-prefix space. Specifically, the same prefix mask $\mathbf{M}_{k_r}$ is used across all recursive rollout steps and replans, as well as in the terminal goal cost. Other deployment rules are possible, such as sampling a capacity from q_ψ or evaluating multiple supported capacities per episode according to a specified selection criterion, yet we did not pursue these directions in this study. Concretely,

$$\hat{\mathbf{z}}_1 = \mathbf{M}_{k_r} \operatorname{enc}_\theta(\mathbf{o}_1), \qquad \hat{\mathbf{z}}_{t+1} = \mathbf{M}_{k_r} \operatorname{pred}_\phi(\hat{\mathbf{z}}_t, \mathbf{a}_t). \tag{10}$$

Let $\mathbf{s}_g$ denote the full encoder embedding of the goal observation. Candidate plans are scored only on the active target dimensions:

$$\mathcal{C}(\mathbf{a}_{b,1:H-1}; k_r) = \frac{1}{k_r} \left\| \hat{\mathbf{z}}_H - \mathbf{M}_{k_r} \mathbf{s}_g \right\|_2^2 . \tag{11}$$

The optimal candidate plan is then selected based on $\arg\min_{\mathbf{a}_{1:H-1}} \mathcal{C}(\mathbf{a}_{1:H-1}; k_r)$ using the Cross-Entropy Method (CEM) (Rubinstein & Kroese, 2004).

3.3 Theoretical Motivation For Ordered Representation

Our analysis separates two effects. The MixSIGReg reference prescribes an ordered variance profile for masked embeddings, while nested prediction rewards information that remains useful across multiple prefix sizes.

Informal proposition 1 (Ordered variance and predictive value)**.** *Let $K_\pi \sim \pi_0$ be independent of the data and define $\rho_\pi^j = \Pr(K_\pi \geq j)$. The MixSIGReg reference satisfies*

$$\operatorname{Var}(z_0^j) = \rho_\pi^j, \qquad \rho_\pi^1 \geq \cdots \geq \rho_\pi^d.$$

Exact population matching transfers this variance profile to the learned masked embeddings.

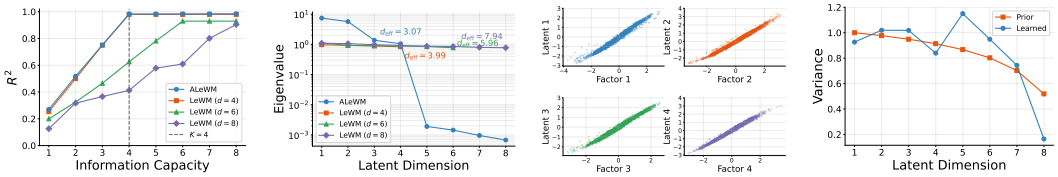
For a fixed encoder with square-integrable targets, let R_k be the minimum population squared error for predicting the full next embedding from the first k coordinates of the embedding history and the actions. Nested information gives $R_\ell \leq R_k$ for $k \leq \ell$. Setting $k_0 = 0$ and $\Delta_c = R_{k_{c-1}} - R_{k_c} \geq 0$, we obtain

$$\mathbb{E}_{K_\pi}[R_{K_\pi}] = R_0 - \sum_{c=1}^{C} \rho_\pi^{k_c} \Delta_c.$$

Each predictive gain is therefore weighted by its block's retention probability. If these gains are additive and unchanged by reordering, placing larger gains earlier minimizes this risk.

Formal assumptions and proofs appear in Appendix C. We note that these results motivate the proposed ordering but do not guarantee that joint training achieves it.

Intuition. One can think of a prefix mask as a cutoff. Coordinate j appears only when the cutoff reaches j. In the reference distribution it has unit variance when present and is zero otherwise, so its overall masked variance is simply its survival probability ρ_π^j. Any cutoff that keeps a later coordinate also keeps all earlier ones; this makes the masked variances non-increasing. Nested prediction supplies the complementary pressure. For a fixed representation, an ideal predictor with a longer prefix cannot have larger minimum squared error because it has all the information in every shorter prefix. The decrease Δ_c measures the predictive value added by block c. Under a random cutoff, that value is available only when the block survives, so early predictive information is reused across more capacity choices. To see the resulting ordering pressure, suppose two blocks carry fixed predictive gains. Placing the larger gain first makes its reduction in error count for more sampled capacities while leaving it later makes it count only when a longer prefix is selected. Exchanging a larger late gain with a smaller early one therefore lowers expected prediction error whenever the two positions have different survival probabilities. The preference therefore is for larger reductions in risk, not for larger remaining risks, near the start. Hence, if later blocks add little oracle predictive value, a short prefix is nearly as predictive as the full embedding.



(a) Linear state recovery. (b) Latent covariance spectrum. (c) Whitened Procrustes alignment. (d) Masked embedding variance.

Figure 2: Four-factor controlled damped oscillator diagnostics. **(a)** Mean linear state-recovery R^2 as prefix size increases; the dashed line marks the true factor count $k = 4$. For LeWM with $d \in \{4, 6\}$ the curves are continued horizontally beyond their trained dimension only as a visual reference. **(b)** Raw unmasked latent covariance spectra. **(c)** ALeWM whitened Procrustes alignment, fitted using training statistics. **(d)** ALeWM empirical masked-latent variance vs the target prior variance.

4 Experiments

We evaluated ALeWM on a toy problem (section 4.1) as well as on a diverse set of simulated 2D and 3D tasks (section 4.2). Unless specified otherwise, we report the average performance and standard error of the mean (SEM) over 3 random seeds. Full experimental details are given in Appendix E.

4.1 Toy Example - Controlled Damped Oscillators

To gain insight into how ALeWM organizes dynamically relevant information across its latent coordinates, we study a controlled system with a known four-dimensional state. This setting allows us to test whether ALeWM can concentrate linearly decodable state information in an early prefix while retaining a wider latent representation. The system contains two independently controlled damped oscillators with state $\mathbf{x}_t = (p_t^1, v_t^1, p_t^2, v_t^2)$, corresponding to the oscillators' positions and velocities. We record trajectories containing eight transitions. The actions (a_t^1, a_t^2) specify the two oscillators' equilibrium positions and are switched after four transitions. To form each observation, an invertible nonlinear map produces four strong observation coordinates, while six noisy nonlinear measurements mix all four factors along fixed random directions. Thus, each observation is ten-dimensional but contains only four dynamical factors: four coordinates are strongly related to the factors, while the remaining six provide weaker, noisier measurements of them. We train an eight-dimensional ALeWM with capacity support $\mathcal{K} = \{1, \ldots, 8\}$ and compare it with separately trained LeWM models with $d \in \{1, \ldots, 8\}$ latent dimensions. The results below are shown for one seed, yet they were consistent across several seeds. Appendix E.1 provides full experimental details.

Figure 2 compares latent-structure diagnostics for ALeWM and fixed-width LeWM models with $d \in \{4, 6, 8\}$ on the held-out test set. LeWM attains $R^2 = 0.980$ when its dimension matches the four-dimensional state, but recovery falls to 0.930 and 0.904 at $d = 6$ and $d = 8$, respectively (Figure 2(a)). This result is consistent with Klindt et al. (2026), whose identifiability guarantee assumes matching representation and latent dimensions and leaves the mismatched regime open. Moreover, LeWM remains nearly isotropic at every width (Figure 2(b)). In contrast, Figure 2(a)–(c) show that ALeWM resolves this width–prefix tradeoff with non-degenerate prior variance for all capacities (Figure 2(d)). Although ALeWM was trained with maximum dimension $d = 8$, its first four coordinates attain $R^2 = 0.983$, with minor improvement from the remaining coordinates. This matches the dimension-matched LeWM model while outperforming the selected wider models. ALeWM also exhibits four dominant covariance directions and close alignment between its four-coordinate prefix and the four factors after fitting an orthogonal alignment matrix. Together, these results show that ALeWM can concentrate linearly decodable state information in a compact prefix without restricting the ambient latent width. We note that these results support prefix concentration; however, coordinate-wise disentanglement or recovery of the true factor count is not guaranteed.

4.2 Main Results

Next, we evaluate goal-conditioned control on the four continuous-action benchmarks used by LeWM (Maes et al., 2026): TwoRoom, PushT, Reacher, and OGBench-Cube. For each dataset, we

Table 1: LeWM and ALeWM mean test success rate ($\pm$ SEM) and mean planning prefix $\mathbb{E}[K_{\text{plan}}]$.

Backbone ($d_{\max}$)	Method	TwoRoom		PushT		Reacher		OGBench-Cube	
		SR (↑)	$\mathbb{E}[K_{\text{plan}}]$	SR (↑)	$\mathbb{E}[K_{\text{plan}}]$	SR (↑)	$\mathbb{E}[K_{\text{plan}}]$	SR (↑)	$\mathbb{E}[K_{\text{plan}}]$
ViT-Tiny (96)	LeWM (best d)	95.00 ± 0.00	32.00	89.33 ± 0.44	96.00	83.67 ± 0.73	64.00	72.33 ± 1.17	96.00
	LeWM ($d = 96$)	92.17 ± 0.17	96.00	89.33 ± 0.44	96.00	82.00 ± 1.76	96.00	72.33 ± 1.17	96.00
	ALeWM	$\mathbf{99.50 \pm 0.00}$	13.33	$\mathbf{94.67 \pm 0.83}$	32.00	$\mathbf{85.83 \pm 0.60}$	21.33	$\mathbf{77.17 \pm 1.17}$	16.00
ViT-Tiny (192)	LeWM (best d)	95.00 ± 0.00	32.00	92.67 ± 0.33	160.0	84.50 ± 1.15	160.0	72.33 ± 1.17	96.00
	LeWM ($d = 192$)	88.17 ± 0.33	192.0	92.67 ± 0.33	192.0	83.67 ± 1.30	192.0	68.67 ± 0.17	192.0
	ALeWM	$\mathbf{100.0 \pm 0.00}$	8.00	$\mathbf{96.00 \pm 0.00}$	53.49	$\mathbf{85.83 \pm 1.01}$	48.59	$\mathbf{79.00 \pm 0.76}$	16.00
ViT-Small (384)	LeWM (best d)	96.50 ± 1.04	32.00	94.17 ± 1.01	256.0	85.83 ± 0.44	64.00	70.50 ± 0.58	16.00
	LeWM ($d = 384$)	83.17 ± 0.17	384.0	92.83 ± 0.33	384.0	84.67 ± 1.20	384.0	69.33 ± 1.20	384.0
	ALeWM	$\mathbf{100.0 \pm 0.00}$	8.16	$\mathbf{95.50 \pm 0.29}$	55.36	$\mathbf{86.17 \pm 0.88}$	32.00	$\mathbf{77.67 \pm 1.20}$	32.00

construct a new episode-level split with no overlap among the training, validation, and test episodes, thereby preventing data leakage. The validation and test sets contain 100 and 200 episodes, respectively, and the remaining episodes are used for training. Validation set is used for hyper-parameter selection in all experiments throughout. We follow the training and planning protocol of LeWM. In addition, we use a small transformer network for $q_\psi(k \mid \mathbf{w})$ that supports variable-length sequences and is permutation-invariant over its input tokens. We consider ViT-Tiny encoders with maximum latent dimensions $d_{\max} \in \{96, 192\}$ and a ViT-Small encoder with $d_{\max} = 384$. Table 1 reports the test success rates of ALeWM and two LeWM variants: one with the latent dimension set to $d_{\max}$ and one with the best latent dimension searched as a hyper-parameter. For ALeWM, we use a coarse capacity support for each $d_{\max}$; for example, when $d_{\max} = 192$, the capacity support is $\mathcal{K} = \{8, 16, 32, 64, 96, 128, 160, 192\}$. This coarser support reduces the categorical search space and allows each capacity increment to activate a nontrivial block of coordinates, making capacity allocation easier to optimize. For each test episode, planning uses the capacity with the highest selector probability, denoted by K_{plan}. Table 1 reports the average of this selected capacity over 200 test episodes within each seed and then over the 3 seeds. Full experimental details are in Appendix E.2.

Across all evaluated configurations, ALeWM achieves the highest mean success rate on all four tasks. Compared with full-width LeWM, it improves mean success by approximately $1.5 - 17$ percentage points while reducing average planning capacity by approximately $67 - 98\%$. Tuning LeWM's latent dimension narrows the performance gap, but ALeWM retains gains of approximately $0.3 - 7.2$ percentage points, with the largest improvement on OGBench-Cube, while using fewer planning dimensions in 11 of the 12 comparisons. These results suggest that a small planning state is only part of the design: how predictive information is learned and organized also matters. A wide embedding gives the encoder room to represent the observations, while prediction of the full next embedding encourages useful information to concentrate in a short prefix. Although the main goal of this paper is to concentrate information in a compact representation of a wide embedding vector, we also report the selected capacities per experiment for ALeWM in Table 4 in Appendix E.2. The table shows that selected capacities tend to be similar within an experiment for all episodes although in some cases they can vary across episodes, most noticeably on Reacher. We believe that this is because the environments are highly controlled with low-variation between episodes. In Appendix D.3 we train ALeWM on mixed datasets and show that it learns different capacity distributions per dataset, albeit with performance degradation. In addition, in Appendix D.4 we analyze the capacity allocation in Reacher, showing that when the goal is further away from the starting position lower capacities are preferred. We also found that capacities can vary across seeds, but for each evaluated dataset and backbone, selections remain confined to neighboring supported values. This variation may arise from the coarse capacity grid and training stochasticity.

4.3 Analysis & Ablation Study

Training with fixed capacity probabilities. In our study, the model learns the capacity allocation during training. A natural question is how a model trained with a fixed capacity distribution would perform. This approach is reminiscent of nested dropout Rippel et al. (2014) and Matryoshka learning Kusupati et al. (2022). We refer to this approach as Nested Dropout. We compare the methods using ViT-Tiny with $d_{\max} = 192$. Nested Dropout uses the same capacity support, optimization settings, and relevant hyper-parameters as ALeWM, but samples from a fixed polynomial capacity distribution during training and uses the SIGReg regularization on active prefix only. After training, we evaluate planning separately at each capacity in $\mathcal{K}$ and report the mean success rate of the best configuration per capacity in Figure 3. On PushT, Nested Dropout success rate im-

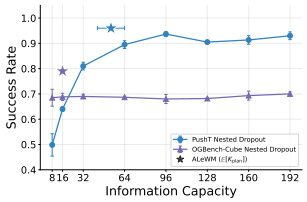


Figure 3: Test success rate for Nested Dropout vs ALeWM.

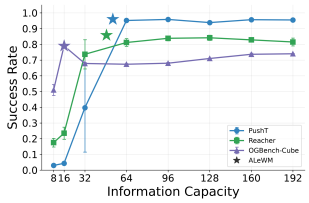


Figure 4: Test success rate per fixed prefix size.

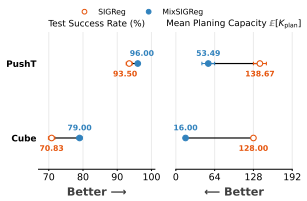


Figure 5: ALeWM with MixSIGReg vs with SIGReg

proves strongly with active dimension and peaks at $93.67 \pm 0.93\%$ using 96 dimensions; ALeWM reaches $96.00 \pm 0.00\%$ success rate while using a shorter prefix of 53.49 ± 10.51. In OGBench-Cube, Nested Dropout results remain between 68.00% and 70.00% across the capacity range, while ALeWM reaches $79.00 \pm 0.76\%$ with active prefix of only 16.00. Overall, this comparison favors the complete ALeWM training design over fixed-prior prefix training at compact planning capacities. Furthermore, Appendix D.2 presents the dynamics of capacity allocation during training showing non-trivial distribution shift as training progresses indicating on learning better capacity allocation.

ALeWM capacity selection criteria. To isolate the role of capacity at test time, we re-evaluate $d_{\max} = 192$ ALeWM models, overriding the selector with each of the 8 supported fixed prefix sizes. No retraining is performed: the model, planning procedure, and active-prefix goal cost in Eq. 11 are held fixed, and only the number of coordinates propagated through the rollout (including the goal state) is varied between experiments. We highlight that, like for Nested Dropout, this strategy requires evaluating each supported capacity separately to pick the best one based on the validation set, while the selection of ALeWM uses the selector network to choose a capacity per episode. Figure 4 compares the success rate of ALeWM with those obtained at fixed prefix sizes, averaged over three seeds. From the figure, no fixed prefix is uniformly best. Moreover, while for OGBench-Cube ALeWM's selection criterion matches the best fixed-prefix mean of 79.00% at $k = 16$, for PushT and Reacher the selected capacity has more variation. ALeWM achieves the success rate 96.00% in PushT and the success rate 85.83% in Reacher, matching or exceeding the success rates of 95.83% in $k = 96$ and 84.17% in $k = 128$ for those datasets, respectively, with smaller average prefix sizes.

ALeWM with SIGReg. To verify the contribution of our proposed prior-weighted mixture regularizer (MixSIGReg) we compare ALeWM with MixSIGReg vs ALeWM with SIGReg on masked embeddings using the ViT-Tiny backbone and $d_{\max} = 192$. For each regularizer and dataset, Figure 5 reports the mean test success rate and capacity of the best configuration, with SEM over 3 seeds. From the figure, MixSIGReg improves the mean test success rate by 2.50 percentage points in PushT and 8.17 points in OGBench-Cube, while reducing the mean selected prefix length from 138.67 to 53.49 and from 128.00 to 16.00, respectively. These results corroborate the importance of MixSIGReg both in terms of accuracy and capacity allocation.

5 Conclusion

This paper presents ALeWM, a method for adaptively selecting the active prefix size in world models. ALeWM introduces two components: a selector network that learns a probability distribution over active prefixes, and MixSIGReg, a regularizer for embeddings with varying prefix sizes that encourages a decreasing variance profile. Together, they encourage predictive information to concentrate in early coordinates through a self-supervised objective. Planning uses only the active prefixes using the largest selector probability. Across the evaluated benchmarks, ALeWM achieves higher mean success with compact planning prefixes, with the clearest gains on TwoRoom and OGBench-Cube. Several directions remain for improving the method. First, the capacity network is learned jointly with the world model, so its behavior can vary with initialization and other sources of randomness. Second, the MixSIGReg prior depends on a polynomial over capacity ranks and its degree. Alternatives that account for capacity values or avoid strict variance drops at capacity boundaries could improve the learned world model and potentially advance identifiability. Third, ALeWM uses the straight-through Gumbel–Softmax estimator, whose gradients are biased despite its practical effectiveness; exploring unbiased alternatives is a promising direction.

AI USE STATEMENT

In this work, we used generative AI tools to assist with developing the conceptual and theoretical framework; formulating and refining hypotheses and mathematical claims; deriving proof ideas and drafting portions of proofs; designing, implementing, and evaluating methods and experiments; developing and running the synthetic toy-data pipeline; implementing deterministic data preparation and episode-splitting code; and interpreting experimental results. We did not use generative AI tools for translation, and qualitative or thematic data analysis was not applicable to this work. Additionally, we used generative AI tools for research brainstorming, literature search and discovery, analysis of related work, suggesting experimental configurations, creating and editing software and figures, drafting and restructuring parts of the manuscript, and improving its readability.

All AI-assisted work was reviewed before inclusion. Mathematical arguments were checked against their stated assumptions and revised or removed when gaps or counterexamples were identified. Code changes were exercised through targeted tests and experiment runs, reported results were checked against saved run logs and artifacts, and literature suggestions were checked against the original sources. We take responsibility for the final content of this work, including text, claims, code, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses synthetically generated oscillator data and offline trajectories from simulated control environments. It does not involve human participants, personal or sensitive data, or deployment in real-world systems. Nevertheless, methods for learned world models and planning could contribute to future autonomous systems, where modeling errors and distribution shift may have safety consequences. The simulation results reported here should therefore not be interpreted as evidence of safety or readiness for real-world deployment. To support research integrity, we explicitly report experimental selection procedures and discuss limitations of this work.

REPRODUCIBILITY STATEMENT

Section 3 defines the model, training objective, and planning procedure. Section 3.3 summarizes the theoretical results, while Appendix C states their assumptions and provides complete proofs. Section 4.2 describes the evaluation protocol, and Appendix E records the synthetic-data construction, dataset preprocessing and episode-disjoint splits, model architectures, optimization settings, hyperparameter search spaces and selected configurations, random seeds, metrics, and planning settings. We report results over multiple seeds where stated. Code is attached to the submission.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Mahmoud 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/2506.09985v1. Version 1.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: Resampling images into 1D token sequences of flexible length. In *Proceedings of the 42nd International Conference on*

Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, pp. 2241–2292. PMLR, 2025. URL https://proceedings.mlr.press/v267/bachmann25a.html.

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

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=QaCCuDfBk2. Featured Certification.

Yufei Cui, Yu Mao, Ziquan Liu, Qiao Li, Antoni B. Chan, Xue Liu, Tei-Wei Kuo, and Chun Jason Xue. Variational nested dropout. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(8):10519–10534, 2023. doi: 10.1109/TPAMI.2023.3241945. URL https://doi.org/10.1109/TPAMI.2023.3241945.

T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.1093/biomet/70.3.723. URL https://academic.oup.com/biomet/article-abstract/70/3/723/248040.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3015–3024. PMLR, 2021. URL https://proceedings.mlr.press/v139/ermolov21a.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, April 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Matthew Ho, Xiaosheng Zhao, and Benjamin D. Wandelt. Ordered embeddings and intrinsic dimensionalities with information-ordered bottlenecks. *Machine Learning: Science and Technology*, 6 (3):035010, 2025. doi: 10.1088/2632-2153/ade94d. URL https://iopscience.iop.org/article/10.1088/2632-2153/ade94d.

HunyuanWorld Team, Zhenwei Wang, Yuhao Liu, Junta Wu, Zixiao Gu, Haoyuan Wang, Xuhui Zuo, Tianyu Huang, Wenhuan Li, Sheng Zhang, Yihang Lian, Yulin Tsai, Lifu Wang, Sicong Liu, Puhua Jiang, Xianghui Yang, Dongyuan Guo, Yixuan Tang, Xinyue Mao, Jiaao Yu, Junlin Yu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Chao Zhang, Yonghao Tan, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Minghui Chen, Zhan Li, Wangchen Qin, Lei Wang, Yifu Sun, Lin Niu, Xiang Yuan, Xiaofeng Yang, Yingping He, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Tian Liu, Peng Chen, Di Wang, Yuhong Liu, Linus, Jie Jiang, Tengfei Wang, and Chunchao Guo. HunyuanWorld 1.0: Generating immersive, explorable, and interactive 3D worlds from words or pixels, 2025. URL https://arxiv.org/abs/2507.21809v2. Version 2.

Petr Ivashkov, Randall Balestriero, and Bernhard Schölkopf. Sensorimotor world models: Perception for action via inverse dynamics, 2026. URL https://arxiv.org/abs/2606.20104v1. Version 1.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos Policy: Fine-tuning video models for visuomotor control and planning. In

C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 71531–71552, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/748becc400a57c0e31cfe6a2e7951467-Abstract-Conference.html.

David Klindt, Yann LeCun, and Randall Balestriero. When does LeJEPA learn a world model?, 2026. URL https://arxiv.org/abs/2605.26379. Preprint.

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

Yilun Kuang, Yash Dagade, Tim G. J. Rudner, Randall Balestriero, and Yann LeCun. Rectified LpJEPA: Joint-embedding predictive architectures with sparse and maximum-entropy representations. In *International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=CJ7aYueFJ3.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30233–30249, 2022. doi: 10.52202/068431-2192. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html.

Yann LeCun. A path towards autonomous machine intelligence, 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. OpenReview preprint, version 0.9.2.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: RC-aux for latent world models, 2026b. URL https://arxiv.org/abs/2605.07278. Preprint.

Chaoqi Liu, Yue Zhao, Haonan Chen, Xiaoshen Han, Jiawei Gao, Ehsan Adeli, and Yilun Du. Ordered action tokens for visuomotor policy learning, 2026. URL https://arxiv.org/abs/2607.21670v1. Version 1.

Xiaocheng Lu, Yuxi Chen, Jie Zhang, Jian Liu, Jingcai Guo, Fangqi Zhu, Tao Han, and Song Guo. AdaTok: Self-budgeting image tokenization with quality-preserving dynamic tokens, 2026. URL https://arxiv.org/abs/2606.07185v1. Version 1.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels, 2026. URL https://arxiv.org/abs/2603.19312. Preprint.

Dang Hong Nguyen, Nhi Ngoc-Yen Nguyen, and Huy-Hieu Pham. MIC: Maximizing informational capacity in adaptive representations via isotropic subspace alignment, 2026a. URL https://arxiv.org/abs/2605.29987v1. Version 1.

Hoang Nguyen, Xiaohao Xu, and Xiaonan Huang. Latent geometry beyond search: Amortizing planning in world models, 2026b. URL https://arxiv.org/abs/2605.08732. Preprint.

NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Yen-Chen Lin, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical AI, 2025. URL https://arxiv.org/abs/2501.03575.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1746–1754, Beijing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/rippel14.html.

Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Information Science and Statistics. Springer New York, 1 edition, 2004. doi: 10.1007/978-1-4757-4321-0. URL https://link.springer.com/book/10.1007/978-1-4757-4321-0.

Anirban Samaddar, Sandeep Madireddy, Prasanna Balaprakash, Tapabrata Maiti, Gustavo de los Campos, and Ian Fischer. Sparsity-inducing categorical prior improves robustness of the information bottleneck. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10207–10222. PMLR, 2023. URL https://proceedings.mlr.press/v206/samaddar23a.html.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim GJ Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *Advances in Neural Information Processing Systems*, 38:43905–43941, 2026.

Vlad Sobal, Jyothir S V, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint embedding predictive architectures focus on slow features, 2022. URL https://arxiv.org/abs/2211.10831. Presented at the NeurIPS 2022 Workshop on Self-Supervised Learning: Theory and Practice (nonarchival).

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training, 2026. URL https://arxiv.org/abs/2606.02572. Preprint.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. ElasticTok: Adaptive tokenization for image and video. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 38036–38056, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5e6cec2a9520708381fe520246018e8b-Abstract-Conference.html.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643. IEEE, June 2025. doi: 10.1109/CVPR52734.2025.00994. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_Thinking_in_Space_How_Multimodal_Large_Language_Models_See_Remember_CVPR_2025_paper.html.

Minghao Yang, Zechen Bai, Jing Lin, Haoqian Wang, and Alex Jinpeng Wang. VaporTok: RL-driven adaptive video tokenizer with prior & task awareness. *Advances in Neural Information Processing Systems*, 38:47459–47489, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Naveed Malik, Kyungmin Lee, William Liang, Nadun Ranawaka Arachchige, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Danfei Xu, Yilun Du, Ryan Julian, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi Fan, and Joel Jang. World action models are zero-shot policies. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026. URL https://openreview.net/forum?id=cd33uUB609.

Hanzhe You, Yonggang Zhang, Maohao Ran, Zhiqin Yang, Zhenyuan Zhang, Wei Xue, Jun Song, Xinmei Tian, and Yike Guo. A control theory of predictability in latent world models, 2026. URL https://arxiv.org/abs/2607.10362. Preprint.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one D4RT at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7382–7392, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_Efficiently_Reconstructing_Dynamic_Scenes_One_D4RT_at_a_Time_CVPR_2026_paper.html.

Xiangteng Zhang, Yang Guan, Bo Zhang, Hongyang Li, Ya-Qin Zhang, and Shengbo Eben Li. On the identifiability of controlled world models, 2026b. URL https://arxiv.org/abs/2607.22430. Preprint.

Kai Zhao, Dongliang Nie, Yuchen Lin, Zhehan Luo, Yixiao Gu, Deng-Ping Fan, and Dan Zeng. Sub-JEPA: Subspace Gaussian regularization for stable end-to-end world models, 2026. URL https://arxiv.org/abs/2605.09241v1. Version 1.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: world models on pre-trained visual features enable zero-shot planning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Eric Zimmermann, Harley Wiltzer, Justin Szeto, David Alvarez-Melis, and Lester Mackey. Ker-JEPA: Kernel discrepancies for Euclidean self-supervised learning, 2025. URL https://arxiv.org/abs/2512.19605v1. Version 1.

A LeWM Background

Joint-embedding prediction. Given observations $\mathbf{o}_t, \mathbf{o}_{t+1}$ and action $\mathbf{a}_t$, LeWM (Maes et al., 2026) forms:

$$\mathbf{s}_t = \mathrm{enc}_\theta(\mathbf{o}_t), \qquad \mathbf{s}_{t+1} = \mathrm{enc}_\theta(\mathbf{o}_{t+1}), \qquad \hat{\mathbf{s}}_{t+1} = \mathrm{pred}_\phi(\mathbf{s}_t, \mathbf{a}_t). \tag{12}$$

Up to empirical averaging and scalar loss weights, the LeWM objective is:

$$\mathcal{L}_{\mathrm{LeWM}} = \|\hat{\mathbf{s}}_{t+1} - \mathbf{s}_{t+1}\|_2^2 + \lambda\ \mathrm{SIGReg}(\mathbf{S}), \tag{13}$$

where $\mathbf{S}$ is the batch/history tensor of full embeddings. In practice, LeWM uses a window of preceding embeddings and actions rather than only the latest pair as shown in Eq. 12.

SIGReg regularization. Let $\mathbf{S} \in \mathbb{R}^{F\times B\times d}$ contain F frame positions for each of B sequences. For P projection directions $\mathbf{u}^{(p)} \in \mathbb{S}^{d-1}$, SIGReg (Balestriero & LeCun, 2025) computes the empirical characteristic function separately at each frame position t,

$$\hat{\phi}_t^{(p)}(\tau) = \frac{1}{B}\sum_{b=1}^{B} \exp\Big(i\tau \mathbf{u}^{(p)\top}\mathbf{s}_{b,t}\Big), \tag{14}$$

compares it with the standard-normal characteristic function $\phi_{\mathcal{N}}(\tau) = \exp(-\tau^2/2)$ through the weighted Epps–Pulley statistic,

$$EP_t^{(p)} = B\int w(\tau)\left|\hat{\phi}_t^{(p)}(\tau) - \phi_{\mathcal{N}}(\tau)\right|^2 d\tau, \tag{15}$$

averaged over p and t. The batch-size factor B is included in both SIGReg and MixSIGReg. Both use the same frequency window and quadrature, specified in Appendix E.2.

Latent planning. Planning in LeWM optimizes candidate action sequences by rolling out the (trained) predictor in latent space based on the encoding of the initial and goal observations $\mathbf{o}_1$ and $\mathbf{o}_g$, respectively:

$$\hat{\mathbf{s}}_1 = \mathrm{enc}_\theta(\mathbf{o}_1), \qquad \hat{\mathbf{s}}_{t+1} = \mathrm{pred}_\phi(\hat{\mathbf{s}}_t, \mathbf{a}_t), \qquad \arg\min_{\mathbf{a}_{1:H-1}} \|\hat{\mathbf{s}}_H - \mathrm{enc}_\theta(\mathbf{o}_g)\|_2^2. \tag{16}$$

Here H is the planning horizon. Candidate actions are optimized using the Cross-Entropy Method (CEM) (Rubinstein & Kroese, 2004).

B Straight-through prefix sampling

Define $q_c \coloneqq q_\psi(k_c|\mathbf{w})$ an element in the vector of probabilities under $\mathbf{q} \in \Delta^{C-1}$. In addition, define a matrix of masks $\mathbf{M} \in \{0,1\}^{C\times d}$, with row c equal to $\mathbf{m}(k_c)^\top$. To sample a prefix mask we apply the following steps:

1. Draw C independent Gumbel variables $g_c = -\log[-\log(U_c)]$, with $U_c \sim \mathrm{Uniform}(0,1)$, and collect them in $\mathbf{g} = (g_1, \ldots, g_C)$.
2. Produce a Gumbel-Softmax sample:

$$\boldsymbol{\chi}_{\mathrm{soft}} = \mathrm{softmax}\left(\frac{\log\mathbf{q} + \mathbf{g}}{\eta}\right), \qquad c^* = \arg\max_c \chi^c_{\mathrm{soft}} = \arg\max_c(\log q_c + g_c). \tag{17}$$

   where $\eta > 0$ is the temperature set to 0.5 throughout.
3. Construct $\boldsymbol{\chi}_{\mathrm{hard}} = \mathrm{onehot}(c^*) \in \{0,1\}^C$.
4. Use the straight-through vector $\boldsymbol{\chi} = \boldsymbol{\chi}_{\mathrm{hard}} - \mathrm{stopgrad}(\boldsymbol{\chi}_{\mathrm{soft}}) + \boldsymbol{\chi}_{\mathrm{soft}}$.
5. Form the prefix mask $\mathbf{m} = \mathbf{M}^\top\boldsymbol{\chi}$.

C THEORETICAL CLAIMS

We next formalize the two mechanisms induced by the capacity-matched target and nested prediction. We distinguish the corresponding prior and selector survival probabilities throughout.

Let $K_\pi \sim \pi_0(\cdot;\alpha)$ and let $\boldsymbol{\epsilon}_0 \sim \mathcal{N}(0, \mathbf{I}_d)$ be independent. The random variable $\mathbf{z}_0 = \mathbf{M}_{K_\pi}\boldsymbol{\epsilon}_0$ has distribution p_0 from Eq. 5. Define the prior survival probability

$$\rho_\pi^j = \Pr(K_\pi \geq j) = \sum_{\substack{k\in\mathcal{K}\\k\geq j}} \pi_0(k;\alpha), \qquad j = 1,\dots,d. \tag{18}$$

Proposition 1 (Ordered aggregate variance)**.** *The capacity-matched target satisfies*

$$\mathbb{E}[\mathbf{z}_0] = 0, \qquad \mathrm{Cov}(\mathbf{z}_0) = \mathrm{diag}(\rho_\pi^1,\dots,\rho_\pi^d), \qquad \rho_\pi^1 \geq \cdots \geq \rho_\pi^d. \tag{19}$$

Moreover,

$$\mathrm{tr}(\mathrm{Cov}(\mathbf{z}_0)) = \sum_{j=1}^{d} \rho_\pi^j = \mathbb{E}[K_\pi], \qquad \rho_\pi^j - \rho_\pi^{j+1} = \pi_0(j;\alpha), \tag{20}$$

where $\rho_\pi^{d+1} = 0$ and $\pi_0(j;\alpha) = 0$ when $j \notin \mathcal{K}$. Hence, the target variance is non-increasing and drops only at supported capacity boundaries; it is constant within every intervening coordinate block.

Proof. Conditional on $K_\pi = k$, the target has mean zero and covariance $\mathbf{M}_k$. The law of total covariance therefore gives

$$\mathrm{Cov}(\mathbf{z}_0) = \sum_{k\in\mathcal{K}} \pi_0(k;\alpha)\mathbf{M}_k. \tag{21}$$

The jth diagonal entry equals $\sum_{k\geq j}\pi_0(k;\alpha) = \rho_\pi^j$, while every off-diagonal entry is zero. The events $\{K_\pi \geq j+1\} \subseteq \{K_\pi \geq j\}$ imply monotonicity, and their probability difference is $\Pr(K_\pi = j) = \pi_0(j;\alpha)$. Finally, exchanging the order of two finite sums yields

$$\sum_{j=1}^{d}\rho_\pi^j = \sum_{k\in\mathcal{K}}\pi_0(k;\alpha)\sum_{j=1}^{d}\mathbb{1}[j\leq k] = \sum_{k\in\mathcal{K}} k\pi_0(k;\alpha) = \mathbb{E}[K_\pi]. \tag{22}$$

□

Proposition 1 says that the capacity prior sets a variance budget for the masked reference distribution. Earlier coordinates receive at least as much of this budget because every cutoff that retains a later coordinate also retains the earlier ones, and the total budget equals the prior's expected capacity. Coordinates in the same capacity block receive the same budget, which drops only where the capacity support permits a cutoff. This statement concerns only the reference distribution; however by optimizing MixSIGReg the learned selector distribution is encouraged to maintain that structure.

We next separate this prior target from the distribution induced by the learned selector. Let the context $\mathbf{w}$ follow the data distribution, draw $K \mid \mathbf{w} \sim q_\psi(\cdot \mid \mathbf{w})$, and let $P_{\theta,\psi,t}$ denote the resulting population distribution of $\mathbf{M}_K\mathbf{s}_t$. Unless stated otherwise, expectations involving K, $\mathbf{w}$, and $\mathbf{s}_t$ are under this joint law. Define the conditional and aggregate selector survival probabilities:

$$\rho_q^j(\mathbf{w}) = \Pr_{q_\psi}(K \geq j \mid \mathbf{w}) = \sum_{\substack{k\in\mathcal{K}\\k\geq j}} q_\psi(k \mid \mathbf{w}), \qquad \bar{\rho}_q^j = \mathbb{E}_{\mathbf{w}}[\rho_q^j(\mathbf{w})]. \tag{23}$$

To state what exact distribution matching transfers to the learned masked latent, let ω be a finite Borel measure whose support is all of $\mathbb{R}^d$, and define the idealized full-frequency characteristic-function discrepancy. Here $\phi_{\theta,\psi,t}$ and ϕ_0 denote the characteristic functions of $P_{\theta,\psi,t}$ and p_0, respectively:

$$D_\omega^2(P_{\theta,\psi,t}, p_0) = \int_{\mathbb{R}^d} |\phi_{\theta,\psi,t}(\boldsymbol{\xi}) - \phi_0(\boldsymbol{\xi})|^2 \, d\omega(\boldsymbol{\xi}). \tag{24}$$

Proposition 2 (Population target-law matching)**.** *For ω as defined above,*

$$D_\omega^2(P_{\theta,\psi,t}, p_0) = 0 \quad \Longleftrightarrow \quad P_{\theta,\psi,t} = p_0. \tag{25}$$

Proof. The reverse implication is immediate. For the forward implication, $D_\omega^2 = 0$ gives $\phi_{\theta,\psi,t}(\boldsymbol{\xi}) = \phi_0(\boldsymbol{\xi})$ for ω-almost every $\boldsymbol{\xi}$. Characteristic functions are continuous. If they differed at one point, continuity would give a nonempty open neighborhood on which their difference remained nonzero, contradicting the full support of ω. They therefore agree on all of $\mathbb{R}^d$, and uniqueness of characteristic functions gives $P_{\theta,\psi,t} = p_0$. □

Corollary 3 (Learned masked-variance profile)**.** *Suppose $D_\omega^2(P_{\theta,\psi,t}, p_0) = 0$. Define the sequence-level activation indicator $m_K^j := [\mathbf{m}(K)]^j = \mathbb{1}[K \geq j]$, so that $z_t^j = m_K^j s_t^j$. Then, for every coordinate j,*

$$\mathbb{E}[m_K^j s_t^j] = 0, \qquad \mathrm{Var}(z_t^j) = \mathbb{E}[m_K^j (s_t^j)^2] = \rho_\pi^j. \tag{26}$$

Consequently,

$$\mathrm{Var}(z_t^1) \geq \cdots \geq \mathrm{Var}(z_t^d), \qquad \sum_{j=1}^{d} \mathrm{Var}(z_t^j) = \mathbb{E}[K_\pi]. \tag{27}$$

Thus, the learned masked aggregate *has the non-increasing target variance profile. Moreover, if $\bar{\rho}_q^j > 0$, then*

$$\mathbb{E}[s_t^j \mid m_K^j = 1] = 0, \qquad \mathrm{Var}(s_t^j \mid m_K^j = 1) = \frac{\rho_\pi^j}{\bar{\rho}_q^j}. \tag{28}$$

If $\rho_\pi^j > 0$, exact matching necessarily implies $\bar{\rho}_q^j > 0$. To relate this result to the unmasked encoder coordinate, suppose additionally that $\mathbb{E}[(s_t^j)^2] < \infty$. When $\bar{\rho}_q^j < 1$, define

$$\mu_0^j = \mathbb{E}[s_t^j \mid m_K^j = 0], \qquad v_0^j = \mathrm{Var}(s_t^j \mid m_K^j = 0). \tag{29}$$

Then the three possible activation regimes give

$$\begin{aligned} \mathrm{Var}(s_t^j) &= \begin{cases} v_0^j, & \bar{\rho}_q^j = 0 \quad (\rho_\pi^j = 0), \\ \rho_\pi^j + (1 - \bar{\rho}_q^j) v_0^j + \bar{\rho}_q^j (1 - \bar{\rho}_q^j)(\mu_0^j)^2, & 0 < \bar{\rho}_q^j < 1, \\ \rho_\pi^j, & \bar{\rho}_q^j = 1, \end{cases} \\ &\geq \rho_\pi^j. \end{aligned} \tag{30}$$

Proof. Proposition 2 gives $P_{\theta,\psi,t} = p_0$, so Proposition 1 gives $\mathbb{E}[z_t^j] = 0$, $\mathbb{E}[(z_t^j)^2] = \rho_\pi^j$, and the stated total variance. Since $z_t^j = m_K^j s_t^j$ and $(m_K^j)^2 = m_K^j$, these are exactly the first two moment identities in Eq. 26.

All learned-side expectations below are under the joint law specified above. By iterated expectation,

$$\begin{aligned} \Pr(m_K^j = 1) &= \mathbb{E}_{\mathbf{w}}\big[\mathbb{E}_{K|\mathbf{w}}[\mathbb{1}[K \geq j]]\big] \\ &= \mathbb{E}_{\mathbf{w}}\left[\sum_{k \in \mathcal{K}} q_\psi(k \mid \mathbf{w}) \mathbb{1}[k \geq j]\right] \\ &= \mathbb{E}_{\mathbf{w}}[\rho_q^j(\mathbf{w})] = \bar{\rho}_q^j. \end{aligned}$$

For $r \in \{1, 2\}$ and $\bar{\rho}_q^j > 0$, the defining indicator identity for conditional expectation, followed by iterated expectation, gives the full calculation

$$\begin{aligned}
\mathbb{E}[(s_t^j)^r \mid m_K^j = 1] &= \frac{\mathbb{E}[\mathbb{1}[m_K^j = 1](s_t^j)^r]}{\Pr(m_K^j = 1)} \\
&= \frac{1}{\bar{\rho}_q^j} \mathbb{E}_{\mathbf{w}}\left[\mathbb{E}_{K|\mathbf{w}}\left[\mathbb{1}[K \geq j](s_t^j)^r\right]\right] \\
&= \frac{1}{\bar{\rho}_q^j} \mathbb{E}_{\mathbf{w}}\left[\sum_{k \in \mathcal{K}} q_\psi(k \mid \mathbf{w}) \mathbb{1}[k \geq j](s_t^j)^r\right] \\
&= \frac{1}{\bar{\rho}_q^j} \mathbb{E}_{\mathbf{w}}[\rho_q^j(\mathbf{w})(s_t^j)^r] \\
&= \frac{\mathbb{E}[m_K^j (s_t^j)^r]}{\bar{\rho}_q^j}.
\end{aligned}$$

The third equality uses that s_t^j is fixed once $\mathbf{w}$ is fixed. This is the conditional-expectation form of Bayes' rule; under a density formulation, the factor $\rho_q^j(\mathbf{w})/\bar{\rho}_q^j$ is precisely the likelihood-ratio weight that changes the data law to the law conditioned on $m_K^j = 1$. Taking $r = 1$ and $r = 2$ and using the first two moment identities gives

$$\mathbb{E}[s_t^j \mid m_K^j = 1] = 0, \qquad \mathbb{E}[(s_t^j)^2 \mid m_K^j = 1] = \frac{\rho_\pi^j}{\bar{\rho}_q^j}.$$

The conditional mean is zero, so the conditional second moment equals the conditional variance, proving Eq. 28. Finally, if $\bar{\rho}_q^j = 0$, then $z_t^j = 0$ almost surely and hence has zero variance; this contradicts $\rho_\pi^j > 0$ under exact matching. For $0 < \bar{\rho}_q^j < 1$, the law of total variance conditional on m_K^j gives

$$\begin{aligned}
\mathrm{Var}(s_t^j) &= \mathbb{E}\Big[\mathrm{Var}(s_t^j \mid m_K^j)\Big] + \mathrm{Var}\Big(\mathbb{E}[s_t^j \mid m_K^j]\Big), \\
\mathbb{E}\Big[\mathrm{Var}(s_t^j \mid m_K^j)\Big] &= \bar{\rho}_q^j \mathrm{Var}(s_t^j \mid m_K^j = 1) + (1 - \bar{\rho}_q^j) v_0^j.
\end{aligned}$$

For the second term, the active conditional mean is zero and the inactive conditional mean is μ_0^j, so

$$\begin{aligned}
\mathrm{Var}\Big(\mathbb{E}[s_t^j \mid m_K^j]\Big) &= \mathbb{E}\Big[\big(\mathbb{E}[s_t^j \mid m_K^j]\big)^2\Big] - \mathbb{E}\Big[\mathbb{E}[s_t^j \mid m_K^j]\Big]^2 \\
&= \bar{\rho}_q^j 0^2 + (1 - \bar{\rho}_q^j)(\mu_0^j)^2 \\
&\quad - \big(\bar{\rho}_q^j 0 + (1 - \bar{\rho}_q^j)\mu_0^j\big)^2 \\
&= \bar{\rho}_q^j (1 - \bar{\rho}_q^j)(\mu_0^j)^2.
\end{aligned}$$

Combining these terms and substituting Eq. 28 gives the middle case of Eq. 30. If $\bar{\rho}_q^j = 0$, then $m_K^j = 0$ almost surely, so $\mathrm{Var}(s_t^j) = v_0^j$, while exact matching gives $\rho_\pi^j = \mathrm{Var}(z_t^j) = 0$. If $\bar{\rho}_q^j = 1$, then $m_K^j = 1$ almost surely, so $s_t^j = z_t^j$ and $\mathrm{Var}(s_t^j) = \rho_\pi^j$. These observations also prove the lower bound in Eq. 30. □

Intuition. Equation 28 shows how activation frequency and active-coordinate scale can compensate for one another: for fixed target variance ρ_π^j, increasing $\bar{\rho}_q^j$ decreases the required conditional variance when the coordinate is active, whereas decreasing $\bar{\rho}_q^j$ increases it. Thus the equation fixes only the product $\bar{\rho}_q^j \mathrm{Var}(s_t^j \mid m_K^j = 1) = \rho_\pi^j$, not either factor separately. Equations 29– 30 also include the encoder values produced while the coordinate is inactive. Variation among those values, or a shift between their mean and the active mean, adds the two nonnegative terms in the middle case. These additions can differ across coordinates. In plain words, the masked coordinates z_t^j have an ordered variance profile, but the raw coordinates s_t^j need not inherit that order: a later raw coordinate may have larger variance than an earlier one. The raw variances may still happen to be ordered in a

learned representation; the target-law equality simply does not guarantee it. This freedom concerns the masked target-law constraint alone, and the shared model and prediction loss may still constrain inactive values. The identities also require exact population matching; a small finite MixSIGReg surrogate need not satisfy them.

We next quantify the predictive value of nested prefixes while holding the representation fixed. Fix the encoder and population data distribution, and consider a fixed prediction time t. Write $\mathbb{E}_{\mathrm{data}}$ for expectation under the induced joint law of $(\mathbf{s}_{1:t+1}, \mathbf{a}_{1:t})$, and assume $\mathbb{E}_{\mathrm{data}}\|\mathbf{s}_{t+1}\|_2^2 < \infty$. Write $\mathbf{s}_{1:t}^{1:k} = \{s_\tau^j : 1 \le \tau \le t,\ 1 \le j \le k\}$ for the first k coordinates of the latent history, and define

$$\mathcal{F}_k = \sigma(\mathbf{s}_{1:t}^{1:k}, \mathbf{a}_{1:t}), \qquad \mu_k = \mathbb{E}_{\mathrm{data}}[\mathbf{s}_{t+1} \mid \mathcal{F}_k], \qquad R_k = \mathbb{E}_{\mathrm{data}}\|\mathbf{s}_{t+1} - \mu_k\|_2^2, \tag{31}$$

where $\mathcal{F}_0 = \sigma(\mathbf{a}_{1:t})$. Here $\sigma(\cdot)$ denotes the sigma-algebra generated by its arguments: the collection of events whose occurrence can be determined from those random variables. Thus, $\mathcal{F}_k$ contains all information in the action history and the first k coordinates of the latent history, while $\mathcal{F}_0$ contains only the action history. The conditional mean μ_k is the Bayes-optimal squared-error prediction using that information. Accordingly, R_k is a *forced-capacity oracle risk*: it is the population squared error of a separate Bayes-optimal predictor at prefix k. It does not assume that the single finite predictor used in training attains these risks. We note that because the training selector can observe target frames, its mask can convey information absent from $\mathcal{F}_k$. Hence, these capacity risks are not automatically lower bounds for a target-dependent selection mechanism, but serve as a useful surrogate.

Proposition 4 (Survival-weighted predictive value)**.** *For $k \le \ell$,*

$$R_k - R_\ell = \mathbb{E}_{\mathrm{data}}\|\mu_\ell - \mu_k\|_2^2 \ge 0. \tag{32}$$

Set $k_0 = 0$ and define the predictive gain of capacity block c as

$$\Delta_c = R_{k_{c-1}} - R_{k_c} \ge 0. \tag{33}$$

If $K_\pi \sim \pi_0(\cdot;\alpha)$ is sampled independently of the data, then

$$\mathbb{E}_{K_\pi}[R_{K_\pi}] = R_0 - \sum_{c=1}^{C} \rho_\pi^{k_c} \Delta_c. \tag{34}$$

Here $\mathbb{E}_{K_\pi}$ averages only over the independent prior-capacity draw. Thus, the predictive gain of each block is weighted by the probability that the block survives the sampled capacity. If, additionally, every block permutation σ is feasible and yields prefix risks $R_{k_r}^{(\sigma)} = R_0 - \sum_{c=1}^{r} \Delta_{\sigma(c)}$ under the same capacity prior, then $\mathbb{E}_{K_\pi}[R_{K_\pi}^{(\sigma)}]$ is minimized by arranging the gains in non-increasing order. For the learned selector, let $\ell_{k_c}(\mathbf{w})$ denote any forced-capacity loss at context $\mathbf{w}$, let $\ell_0(\mathbf{w})$ denote the corresponding all-zero-prefix loss, and define $\delta_c(\mathbf{w}) = \ell_{k_{c-1}}(\mathbf{w}) - \ell_{k_c}(\mathbf{w})$. Then, context by context,

$$\mathbb{E}_{K \sim q_\psi(\cdot|\mathbf{w})}[\ell_K(\mathbf{w})] = \ell_0(\mathbf{w}) - \sum_{c=1}^{C} \rho_q^{k_c}(\mathbf{w}) \delta_c(\mathbf{w}). \tag{35}$$

The expectation in Eq. 35 is only over the categorical draw of K, with the context $\mathbf{w}$ held fixed. The increments $\delta_c(\mathbf{w})$ need not be positive for a shared finite predictor.

Proof. For $k \le \ell$, $\mathcal{F}_k \subseteq \mathcal{F}_\ell$. The space $L^2(\mathcal{F}_k; \mathbb{R}^d)$ is the closed subspace of square-integrable, $\mathcal{F}_k$-measurable random vectors, with inner product $\langle \mathbf{u}, \mathbf{v} \rangle_{L^2} = \mathbb{E}_{\mathrm{data}}[\mathbf{u}^\top \mathbf{v}]$. The conditional mean μ_k is the orthogonal projection of $\mathbf{s}_{t+1}$ onto this subspace. The projections are *nested* because the larger prefix gives the inclusion $L^2(\mathcal{F}_k; \mathbb{R}^d) \subseteq L^2(\mathcal{F}_\ell; \mathbb{R}^d)$. In particular, $\mu_\ell - \mu_k$ is $\mathcal{F}_\ell$-measurable, whereas the residual $\mathbf{s}_{t+1} - \mu_\ell$ is orthogonal to every vector in that larger subspace. Explicitly,

$$\begin{aligned}
&\mathbb{E}_{\mathrm{data}}\big[(\mathbf{s}_{t+1} - \mu_\ell)^\top (\mu_\ell - \mu_k)\big] \\
&\quad = \mathbb{E}_{\mathrm{data}}\big[\mathbb{E}_{\mathrm{data}}\big[(\mathbf{s}_{t+1} - \mu_\ell)^\top (\mu_\ell - \mu_k) \mid \mathcal{F}_\ell\big]\big] \\
&\quad = \mathbb{E}_{\mathrm{data}}\big[\mathbb{E}_{\mathrm{data}}[\mathbf{s}_{t+1} - \mu_\ell \mid \mathcal{F}_\ell]^\top (\mu_\ell - \mu_k)\big] = 0.
\end{aligned}$$

Using $\mathbf{s}_{t+1} - \mu_k = (\mathbf{s}_{t+1} - \mu_\ell) + (\mu_\ell - \mu_k)$ and expanding the squared norm now gives

$$\begin{aligned} R_k &= \mathbb{E}_{\text{data}}\|\mathbf{s}_{t+1} - \mu_\ell\|_2^2 + \mathbb{E}_{\text{data}}\|\mu_\ell - \mu_k\|_2^2 \\ &\quad + 2\mathbb{E}_{\text{data}}\left[(\mathbf{s}_{t+1} - \mu_\ell)^\top(\mu_\ell - \mu_k)\right] \\ &= R_\ell + \mathbb{E}_{\text{data}}\|\mu_\ell - \mu_k\|_2^2, \end{aligned}$$

which proves Eq. 32.

For a supported capacity k_r, telescoping gives $R_{k_r} = R_0 - \sum_{c=1}^{r} \Delta_c$. Since K_π has the finite support $\{k_1, \ldots, k_C\}$, taking expectation over the prior-capacity draw and exchanging the two finite sums gives the full calculation

$$\begin{aligned} \mathbb{E}_{K_\pi}[R_{K_\pi}] &= \sum_{r=1}^{C} \pi_0(k_r;\alpha) R_{k_r} \\ &= \sum_{r=1}^{C} \pi_0(k_r;\alpha) \left(R_0 - \sum_{c=1}^{r} \Delta_c\right) \\ &= R_0 - \sum_{c=1}^{C} \left(\sum_{r=c}^{C} \pi_0(k_r;\alpha)\right) \Delta_c \\ &= R_0 - \sum_{c=1}^{C} \Pr(K_\pi \geq k_c)\Delta_c. \end{aligned} \tag{36}$$

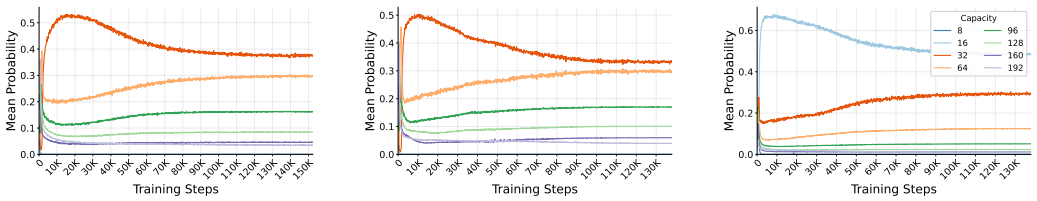
The last equality uses $\{K_\pi \geq k_c\} = \{K_\pi \in \{k_c, \ldots, k_C\}\}$. Since $\Pr(K_\pi \geq k_c) = \rho_\pi^{k_c}$, Eq. 36 is exactly Eq. 34. For $i < j$ with $\Delta_i < \Delta_j$, exchanging the two gains reduces the expected risk by $(\rho_\pi^{k_i} - \rho_\pi^{k_j})(\Delta_j - \Delta_i) \geq 0$; successive exchanges establish the ordering claim. For the final identity, the same telescope gives $\ell_{k_r}(\mathbf{w}) = \ell_0(\mathbf{w}) - \sum_{c=1}^{r} \delta_c(\mathbf{w})$. Taking expectation over $K \sim q_\psi(\cdot \mid \mathbf{w})$ and exchanging the two finite sums yields Eq. 35. □

Predictive-allocation interpretation. Proposition 4 explains why a component's position in the prefix matters. In the independent-prior case of Eq. 34, suppose each component has a fixed nonnegative gain, and these gains add without changing when components are moved or combined. Each position inherits the probability that its block survives the cutoff, so Proposition 4 weights each component's gain by that probability. A component placed earlier is retained at least as often; therefore, exchanging a larger late gain with a smaller early gain cannot increase expected prediction error, and strictly decreases it when their retention probabilities differ. Positions with the same retention probability remain unordered. For the learned selector, Eq. 35 provides the same accounting within each context, but the gains can interact, change when components move, or become negative, and one shared representation need not realize every context-specific ordering. This argument therefore explains the ordering pressure created by Proposition 4; it is not a convergence guarantee for joint training.

D Additional Experiments & Analysis

D.1 Rollout target: active vs full

Here, we test whether planning should compare the predicted (unmasked) state $\mathbf{s}_H$ with the full goal embedding ($\mathbf{s}_g$) or only with the selected prefix embedding by the capacity network according to Eq. 11. Both conditions keep the selected prefix fixed and mask all intermediate rollout states. For each setting, we re-evaluate ViT-Tiny ($d_{\max} = 192$) checkpoints without retraining. Table 2 shows that restricting the rollout objective to active dimensions improves the success rate for both OGBench-Cube and Reacher, with an unchanged mean success on PushT. This experiment supports our selection, providing an indication that the information important for accurate prediction of the goal state concentrates mostly at the prefix.



(a) PushT. (b) Reacher. (c) OGBench-Cube.

Figure 6: Mean capacity probability over training for the ViT-Tiny ALeWM models with $d_{\max} = 192$ for one seed. Each curve is the capacity probability averaged over the training batch at each logged step.

Table 2: Effect of the rollout target capacity size on test success rate (SR, %) with ViT-Tiny ALeWM having $d_{\max} = 192$. Results present test success rate (SR, %) $\pm$ SEM over 3 seeds.

Rollout target	PushT SR (↑)	Reacher SR (↑)	OGBench-Cube SR (↑)
Full Capacity (d_{max})	$\mathbf{96.00 \pm 0.58}$	83.50 ± 2.25	75.17 ± 0.44
Max Probability (K_{plan})	$\mathbf{96.00 \pm 0.00}$	$\mathbf{85.83 \pm 1.01}$	$\mathbf{79.00 \pm 0.76}$

D.2 Capacity dynamics

Figure 6 tracks the learned categorical capacity distribution during training for one ViT-Tiny run of ALeWM with $d_{\max} = 192$ for PushT, Reacher, and OGBench-Cube. The figure shows that in all three datasets, most redistribution occurs during the first few thousand optimization steps, after which the probabilities evolve more gradually, sometimes crossing one another. The converged mixtures are task dependent: PushT and Reacher assign the largest mean probability to $k = 32$, whereas OGBench-Cube assigns the largest probability to $k = 16$. OGBench-Cube also concentrates more strongly on the two smallest nonzero-probability capacities, $k = 16$ and $k = 32$. By contrast, PushT and Reacher retain more mass across $k \in \{32, 64, 96\}$. The probability of $k = 8$ approaches zero in every dataset. These trajectories show that the capacity selector does not merely preserve its initialization: it learns distinct distributions for the three control domains.

D.3 Mixed-dataset training: different capacities for different datasets

As a proof of concept, we test whether one capacity network can learn dataset-dependent capacities when a single model is trained jointly on DMC and TwoRoom. Figure 7 shows that the selector learns distinct distributions for the two datasets. DMC shifts toward a broad high-capacity distribution whose largest component is $k = 64$, while retaining substantial mass on $k \in \{96, 128, 160\}$. TwoRoom instead concentrates on the smaller $k \in \{16, 32\}$ prefixes. At the terminal test evaluation, DMC obtains $85.17 \pm 0.73\%$ success with mean selected test-time prefix length $\mathbb{E}[K_{\text{plan}}] = 63.95 \pm 0.05$, whereas TwoRoom obtains only $36.17 \pm 1.01\%$ success with $\mathbb{E}[K_{\text{plan}}] = 21.33 \pm 5.33$ (mean $\pm$ SEM over three seeds). Thus, the experiment provides a simple demonstration that the capacity network can assign different capacities to different data while maintaining strong DMC performance. However, TwoRoom performance degrades severely, so further work is required to preserve performance on both datasets during joint training.

D.4 Reacher Capacity Allocation

Figure 8 examines per-episode capacity allocation for ViT-Tiny ALeWM models with $d_{\max} = 192$, evaluated on the same 200 Reacher test episodes across three training seeds. For seeds 3072 and 3073, the selector tends to assign $K = 64$ to episodes with smaller physical start–goal gaps. As the distance between the start and goal encoder representations increases, the probability ratio $q_\psi(K = 64 \mid \mathbf{w})/q_\psi(K = 32 \mid \mathbf{w})$ tends to decrease, favoring $K = 32$. Seed 3074, however, selects $K = 64$ for all episodes, indicating that the allocation pattern depends on the training seed, although all selections remain within the two neighboring supported capacities, 32 and 64. Within each seed

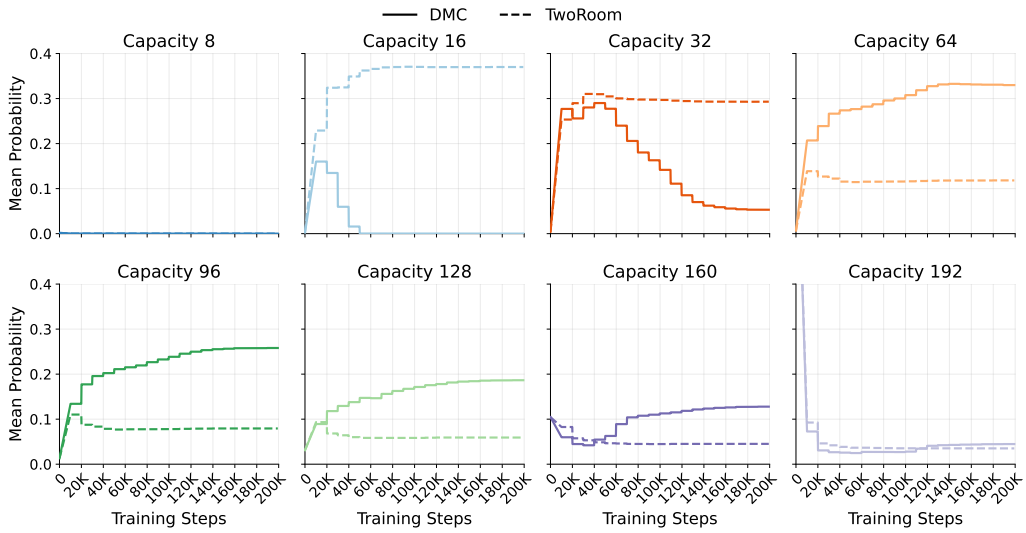


Figure 7: Dataset-conditioned capacity dynamics for joint DMC–TwoRoom training, averaged over three seeds at fixed steps. Each panel corresponds to one capacity; solid and dashed curves show the mean selector probability for DMC and TwoRoom, respectively. For better visibility, the vertical range is capped at 0.4, thereby omitting the initial probability 0.837 of capacity $k = 192$.

and selected-capacity group, failures also tend to be more frequent for episodes with larger physical start–goal arm-tip gaps.

D.5 Example Rollouts

Figure 9 compares two matched test episodes *randomly selected* per dataset for ViT-Tiny models trained with seed 3072, using $d_{\max} = 192$ for ALeWM and the best-d LeWM alternative. We select episodes on which ALeWM meets the environment's success criterion while LeWM does not, with both methods starting from the same state and targeting the same goal. These examples illustrate differences in goal-directed behavior between the two approaches.

E Implementation Details

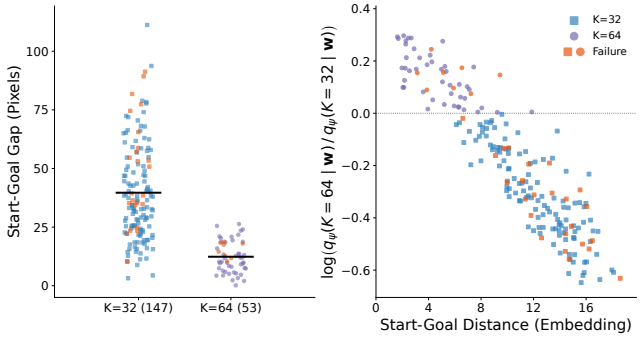
E.1 Toy Example

Problem and data. The state comprises two independent controlled damped oscillators, $\mathbf{x}_t = (p_t^1, v_t^1, p_t^2, v_t^2)$. For oscillator $m \in \{1, 2\}$, semi-implicit Euler integration gives

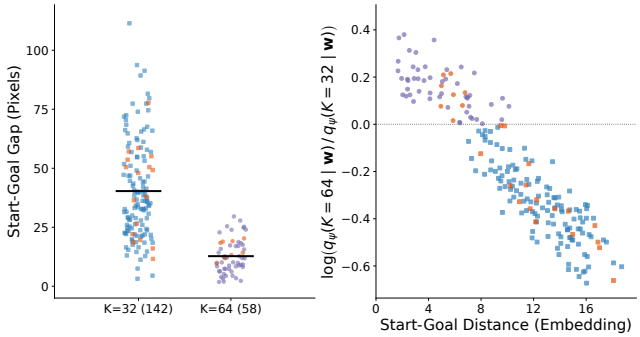
$$v_{t+1}^m = v_t^m + \Delta t[-\gamma v_t^m - \kappa(p_t^m - a_t^m)] + \epsilon_t^m, \qquad p_{t+1}^m = p_t^m + \Delta t\, v_{t+1}^m + \eta_t^m. \tag{37}$$

We set $\Delta t = 0.2$, $\gamma = 0.25$, $\kappa = 1.03887957$, $\epsilon_t^m \sim \mathcal{N}(0, 0.02730495^2)$, and $\eta_t^m \sim \mathcal{N}(0, 0.01365248^2)$. Each component of the action is sampled uniformly from $[-1.36524771, 1.36524771]$ and held for four transitions. We run 256 burn-in transitions, then record nine states and eight actions per trajectory. The time step and damping yield stable, underdamped dynamics. The non-round constants reflect a scale calibration that gives position and velocity approximately unit stationary variance under the four-step action-hold policy, while retaining small process noise. The burn-in reduces dependence on the zero initial state. The train, validation, and test splits contain 5000, 1000, and 1000 trajectories.

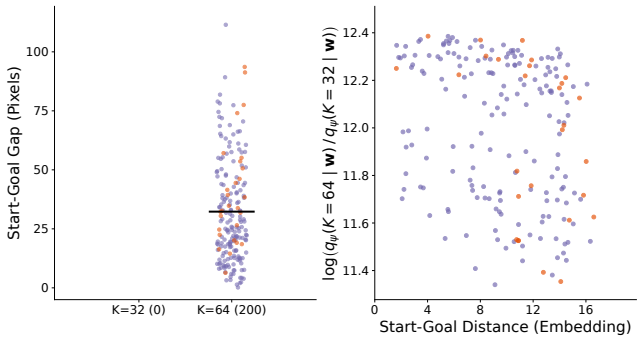
The ten-dimensional observation contains four strong coordinates produced by an invertible nonlinear wave map followed by additive noise, and six weak coordinates of the form $a_o \sin(\mathbf{w}_o^\top \mathbf{x}_t + \phi_o) + \epsilon_{t,o}$, for $o \in \{1, \ldots, 6\}$. The fixed dense vectors $\mathbf{w}_o$ mix all four factors and form a rank-four matrix when stacked as rows. The fixed parameters ϕ_o and a_o specify the phase shift and noiseless signal amplitude, respectively. Measurement noise is independent, zero-mean Gaussian across coordinates, time steps, and trajectories, with standard deviations 0.3 for weak coordinates and 0.05



(a) Training seed 3072



(b) Training seed 3073



(c) Training seed 3074

Figure 8: Reacher rollout capacity allocation for ViT-Tiny ALeWM with $d_{\max} = 192$. Left: Euclidean start–goal arm-tip displacement in pixels, grouped by selected capacity; black bars indicate means and parentheses give episode counts. Right: the logged selector log probability ratio versus Euclidean distance between the start and goal encoder CLS tokens. Squares denote $K = 32$ and circles $K = 64$; blue and purple indicate successful rollouts, respectively, and orange indicates failures.

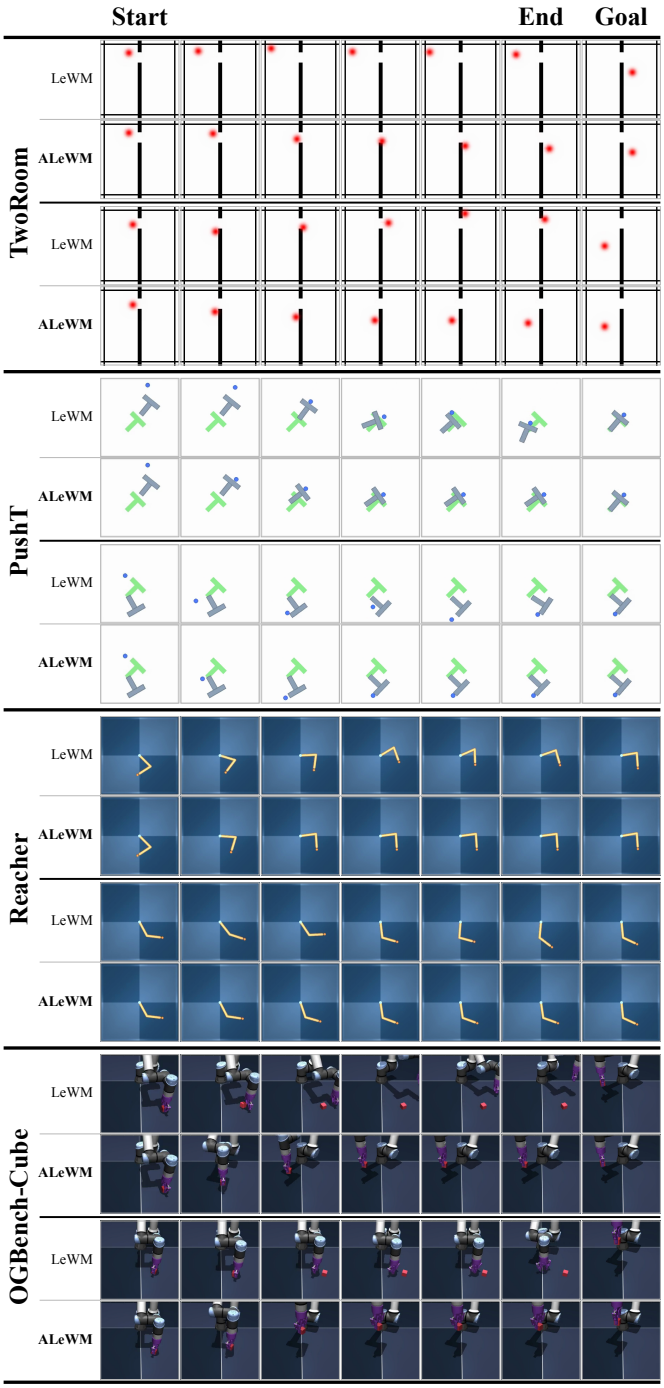


Figure 9: Qualitative comparison on two matched test episodes per dataset, selected for ALeWM success and LeWM failure. Each pair places LeWM above ALeWM. Columns show the initial observation, four approximately uniformly spaced intermediate observations, the final recorded observation, and the shared goal. Images are environment observations under model-planned actions.

for strong coordinates. The weak coordinates therefore add nonlinear redundant measurements, not independent dynamical factors.

Models and optimization. We use an encoder network with two hidden layers of width 64, while the embedding projector network, the predictor, and ALeWM's capacity network each have one hidden layer of width 64 with SiLU activations. ALeWM uses maximum dimension eight and supports every prefix $K = 1, \ldots, 8$; LeWM uses the fixed dimension specified by each baseline run. We train for 200 epochs with batch size 256 using AdamW, learning rate 10^{-3}, weight decay 10^{-4}, and gradient-norm clipping at 1.0. SIGReg uses 17 knots and 64 random projections. All data, initialization, sampling, SIGReg, and diagnostic random streams use fixed seeds for consistency and reproducibility.

Metrics. All reported metrics use the test split. For linear state recovery, a multivariate ridge probe with coefficient 10^{-6} is fitted on standardized training pairs for every prefix and time, $\mathbf{s}_t^{1:K}$, and applied unchanged to test pairs. We report the arithmetic mean of the four factor-wise coefficients of determination,

$$R_K^2 = \frac{1}{4}\sum_{r=1}^{4}\left(1 - \frac{\sum_n (x_{n,r} - \hat{x}_{n,r}^{(K)})^2}{\sum_n (x_{n,r} - \bar{x}_r)^2}\right), \tag{38}$$

where n indexes all test trajectory–time pairs, $\bar{x}_r$ is the mean of factor r over these pairs, and $\hat{x}_{n,r}^{(K)}$ is the prediction from the corresponding latent prefix $\mathbf{s}_t^{1:K}$ using the ridge probe fitted on training data. For the latent spectrum, we eigendecompose the empirical covariance (with divisor equal to sample count) of all unmasked test embeddings, normalize its eigenvalues as $\nu_j = \lambda_j / \sum_\ell \lambda_\ell$, and report $d_{\text{eff}} = \exp(-\sum_j \nu_j \log \nu_j)$.

For ALeWM, let $\gamma_{b,j} = \Pr_{q_\psi}(K \geq j \mid \mathbf{w}_b)$ be the posterior survival probability of coordinate j for sequence b. Its soft masked-mixture variance over B sequences and T encoded time points is

$$v_j^{\text{mix}} = \frac{1}{BT}\sum_{b,t}\gamma_{b,j}(s_{b,t}^j)^2 - \left(\frac{1}{BT}\sum_{b,t}\gamma_{b,j}s_{b,t}^j\right)^2, \tag{39}$$

which we compare with the prior survival $\Pr_{\pi_0}(K \geq j)$ in Fig. 2(d).

Whitened Procrustes alignment. We examine whether the first four latent coordinates capture the four-dimensional physical state up to a linear change of coordinates. For each training trajectory–time pair n, let $\mathbf{y}_n = \mathbf{s}_t^{1:4}$ denote the latent prefix and $\mathbf{x}_n$ the corresponding true state. Stacking their transposes as rows gives matrices $\mathbf{Y}^{\text{tr}}, \mathbf{X}^{\text{tr}} \in \mathbb{R}^{n_{\text{tr}} \times 4}$, where n_{tr} is the number of training pairs.

We compute the training mean vectors $\boldsymbol{\mu}_y, \boldsymbol{\mu}_x$ and sample covariance matrices $\boldsymbol{\Sigma}_y, \boldsymbol{\Sigma}_x$. Let $\mathbf{Y}_c^{\text{tr}}$ and $\mathbf{X}_c^{\text{tr}}$ contain the centered rows $(\mathbf{y}_n - \boldsymbol{\mu}_y)^\top$ and $(\mathbf{x}_n - \boldsymbol{\mu}_x)^\top$. We apply regularized whitening:

$$\begin{aligned}\mathbf{Y}_w^{\text{tr}} &= \mathbf{Y}_c^{\text{tr}}(\boldsymbol{\Sigma}_y + \lambda_w \mathbf{I}_4)^{-1/2},\\ \mathbf{X}_w^{\text{tr}} &= \mathbf{X}_c^{\text{tr}}(\boldsymbol{\Sigma}_x + \lambda_w \mathbf{I}_4)^{-1/2},\end{aligned} \tag{40}$$

where $\lambda_w = 10^{-6}$ stabilizes the inverse. Whitening normalizes coordinate scales and correlations, allowing the alignment to focus on the correspondence between the two representations.

We then fit the orthogonal transformation

$$\mathbf{R}^\star = \arg\min_{\mathbf{R}^\top\mathbf{R}=\mathbf{I}_4} \left\|\mathbf{Y}_w^{\text{tr}}\mathbf{R} - \mathbf{X}_w^{\text{tr}}\right\|_F^2. \tag{41}$$

This objective minimizes the total squared discrepancy between paired, whitened representations. If $(\mathbf{Y}_w^{\text{tr}})^\top \mathbf{X}_w^{\text{tr}} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ is its singular value decomposition, a solution is $\mathbf{R}^\star = \mathbf{U}\mathbf{V}^\top$. The orthogonality constraint permits rotations and reflections.

Finally, we apply the training means, whitening matrices, and $\mathbf{R}^\star$ unchanged to the test data. The plot compares $\mathbf{Y}_w^{\text{te}}\mathbf{R}^\star$ with $\mathbf{X}_w^{\text{te}}$, so its coordinates are expressed in whitened space. Close agreement supports recovery of state information from the prefix up to a linear transformation; it does not require individual latent coordinates to correspond directly to individual state factors.

Hyperparameter search. We search over MixSIGReg/SIGReg weight in $\{0.001, 0.0025, 0.005, 0.0075, 0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 1, 5\}$. LeWM searches over fixed dimension $d \in \{2, 3, 4, 5, 6, 7, 8\}$, while the ALeWM sweep searches polynomial prior degree in $\{-3, -1.5, -0.5, 0, 0.5, 1.5, 3\}$. The main text reports degree -1.5 with MixSIGReg weight 0.01 for ALeWM, and SIGReg weights 0.005, 0.005, and 0.0075 for LeWM dimensions 4, 6, and 8, respectively.

E.2 Main Results

Data and preprocessing. For TwoRoom, PushT, Reacher, and OGBench-Cube, we follow the environments, offline-data collection, and goal-conditioned evaluation protocol of LeWM; Appendices D–F.1 of Maes et al. (2026) provide the remaining environment and collection details. We create a new episode-disjoint partitions. Reacher, OGBench-Cube, and TwoRoom each contain 10,000 episodes, split into 9,700/100/200 train/validation/test episodes. Validation is used for hyperparameter selection. The available PushT file contains 18,685 episodes, split into 18,385/100/200 episodes. Images are resized to 224×224. Training examples contain four frames: three context frames and one prediction target. Each successive pair is separated by five environment steps, and the intervening actions are flattened into a five-action block. We train and evaluate three model seeds, 3072, 3073, and 3074. The success rates in Table 1 first average over the 200 held-out test episodes for each seed and then report the mean and sample SEM over the three seeds.

World-model architecture. We retain the LeWM architecture and refer to Maes et al. (2026, Appendix D) for details not specified here. The $d_{\max} \in \{96, 192\}$ models use a randomly initialized ViT-Tiny/14 image encoder (12 blocks, 3 attention heads, and encoder width 192), while the $d_{\max} = 384$ models use ViT-Small/14 (12 blocks, 6 heads, and encoder width 384). The encoder CLS token is mapped to the latent state by a one-hidden-layer MLP with hidden width 2048 and batch normalization. The action-conditioned autoregressive predictor has six transformer blocks, 16 attention heads, an MLP width of 2048, and dropout 0.1. The encoded action blocks condition its transformer blocks through adaptive layer normalization; a second one-hidden-layer MLP projects the output to the prediction target.

Capacity network. ALeWM adds the categorical selector q_ψ. It takes the raw per-frame image encoder CLS tokens, before the latent projector, and applies four pre-norm transformer-encoder blocks with eight heads, feed-forward width 768. A final layer normalization and mean over the frame-token axis are followed by a linear C-class head, whose logits parameterize the supported capacities. Self attention without positional embeddings is permutation equivariant and the mean is permutation invariant. Consequently, the architecture and parameter count do not depend on the number of frames: the same network consumes all four training frames and the two-image initial/goal context at evaluation. The selector uses only supplied observations: local trajectory frames during training and the available initial and goal images at deployment. Permutation invariance makes their order irrelevant, and mean pooling allows variable frame counts. While adding or removing observations can change the output, in our short-horizon, controlled tasks, we expect capacity requirements to remain broadly similar across these contexts, although this does not establish calibration under the context shift. Longer or less controlled settings are an interesting direction for studying this dependence. Closed-loop planning could also update capacity using newly observed history, whereas our evaluation keeps it fixed per episode. During training, we intentionally conditioned the capacity network on the entire window of frames, and specifically the predicted frames, as it simulates testing environment where we access to goal frame and possibly other future frames. The CLS tokens are detached on the selector branch, so selector gradients do not flow back into the image encoder. The capacity network contains $1,780,805$, $1,781,384$, and $4,740,491$ trainable parameters for $d_{\max} = 96, 192, 384$, respectively, excluding the shared image encoder.

Optimization and regularization. All models are trained end-to-end for 10 epochs with batch size 128 using AdamW, learning rate 5×10^{-5}, weight decay 10^{-3}, bfloat16 mixed precision, and gradient-norm clipping at 1.0. Training is deterministic for each seed. SIGReg and MixSIGReg use 1,024 random projections and the Gaussian frequency window $w(\tau) = \exp(-\tau^2/2)$. We truncate the integral to $[-3, 3]$ and exploit the even squared characteristic-function discrepancy to evaluate twice the trapezoidal rule on $[0, 3]$. The 17 knots are $\tau_j = 3j/16$, $j = 0, \ldots, 16$, with spacing

$h = 3/16$ and weights $a_0 = a_{16} = h$ and $a_j = 2h$ for $1 \leq j \leq 15$, each multiplied by $w(\tau_j)$. The weighted discrepancy is multiplied by batch size B and averaged over projections and frame positions in both regularizers. For ALeWM, the full next embedding is the prediction target, the sampled prefix is obtained with the straight-through Gumbel–Softmax estimator at a constant relaxation temperature of 0.5. The selector head is initialized to the polynomial prior. We set the selector learning rate to $s_q(5 \times 10^{-5})$, where the multiplier s_q is a hyper-parameter as reported below.

The LeWM candidate grids contain the fixed latent widths $\{8, 16, 32, 64, 96\}$ for the 96-dimensional block, $\{8, 16, 32, 64, 96, 128, 160, 192\}$ for the 192-dimensional block, and $\{8, 16, 32, 64, 96, 128, 160, 192, 256, 320, 384\}$ for the 384-dimensional block. The corresponding ALeWM capacity supports are fixed and set to $\{8, 16, 32, 64, 96\}$, $\{8, 16, 32, 64, 96, 128, 160, 192\}$, and $\{8, 16, 32, 64, 96, 128, 160, 192, 256, 320, 384\}$, respectively. Both methods sweep the MixSIGReg/SIGReg coefficient over task-specific subsets of $\{0.09, 0.15, 0.2, 0.3\}$. ALeWM sweeps the selector learning-rate multiplier in $\{1, 2, 3\}$ and polynomial-prior degree in $\{-1, -0.5, 0, 1, 2, 3\}$. Table 3 records the exact ALeWM configurations used in Table 1.

Table 3: Selected ALeWM hyperparameters for the main control results. λ is the MixSIGReg coefficient, s_q multiplies the base learning rate for q_ψ, and α is the polynomial-prior degree.

$d_{\max}$	Dataset	λ	s_q	α
96	TwoRoom	0.15	3	3
96	PushT	0.09	2	-1
96	Reacher	0.30	3	2
96	OGBench-Cube	0.30	3	1
192	TwoRoom	0.30	3	3
192	PushT	0.15	2	-0.5
192	Reacher	0.30	3	-0.5
192	OGBench-Cube	0.20	3	1
384	TwoRoom	0.20	2	3
384	PushT	0.09	1	-0.5
384	Reacher	0.20	2	1
384	OGBench-Cube	0.15	2	0

Planning and evaluation. The four control environments use CEM with 300 candidate action sequences, initial variance 1, 30 refinement iterations, and 30 elites. The planning horizon is five action blocks of five actions each. These tasks execute five blocks before replanning. Each held-out episode is initialized from an offline trajectory, uses the observation 25 environment steps later as its goal, and has a 50-step interaction budget. Standard-task rollouts are evaluated in batches of 50. For ALeWM, q_ψ processes the initial and goal images once per episode, the modal supported capacity is cached across CEM iterations and replans, and both recursive dynamics rollout and goal distance use only that active prefix. LeWM uses its entire fixed-width representation. These choices match the "active" target-dimension setting reported in the main table.

Table 4: Per-seed planning-capacity counts for the selected ALeWM configurations in Table 1. An entry k: n means that $K_{\text{plan}} = k$ for n episodes; capacities that were never selected are omitted. Test contains 200 episodes per seed.

Backbone ($d_{\max}$)	Dataset	Seed	K_{plan}: #rollouts
ViT-Tiny (96)	TwoRoom	3072	16: 200
		3073	8: 200
		3074	16: 200
	PushT	3072	32: 200
		3073	32: 200
		3074	32: 200
	Reacher	3072	32: 200
		3073	16: 200
		3074	16: 200
	OGBench-Cube	3072	16: 200
		3073	16: 200
		3074	16: 200
ViT-Tiny (192)	TwoRoom	3072	8: 200
		3073	8: 200
		3074	8: 200
	PushT	3072	64: 200
		3073	32: 197, 64: 3
		3074	64: 200
	Reacher	3072	32: 147, 64: 53
		3073	32: 142, 64: 58
		3074	64: 200
	OGBench-Cube	3072	16: 200
		3073	16: 200
		3074	16: 200
ViT-Small (384)	TwoRoom	3072	8: 196, 16: 4
		3073	8: 195, 16: 5
		3074	8: 197, 16: 3
	PushT	3072	64: 200
		3073	32: 162, 64: 38
		3074	64: 200
	Reacher	3072	32: 200
		3073	32: 200
		3074	32: 200
	OGBench-Cube	3072	32: 200
		3073	32: 200
		3074	32: 200