CONTROL-GEOMETRY STRAIGHTENING FOR SAMPLING-BASED LATENT PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint-embedding predictive architectures enable planning with latent world models, but accurate transition prediction alone does not ensure that the planning objective is easy to optimize. We introduce Control-Geometry Straightening (CGS), a single auxiliary loss that learns planner-friendly representations by directly straightening control geometry for sampling-efficient planning. CGS matches pairwise cosine similarities among actions to those among corresponding latent differences only using local transitions from pixel–action pairs. The loss can be applied across world-model architectures using end-to-end learned or pretrained representations. Under linear-dynamics, our theoretical analysis connects this objective to temporal straightening and more balanced terminal-cost curvature across the full planning horizon, yielding finite-budget guarantees for MPPI, local contraction results for CEM, and convergence bounds for gradient descent. Across four control environments and multiple planners, CGS improves planning with fewer sampled candidates and refinement steps, achieving success-rate gains up to 20 and 12.6 percentage points over LeWorldModel (LeWM) and its temporal-straightening variant (LeWM+TS), respectively, with sampling-based planners using 128 candidates per update. Probes, comparisons with DINO-WM architecture, and planner-side ablations clarify how latent motion organization, state dependence, and dynamical context shape planning behavior. Straightening control geometry thus makes good action sequences easier to find under limited planning budgets.

1 INTRODUCTION

Latent world models enable agents to plan by predicting the consequences of actions in a learned representation space (Hafner et al., 2019; LeCun, 2022). Joint-embedding predictive architectures (JEPAs) predict future embeddings (Assran et al., 2023; Bardes et al., 2024). DINO-WM demonstrates the effectiveness of prediction on pretrained visual features (Zhou et al., 2025), while LeWorldModel (LeWM) enables stable end-to-end learning of the encoder and predictor directly from pixels (Maes et al., 2026). Across these approaches, planning relies on anti-collapsing predictive representations, while the geometry exposed to action optimization remains underspecified.

Sampling-based planners address nonconvex action-sequence optimization by evaluating predicted rollouts in parallel. CEM and MPPI optimize action sequences through elite selection or cost-weighted updates iteratively (de Boer et al., 2005; Williams et al., 2017; Yi et al., 2024). A central bottleneck is planning time, especially under tight sampling budgets, since each refinement requires many multi-step model rollouts. We therefore ask: *what representation geometry makes good action sequences easier to find with limited sampled candidates and refinement steps?*

Moving from predictive to *planner-friendly representations* requires more than accurate transition prediction: the representation geometry must also make the planning objective easier to optimize. Temporal Straightening (TS) regularizes latent-trajectory curvature, making latent distances reflect feasible progress and improving conditioning for gradient-based (GD) planning (Wang et al., 2026b). LeWM already exhibits implicit temporal straightening without a curvature loss (Maes et al., 2026); adding explicit TS shows environment-dependent effects in straightness and planning (Figure 12; Table 1). Yet TS aligns consecutive latent differences without explicitly structuring how actions shape latent transitions. This motivates directly shaping *control geometry*: how action effects are represented and accumulate through learned dynamics to determine the terminal-cost landscape.

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.

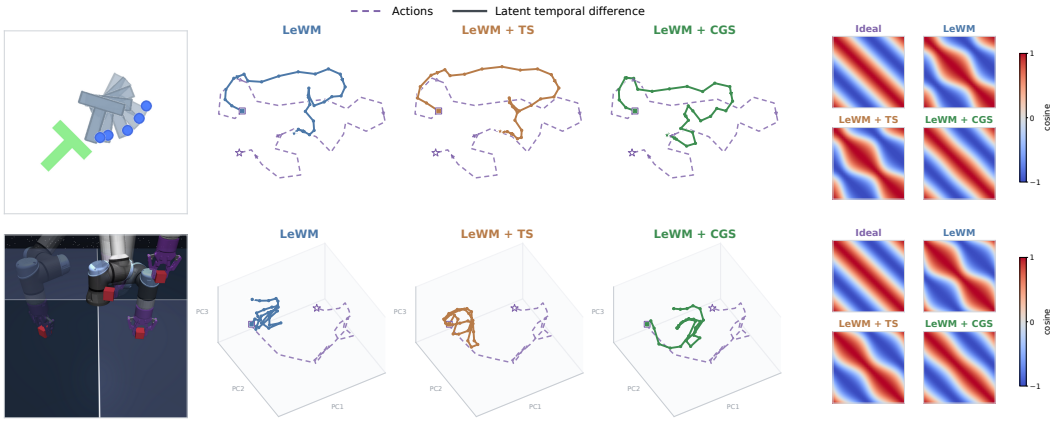


Figure 1: **CGS preserves action-related turns and control-response geometry.** PushT (top) and Cube (bottom), with LeWM (blue), LeWM+TS (brown), and LeWM+CGS (green). **Middle: transition shaping.** CGS makes latent transitions follow action-related turns more closely than LeWM and TS. **Right: control-response geometry.** The heatmaps show pairwise cosines of predicted endpoint changes induced by action-sequence perturbations; *Ideal* gives the angular relations among the perturbations themselves. CGS more closely preserves these relations in the tested control plane.

We introduce **Control-Geometry Straightening (CGS)**, which learns this geometry from local transitions by matching pairwise cosine similarities among actions to those among their corresponding latent differences. This single auxiliary loss applies directly to latent transitions and transfers across both end-to-end and pretrained world-model representations. Our linear-dynamics theoretical analysis shows how CGS *straightens* this geometry, promoting more balanced terminal-cost curvature across the full planning horizon for more effective action search. Experiments show substantial success-rate gains over LeWM and LeWM+TS across multiple planners, together with improved performance using fewer sampled candidates and refinement steps.

Our contributions are threefold:

- **Planner-friendly representation learning.** We introduce CGS, a single auxiliary loss that matches action and latent-difference cosine similarities to learn planner-friendly representations only using local transitions from pixel–action pairs, directly straightening control geometry for more effective multi-step planning under limited planning budgets.
- **Geometry-to-planning theory.** Under linear-dynamics, we develop a theoretical analysis linking world-model representation geometry to the optimization behavior of planners through finite-horizon planning curvature. This yields finite-budget guarantees for MPPI, local contraction results for CEM, and convergence bounds for gradient descent.
- **Empirical validation and characterization.** Across multiple planners, CGS improves planning performance and sample efficiency, reaching success-rate gains up to 20 and 12.6 percentage points over LeWM and LeWM+TS, respectively. Probes, cross-architecture comparisons with DINO-WM, and targeted planner-side ablations further clarify how latent motion organization, state dependence, and dynamical context shape planning behavior.

2 Related Work

Latent world models and JEPA. Latent world models learn compact dynamics for planning (Ha & Schmidhuber, 2018; Hafner et al., 2019; Hansen et al., 2024). JEPAs predict embeddings without reconstructing pixels (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024). DINO-WM and V-JEPA 2-AC build on pretrained visual features, while PLDM and LeWM learn representations and dynamics jointly (Zhou et al., 2025; Assran et al., 2025; Sobal et al., 2025; Maes et al., 2026). Much of this development addresses representation collapse through stable targets, variance regularization, or distribution matching (Assran et al., 2023; Bardes et al., 2022; Balestriero & LeCun, 2025). These mechanisms stabilize predictive learning, while leaving underdetermined how representation geometry should support planning by making good action sequences easier to find (Li et al., 2026).

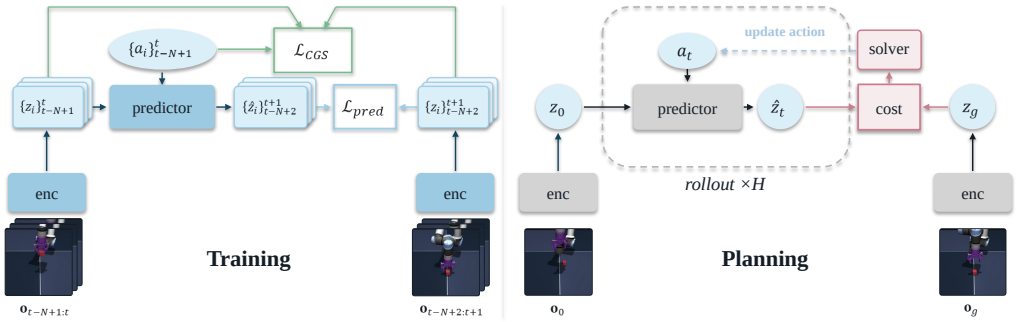


Figure 2: **CGS training and planning.** During training, we minimize the prediction loss between predicted and encoded target embeddings, together with a CGS loss that matches pairwise cosine similarities of standardized actions and consecutive latent differences; SIGReg is omitted from the diagram. During planning, we roll out candidate action sequences for H steps using the frozen predictor and refine them with a sampling-based planner to minimize the distance between the predicted terminal embedding and the goal embedding.

Planner-friendly representations. Earlier work learns locally linear latent dynamics for trajectory optimization or representations through differentiable planners (Watter et al., 2015; Zhang et al., 2019; Srinivas et al., 2018). Recent JEPA methods add goal-conditioned values, budget-conditioned reachability, or future-goal inverse control (Destrade et al., 2025; Li et al., 2026; Sun et al., 2026). Inverse-dynamics objectives retain action information in latent transitions (Ivashkov et al., 2026; Zhang et al., 2026b). Temporal Straightening aligns consecutive latent displacements to reduce trajectory curvature (Wang et al., 2026b), without structuring how actions shape latent transitions. Relational and similarity-preserving objectives in vision knowledge distillation transfer structure between teacher and student representations (Park et al., 2019; Tung & Mori, 2019).

Sampling-based MPC. CEM fits proposals to low-cost elites; MPPI weights candidates by rollout cost (de Boer et al., 2005; Williams et al., 2017). iCEM reuses samples, CoVO-MPC adapts covariance, and Biased-MPPI and PRISM use controller or learned-prior proposals (Pinneri et al., 2021; Yi et al., 2024; Trevisan & Alonso-Mora, 2024; Wang et al., 2026c). Theory studies MPPI updates and finite-sample stability (Fazlyab et al., 2026; Yoon & Kim, 2026). Sampling efficiency also depends on the horizon, effective action dimension, cost landscape, and proposal design (Yoon et al., 2022). See Appendix A for more discussions about extended related work.

3 Control-Geometry Straightening

For our main analysis, we formulate CGS using the end-to-end LeWM architecture (Maes et al., 2026) and offline pixel–action trajectories (o_t, a_t). CGS learns control geometry by matching pairwise cosine similarities among actions to those among their corresponding latent differences. As developed below, this local geometric constraint straightens the finite-horizon planning geometry. Figure 2 shows the training and planning pipeline; the SIGReg term is omitted from the diagram.

3.1 Latent world model and planning setup

Following LeWM (Maes et al., 2026), our latent world model has two components:

$$\begin{aligned} \text{Encoder:} \quad & z_t = f_\theta(o_t), \\ \text{Predictor:} \quad & \widehat{z}_{t-N+2:t+1} = g_\phi(z_{t-N+1:t}, a_{t-N+1:t})\,. \end{aligned} \tag{1}$$

The encoder f_θ is a tiny ViT (Dosovitskiy et al., 2021). It constructs embedding $z_t \in \mathbb{R}^d$ with observation o_t from the output [CLS] token through an MLP projection step. The transformer predictor g_ϕ, followed by an MLP projector, takes a history of N frame representations and corresponding actions $a_t \in \mathbb{R}^{d_a}$ as input. Equation (1) expresses prediction in sequence-to-sequence form, with N input frames and N next-step predictions. Specifically, temporal causal masking is used in the predictor to restrict each next-step prediction to inputs at or before the current time, as shown in Eq. (2):

$$\widehat{z}_{t-N+2} = g_\phi(z_{t-N+1}, a_{t-N+1})\,,\ \ldots\ ,\ \widehat{z}_{t+1} = g_\phi(z_{t-N+1:t}, a_{t-N+1:t})\,. \tag{2}$$

At planning time, we optimize an action sequence $\boldsymbol{a} = (a_0, \ldots, a_{H-1}) \in \mathbb{R}^{H \times d_a}$ with frozen world model as the dynamical model. We encode the initial and goal observation pixels to z_0 and z_g, and search for an action sequence whose predicted latent rollout minimizes the goal-matching cost

$$C(\boldsymbol{a}) = \tfrac{1}{2}\|\widehat{F}_H(\boldsymbol{a}) - z_g\|_2^2, \tag{3}$$

where the terminal latent prediction over a horizon H is denoted as $\widehat{F}_H(\boldsymbol{a})$.

3.2 Learning Control Geometry by Pairwise Similarity Matching

We seek to straighten control geometry in the latent space by matching the pairwise cosine similarities of latent differences to those of actions. For each transition t, let $\Delta z_t = f_\theta(o_{t+1}) - f_\theta(o_t)$. Consider a history window with $N \geq 2$ transitions, and define $u_t = \frac{a_t}{\|a_t\|_2}$,$v_t = \frac{\Delta z_t}{\|\Delta z_t\|_2}$. Stack them as the rows of $U \in \mathbb{R}^{N \times d_a}$ and $V \in \mathbb{R}^{N \times d}$. Their *cosine Gram matrices* are

$$K_A = UU^\top, \qquad K_Z = VV^\top, \tag{4}$$

Thus, K_A and K_Z contain pairwise action and latent-difference cosines, respectively. CGS minimizes

$$\mathcal{L}_{\mathrm{CGS}} = \frac{\|K_Z - K_A\|_F^2}{N(N-1)} = \frac{1}{N(N-1)} \sum_{t \neq s} \left[\cos(\Delta z_t, \Delta z_s) - \cos(a_t, a_s)\right]^2, \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Both matrices have unit diagonal, so the loss averages squared discrepancies over $N(N-1)$ off-diagonal entries. CGS uses action directions as a geometric reference for latent transitions: aligned, orthogonal, and opposing actions encourage corresponding angular relationships between latent differences. The pairwise matching form resembles similarity-preserving knowledge distillation (Tung & Mori, 2019). Here, actions instead provide the reference geometry for latent transitions, and the matched relations encode control geometry for planning. As developed in Section 4, this local geometric constraint promotes more balanced finite-horizon planning curvature and more effective action search under limited sampling budgets.

Relation to temporal straightening. When consecutive actions point in the same direction, $u_t^\top u_{t+1} = 1$, TS and CGS both encourage consecutive latent differences to align. The adjacent CGS term is then $(1 - v_t^\top v_{t+1})^2$, the squared angular TS loss in Wang et al. (2026b).

3.3 Training objective

We jointly train the encoder and predictor with CGS and the LeWM objectives. Using the N-frame history window, the prediction loss is

$$\mathcal{L}_{\mathrm{pred}} = (Nd)^{-1} \, \|\widehat{z}_{t-N+2:t+1} - z_{t-N+2:t+1}\|_F^2 \,. \tag{6}$$

The factor $(Nd)^{-1}$ averages squared errors over the N next-step predictions and d latent coordinates. The base objective of LeWM (Maes et al., 2026) is

$$\mathcal{L}_{\mathrm{LeWM}} = \mathcal{L}_{\mathrm{pred}} + \lambda_{\mathrm{SIGReg}} \mathcal{L}_{\mathrm{SIGReg}}, \tag{7}$$

where $\mathcal{L}_{\mathrm{SIGReg}}$ is LeWM's SIGReg anti-collapse term, which promotes feature diversity by encouraging an isotropic Gaussian distribution of latent embeddings (Maes et al., 2026). The full training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{LeWM}} + \lambda_{\mathrm{CGS}} \mathcal{L}_{\mathrm{CGS}}, \tag{8}$$

where $\lambda_{\mathrm{CGS}} \geq 0$ controls the strength of the control geometry straightening. To help preserve the predictive geometry while gradually introducing control-geometry regularization, we linearly ramp the CGS weight from zero to the final value λ_{CGS} over the first 5% of training updates. We use the default history length $N = 3$ throughout training, following LeWM (Maes et al., 2026). Additional ablations and discussion of the ramp schedule and history length appear in Appendix C.6.

4 Theory: From CGS to Sampling-Efficient Planning

In this section, we analyze how CGS supports sampling-efficient planning under linear latent dynamics. We connect CGS to temporal straightening and balanced action effects, then derive consequences for finite-budget planning over a finite horizon. Full proofs, theoretical results of CEM and gradient-based planners, and finite-budget detailed discussions are provided in Appendix D.

Setup. For analysis, we study linear latent dynamics,

$$z_{t+1} = Az_t + Ba_t, \tag{9}$$

with $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times d_a}$. We first consider $d_a = d$ with B invertible; see Remark 4.5 for $d_a < d$. For goal-reaching planning, we optimize the action sequence $\boldsymbol{a} = (a_0, \dots, a_{H-1}) \in \mathbb{R}^{H\times d_a}$, $n_H = H \times d_a$, to minimize the terminal latent goal-matching objective

$$C(\mathbf{a}) = \tfrac{1}{2}\|z_H - z_g\|_2^2, \quad z_H = \widehat{F}_H(\boldsymbol{a}),$$

with latent goal z_g and terminal prediction $\widehat{F}_H(\boldsymbol{a})$ as in Eq. (3), where planning horizon $H \geq 2$.

4.1 Latent Control Geometry Learned by CGS

Letting $D := A - I_d$, with I_d denoting the $d \times d$-dimensional identity matrix, the latent difference is $\Delta z_t = Dz_t + Ba_t$. Under the fixed-radius assumption in Eq. (35), the population CGS loss is a positive multiple of $\mathcal{L}_G(c) := \mathbb{E}[(\Delta z^\top \Delta z' - c\, a^\top a')^2]$, where $(\Delta z, a)$ and $(\Delta z', a')$ are independent transition samples and $c > 0$ is a scale factor set by fixed radii of actions and latent differences.

Assumption 4.1 (Bounded features and joint coverage)**.** Following Yin et al. (2022), we assume there exists a feature map ψ satisfying (i) boundedness, $\|\psi(z_t, a_t)\|_2^2 \leq \bar{\rho} < \infty$ a.s., and (ii) coverage, $\lambda_{\min}(\Sigma_x) \geq \rho > 0$, where $\Sigma_x := \mathbb{E}[\psi(z_t, a_t)\psi(z_t, a_t)^\top]$. Here we use $\psi(z_t, a_t) = x_t = [z_t^\top, a_t^\top]^\top$.

Proposition 4.2 (Temporal straightening and control isotropy under CGS)**.** *Under Assumption 4.1, $\rho I_{2d} \preceq \Sigma_x \preceq \bar{\rho} I_{2d}$. With $\mathcal{R}(A, B) := \|D^\top D\|_F^2 + 2\|D^\top B\|_F^2 + \|B^\top B - cI_{d_a}\|_F^2$, we have*

$$\rho^2 \mathcal{R}(A, B) \leq \mathcal{L}_G(c) \leq \bar{\rho}^2 \mathcal{R}(A, B). \tag{10}$$

In this model class, $\mathcal{L}_G(c) = 0$ iff $A = I_d$ and $B^\top B = cI_{d_a}$; Proof is in Appendix D.1.1.

Under Assumption 4.1, CGS encourages temporal straightening ($A \approx I_d$), reduces drift–control coupling ($D^\top B \approx 0$) and promotes balanced action effects ($B^\top B \approx cI_{d_a}$). During joint training, prediction and other losses may favor different representation geometry, so the CGS loss may remain nonzero. CGS can still improve control geometry under these constraints; see Appendix D.7.

4.2 From Finite-Horizon Planning Geometry to Finite-Budget MPPI

Unrolling (9) gives $z_H = A^H z_0 + \Gamma_H \mathbf{a}$, where $\Gamma_H = [A^{H-1}B, \dots, B]$, and the planning Hessian is $G_H := \nabla_{\mathbf{a}}^2 C = \Gamma_H^\top \Gamma_H$. Since B is invertible, G_H has rank d, and its positive eigenvalues coincide with those of the controllability Gramian $W_H := \Gamma_H \Gamma_H^\top$. We measure conditioning by the ratio of the largest to smallest positive eigenvalues. Let $P_H := H^{-1}(\mathbf{1}_H \mathbf{1}_H^\top) \otimes I_{d_a}$, where $\mathbf{1}_H \in \mathbb{R}^H$ is the all-ones vector. This rank-d_a orthogonal projector maps onto the subspace of action-sequence perturbations constant across time. When $A = I_d$ and $B^\top B = cI_{d_a}$, $z_H = z_0 + B\sum_t a_t$ and $G_H = cHP_H$, yielding equal curvature along all directions in this active subspace, termed *planning isotropy*. With isotropic Gaussian proposals, population MPPI updates then contract all active directions equally, balancing refinement across action directions. The rightmost heatmaps of Figure 1 illustrate CGS's strong performance in *planning isotropy* (Appendix C.2.2).

Corollary 4.3 (From CGS loss to drift and control bounds)**.** *Under Assumption 4.1, Proposition 4.2 implies the following. If $\mathcal{L}_G(c) \leq \ell$, define $\varepsilon = (\sqrt{\ell}/\rho)\max\{1, (2c)^{-1/2}, c^{-1}\}$. Then $\|D\|_{\mathrm{op}}^2 \leq \varepsilon$, $\|D^\top B\|_{\mathrm{op}} \leq \sqrt{c}\,\varepsilon$, and $\|B^\top B - cI_{d_a}\|_{\mathrm{op}} \leq c\varepsilon$. Here $\|\cdot\|_{\mathrm{op}}$ is the spectral norm. These quantify worst-direction geometric deviations. We call these the ε-CGS bounds. Proof is in Appendix D.1.2.*

Theorem 4.4 (CGS controls the planning Hessian)**.** *Suppose the three bounds in Corollary 4.3 hold for $0 \leq \varepsilon < 1$, there exists ξ_H (defined in Eq. (50)) such that*

$$\left\| \frac{G_H}{cH} - P_H \right\|_{\mathrm{op}} \leq \xi_H(\varepsilon), \quad \xi_H(\varepsilon) = (2H^2 - 3H + 2)\varepsilon + O_H(\varepsilon^{3/2}), \tag{11}$$

where O_H hides constants depending only on the fixed horizon H. If $\xi_H := \xi_H(\varepsilon) < 1$, then, writing $r = d_a = d$ and $\sigma_1(G_H) \geq \cdots \geq \sigma_{n_H}(G_H)$, the first r singular values (equivalently, eigenvalues) lie in $[cH(1-\xi_H), cH(1+\xi_H)]$, all remaining singular values are zero, and $\kappa_{\mathrm{eff}}(G_H) := \sigma_1(G_H)/\sigma_r(G_H) \leq (1+\xi_H)/(1-\xi_H)$. Proof is in Appendix D.2.1.

For fixed $H \geq 2$ and the same control-isotropy tolerance, CGS directly controls $\|D^\top B\|_{\mathrm{op}} = O(\varepsilon)$ via its explicit mixed drift–control defect bound, while temporal straightening gives the indirect $O(\sqrt{\varepsilon})$ bound. This sharpens the Hessian perturbation from $O_H(\sqrt{\varepsilon})$ to $O_H(\varepsilon)$; see Appendix D.2.

Remark 4.5 (Low-dimensional actions)**.** When $d_a < d$, the bound on $\|G_H/(cH) - P_H\|_{\mathrm{op}}$ in Theorem 4.4 remains valid on the full action-sequence space $\mathbb{R}^{n_H}$. For $\xi_H < 1$, its condition-number bound holds on the subspace spanned by the leading d_a eigenvectors of G_H, and the subsequent planner analyses are carried out on these directions. See Appendix D.2.2.

Theorem 4.6 (MPPI population mean update)**.** *Assume $\xi_H < 1$. Let $\tau > 0$ be the MPPI temperature and $\sigma_{\mathrm{p}} > 0$ the sampling standard deviation. Let $\mathbf{a}^\star$ minimize C, condition on the proposal $\mathcal{N}(\mu, \sigma_{\mathrm{p}}^2 I_{n_H})$, and set $\nu = \sigma_{\mathrm{p}}^{-1}(\mu - \mathbf{a}^\star)$, $M_H = G_H/(cH)$, and $s = \sigma_{\mathrm{p}}^2 cH/\tau$. Let $\mathcal{S}_H := \mathrm{range}(G_H)$, which has dimension $r = d$, and contraction $q_{\mathrm{M}} := \|J_{\mathrm{M}}|_{\mathcal{S}_H}\|_{\mathrm{op}}$, defined as the spectral norm of the Jacobian restricted to the active subspace $\mathcal{S}_H$. The population map, Jacobian, and active contraction are*

$$T_{\mathrm{M}}(\nu) = J_{\mathrm{M}}\nu, \qquad J_{\mathrm{M}} = (I_{n_H} + sM_H)^{-1},$$
$$q_{\mathrm{M}} = \frac{1}{1 + (\sigma_{\mathrm{p}}^2/\tau)\sigma_r(G_H)} \leq \frac{1}{1 + s(1 - \xi_H)}. \tag{12}$$

Moreover, $\kappa(J_{\mathrm{M}}|_{\mathcal{S}_H}) \leq [1+s(1+\xi_H)]/[1+s(1-\xi_H)]$; at $M_H = P_H$, all active directions contract uniformly by $(1+s)^{-1}$. Proof is in Appendix D.4.1.

Finite-budget planning cost. Let μ_i be the mean after i MPPI updates and $\Delta_i := C(\mu_i) - C(\mathbf{a}^\star)$ its planning cost gap. Fix σ_{p} and τ, use K independent samples per update, and impose the minimum-K condition in Eq. (91). With probability at least $1 - \zeta$, the cost gap after I updates satisfies $\sqrt{\Delta_I} \leq q_{\mathrm{M}}^I \sqrt{\Delta_0} + (1 - q_{\mathrm{M}}^I) b_K/(1 - q_{\mathrm{M}})$ whenever $\|\mu_i - \mathbf{a}^\star\|_2 \leq \sigma_{\mathrm{p}} R_\nu$ for all $0 \leq i < I$, where R_ν is defined in Eq. (76). Here $b_K := \sigma_{\mathrm{p}} \mathcal{A}_{\mathrm{M}}(R_\nu)\sqrt{cH(1+\xi_H) r \log(4rI/\zeta)/(2K)}$ bounds the per-update sampling contribution to the square-root cost gap, and $\mathcal{A}_{\mathrm{M}}(R_\nu)$ is a dimension-dependent importance-weight factor. With $q_{\mathrm{M}} < 1$, more updates suppress the initial error, while more samples reduce the sampling term. Appendix D.4.3 gives the factor's definition, the proof, and sufficient budgets for a prescribed target error.

Theorem 4.6 shows how CGS balances population MPPI updates by promoting comparable curvature across active action directions. With the cost scale calibrated, this spectral balance can support faster population contraction and a higher probability of progress under a limited sampling budget. In trace-normalized quadratic probes of Cube's XYZ translation subspace, CGS yields a lower population contraction factor q_∞ and a higher one-step progress probability $\Pr(q_K < 1)$ at the same sample budget per update across models (Figure 3).

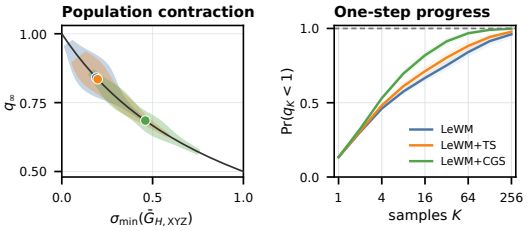


Figure 3: **MPPI contraction and finite-sample improvement.** (a) CGS lowers population contraction q_∞; $\sigma_{\min}$ measures the weakest curvature. (b) CGS increases $\Pr(q_K < 1)$, where q_K is the K-sample contraction factor. Definitions and protocol: Appendix C.4.

Remark 4.7 (CEM and gradient-based planner)**.** Appendix D.5 shows that the same projector geometry of the planning Hessian makes the local CEM mean update approximately scalar on its active subspace, reducing directional imbalance in elite selection. For gradient descent planner, Appendix D.6 also gives planning-cost convergence bounds on the leading controllable directions across iterative updates, linking CGS Hessian geometry to first-order action optimization.

5 Main Results: Planning Performance and Sample Efficiency

5.1 Experimental Setup

Environments and models. We evaluate PushT (Chi et al., 2023), Cube (Park et al., 2025), TwoRooms (Sobal et al., 2025), and Reacher (Tunyasuvunakool et al., 2020) using offline pixel–action data. All LeWM-based models follow the training setup of Maes et al. (2026). We keep the base LeWM hyperparameters fixed across methods, including a shared SIGReg weight of 0.09, and vary only the coefficient introduced by each additional regularizer. For LeWM+TS (Wang et al.,

Table 1: **Goal-reaching success rate (SR) at $K = 128$.** Values are reported as means ± sample standard deviation (SD), in %. Bold marks the best per planner and environment, including ties. Parentheses report gains over LeWM in percentage points; the top two per environment are underlined. Prior-guided (PG) variants (†) use action-prior initialization. Sampling planners use $I = 30$ optimization steps; gradient descent (GD) uses 100 Adam optimization steps, independent of K.

World model	Planner	PushT	Cube	TwoRooms	Reacher
LeWM	MPPI	64.0 ±5.3	59.3 ±4.6	92.0 ±3.5	65.3 ±4.2
	PG-MPPI†	72.0 ±6.0	89.3 ±4.2	98.0 ±2.0	62.7 ±11.0
	CEM	89.3 ±6.1	63.3 ±2.3	76.7 ±10.1	**83.3** ±8.3
	PG-CEM†	92.0 ±5.3	90.0 ±2.0	93.3 ±3.1	**86.7** ±6.1
	GD	85.3 ±5.0	59.3 ±8.1	41.3 ±7.0	76.7 ±4.2
LeWM + TS	MPPI	64.7 ±8.3 (+0.7)	75.3 ±5.0 (+16.0)	**97.3** ±3.1 (+5.3)	64.0 ±2.0 (−1.3)
	PG-MPPI†	69.3 ±9.0 (−2.7)	92.7 ±3.1 (+3.3)	98.0 ±2.0 (+0.0)	66.7 ±2.3 (+4.0)
	CEM	86.7 ±5.0 (−2.7)	73.3 ±10.3 (+10.0)	**97.3** ±1.2 (+20.7)	81.3 ±8.1 (−2.0)
	PG-CEM†	90.0 ±5.3 (−2.0)	90.0 ±2.0 (+0.0)	**100.0** ±0.0 (+6.7)	78.0 ±2.0 (−8.7)
	GD	82.7 ±5.8 (−2.7)	65.3 ±4.6 (+6.0)	65.3 ±9.9 (+24.0)	81.3 ±5.0 (+4.7)
LeWM + CGS	MPPI	**77.3** ±6.4 (+13.3)	**79.3** ±4.6 (+20.0)	**97.3** ±1.2 (+5.3)	**66.7** ±2.3 (+1.3)
	PG-MPPI†	**80.0** ±2.0 (+8.0)	**96.7** ±3.1 (+7.3)	**100.0** ±0.0 (+2.0)	**69.3** ±3.1 (+6.7)
	CEM	**92.0** ±3.5 (+2.7)	**76.7** ±9.5 (+13.3)	95.3 ±1.2 (+18.7)	80.7 ±3.1 (−2.7)
	PG-CEM†	**96.0** ±0.0 (+4.0)	**96.0** ±4.0 (+6.0)	**100.0** ±0.0 (+6.7)	81.3 ±3.1 (−5.3)
	GD	**89.3** ±6.4 (+4.0)	**71.3** ±4.6 (+12.0)	**70.7** ±3.1 (+29.3)	**82.0** ±5.3 (+5.3)

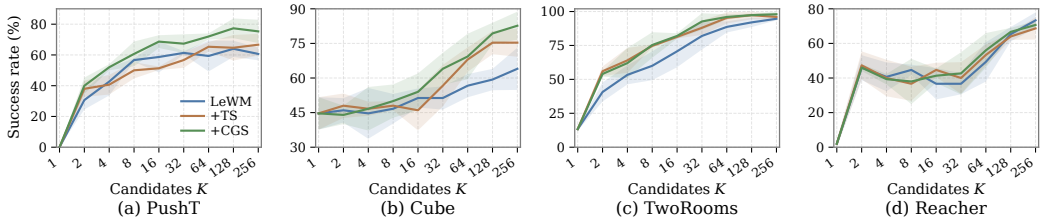


Figure 4: **MPPI success rate versus sampling budget.** Means and ± one sample SD, with $I = 30$ optimization steps and no action prior. CGS leads on PushT and at larger budgets on Cube; TwoRooms approaches saturation. Reacher shows smaller, budget-dependent differences. Environment-specific temperatures are fixed across budgets; See Appendix C.1.2 for results of other planners.

2026b), we sweep $\lambda_{\text{TS}} \in \{0.01, 0.05, 0.1, 0.5\}$, yielding selected weights of 0.01 on PushT, 0.1 on Cube and Reacher, and 0.5 on TwoRooms. CGS uses $\lambda_{\text{CGS}} = 0.5$ by default and 0.01 on PushT. Each checkpoint remains fixed across planners and planning budgets. See Appendix C.1.1 for details.

Evaluation and planners. We report goal-reaching success rate (SR, %) as mean ± sample standard deviation (SD). MPPI and CEM use $I = 30$ optimization steps (iterations), with $K = 1$–256 candidates and $K = 128$ in Table 1. Gradient descent (GD) uses 100 Adam optimization steps, independent of K. Prior-guided (PG) variants, PG-MPPI and PG-CEM, initialize the proposal mean with a representation-matched behavior-cloning prior. Evaluation details appear in Appendix C.1.1.

5.2 Toward Efficient Planning with Action-Structured Latent Dynamics

Under a shared evaluation protocol, CGS consistently improves over LeWM for all five planners on PushT, Cube, and TwoRooms at $K = 128$ (Table 1), across distinct planning mechanisms. On PushT and Cube, its MPPI gains are 13.3 and 20.0 percentage points over LeWM, respectively. Combining CGS with a learned action prior further improves planning, achieving near-perfect success in several settings. TS also provides a strong baseline, with substantial gains on Cube and TwoRooms and improvements in several other settings. Figure 12 shows that explicit TS further straightens latent trajectories across all four environments, beyond the implicit straightening reported by Maes et al. (2026, Appendix H). On PushT, where this implicit effect is already strong, TS yields only a subtle additional straightening and correspondingly modest planning changes.

The budget sweep in Figure 4 demonstrates *sampling-efficient planning* under a limited planning budget. On PushT, CGS improves MPPI even with few samples and retains its advantage over both baselines as K increases. On Cube, the gap widens with more samples, reaching around 80% success with CGS versus 60% with LeWM at $K = 128$–256, suggesting better action search at a fixed candidate budget. TwoRooms is easy for MPPI, plausibly because its limited data diversity and

Table 2: **Physical-motion probing on PushT and Reacher.** P_A and P_{AS} measure how well physical motion is identified from action alone or from action and state; these environment-level scores are shared across representations. D_{lin} and D_{MLP} decode true motion from Δz_t, while C_A and B_{AS} measure agreement with action-only and state-conditioned control effects. Scores are held-out R^2, except the normalized agreement score B_{AS} (Appendix C.3). Higher is better; bold marks the best representation per metric using unrounded scores.

		Physical identification		Motion decoding		Control effects	
Environment / effect	Representation	P_A	P_{AS}	D_{lin}	D_{MLP}	C_A	B_{AS}
PushT / agent	LeWM	0.999	0.999	0.809	0.888	0.807	0.808
	LeWM+TS			0.831	0.891	0.829	0.830
	LeWM+CGS			**0.903**	**0.959**	**0.901**	**0.902**
Reacher / fingertip	LeWM	-0.006	0.983	0.951	**0.994**	0.155	0.932
	LeWM+TS			**0.974**	0.994	**0.157**	**0.959**
	LeWM+CGS			0.899	0.993	0.041	0.884

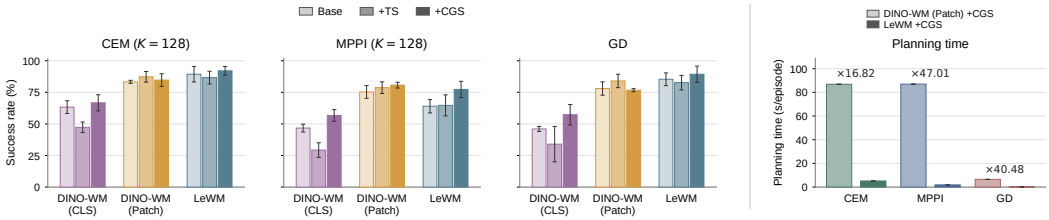


Figure 5: **Planning across representation architectures on PushT.** Left: planning success with DINO-WM (with proprioceptive information) global CLS features, spatial Patch features, and the end-to-end LeWM latent under CEM, MPPI, and GD, with Base (w/o TS or CGS), TS, and CGS representation shaping. Right: planner-solve time for DINO-WM (Patch)+CGS versus LeWM+CGS; annotated ratios are DINO-WM/LeWM. Full experimental details are provided in Appendix C.5.

low intrinsic dimensionality simplify action search, yet CGS and TS outperform LeWM at limited budgets, with CGS outperforming both at $K = 32$. Across these environments, CGS improves both attainable success and the rate at which planning budgets translate into successful trajectories.

Reacher exhibits a planner-dependent pattern. At $K = 128$, CGS improves GD from 76.7% to 82.0% and MPPI from 65.3% to 66.7%, while CEM remains close to LeWM but slightly lower (80.7% versus 83.3%). TS shows a similar pattern, with a clear GD gain and smaller changes under the sampling-based planners. Across the sample-budget sweep, CGS gives a modest MPPI improvement. Section 6 shows that Reacher fingertip motion depends strongly on the current state beyond the executed action. Appendix C.1.4 compares action-only CGS with state-informed transition references that implicitly capture state and action information. Appendix D.7 then analyzes how irreducible state-dependent drift can limit straightening and reshape the control map favored by CGS.

6 Characterizing CGS: Latent Representations and Planning

Probing the physical effects of control. With frozen encoders, we probe how latent differences Δz_t represent physical motion on PushT and Reacher. We first ask whether the evaluated displacement is determined by action alone or requires the current state, and then measure how explicitly that motion is represented in Δz_t. Let a_t denote the action, s_t the physical state, and Δs_t the evaluated displacement. Table 2 reports six held-out scores, with disjoint training, validation, and test episodes.

Physical identification (P_A, P_{AS}). We predict Δs_t from a_t and (a_t, s_t) using MLPs. On PushT, action alone nearly determines agent displacement ($P_A = P_{AS} = 0.999$). On Reacher, adding state raises R^2 from -0.006 to 0.983, showing that fingertip motion depends strongly on the current configuration and is only weakly specified by action alone for the observed one-step motion.

Motion decoding ($D_{\text{lin}}, D_{\text{MLP}}$). We decode the true displacement Δs_t from Δz_t with a ridge regressor or MLP. On PushT, CGS raises linear R^2 from 0.809 to 0.903 and nonlinear R^2 from 0.888 to 0.959. On Reacher, all three representations retain nearly all motion information nonlinearly, while linear decoding varies more across the three representations: TS reaches 0.974 and CGS 0.899.

Control effects (C_A, B_{AS}). We ask whether Δz_t reflects what the action does physically. C_A tests whether a linear readout from Δz_t recovers the displacement predicted from action alone; B_{AS}

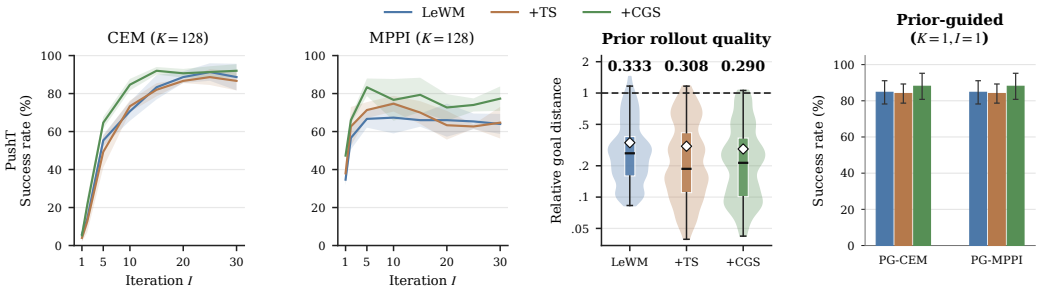


Figure 6: **CGS improves early refinement and BC-prior quality on PushT.** Left: CEM/MPPI success over optimization steps I (iterations) at $K = 128$ without an action prior. Middle: BC-prior/no-op latent goal-distance ratios for the first plan (lower is better). Right: prior-only success using the prior mean as the sole candidate without additional candidate sampling ($K = 1$, $I = 1$).

compares its decoded displacement with that predicted from action and current state. CGS is strongest on PushT ($C_A = 0.901$, $B_{AS} = 0.902$). TS leads both on Reacher, where the same action can produce different fingertip motion across configurations. See Appendix C.3 for details.

CGS across representation architectures. Figure 5 follows the global-versus-spatial representation settings studied by DINO-WM and TS (Zhou et al., 2025; Wang et al., 2026b) to examine how CGS interacts with different latent architectures. Models with DINO-WM architecture are trained following Wang et al. (2026b), with proprioceptive information. CGS improves planning with both DINO CLS and spatial Patch features, with notably larger gains for the compact CLS representation on PushT. This comparison also motivates our use of LeWM in the main experiments: frozen DINO CLS features are compact but weaker for dynamics-based planning, whereas spatial Patch features improve planning at substantially higher rollout cost. LeWM instead learns a compact global latent jointly with predictive dynamics, retaining strong planning performance while making repeated planner rollouts much cheaper. Importantly, applying TS to the same LeWM backbone remains stronger than the global DINO-WM+TS configuration and competitive with Patch+TS under CEM and GD, establishing LeWM+TS as a strong baseline. CGS then benefits from this efficient learned latent while enjoying significantly faster planning.

Refinement efficiency and BC priors. Each CEM/MPPI optimization step refines the sampling proposal to favor lower-cost action sequences. At a fixed candidate budget, CGS reaches high PushT success in fewer optimization steps (Figure 6), showing gains early in this refinement process. CGS also improves the matched BC prior: its rollouts achieve lower latent goal-distance ratios and higher prior-only success before search begins. Thus, CGS benefits both iterative refinement and prior-guided initialization. Results for other environments appear in Appendix C.8.

7 Conclusion

We introduced CGS, a simple auxiliary loss that learns planner-friendly representations by straightening control geometry. Our linear-dynamics analysis connects this geometry to more balanced terminal-cost curvature along controllable directions and more efficient finite-budget optimization. Experiments across multiple tasks and planners demonstrate gains in planning success and refinement efficiency using limited planning budget. Together, these results show that local geometric regularization can shape the full-horizon optimization landscape without modifying the planner at test time. More broadly, our findings highlight the importance of how predictive representations organize action effects for planning. Extending CGS to state-dependent control effects and richer dynamical context using only pixel–action data is a natural next step. Its focus on the geometry used by planners also motivates studying how to maintain planner-friendly representations under distribution shift, potentially in combination with test-time adaptation (Wang et al., 2026a).

AI USE STATEMENT

In this work, we used generative AI tools to assist with selected parts of the research process, including feedback on experimental design, portions of mathematical derivations and proofs, and code organization, cleanup, and debugging. The main theoretical development and experimental design were carried out by the authors. Additionally, generative AI tools were used to polish the writing for clarity and phrasing, identify and retrieve relevant literature, and assist with LaTeX and bibliography formatting. Generative AI tools also assisted with drafting and editing portions of the manuscript; they were not used to generate synthetic datasets. We reviewed all AI-assisted work: mathematical arguments and proof steps were checked by the authors, code changes were reviewed and tested, experimental suggestions were evaluated before use, and cited literature was verified against the original sources. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The model and planner definitions are given in Sections 3 and 5, and Appendix B specifies the referenced planning algorithms. Appendix C.1.1 records the training and evaluation protocols, including model settings, regularization coefficients, planning horizons, and evaluation budgets, while the remaining sections of Appendix C document the complete budget sweeps, ablations, probe definitions, and analysis protocols. The anonymous supplementary archive provides training and evaluation code for the main LeWM-based experiments on all four environments, including LeWM, LeWM+TS, and LeWM+CGS with MPPI, CEM, GD, and matched behavior-cloning priors, together with configuration files, pinned dependencies, data instructions, and evaluation seeds. Appendix D states the theoretical assumptions and provides complete proofs of the theoretical results.

REFERENCES

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. ISSN 08834237, 21688745. URL http://www.jstor.org/stable/26408299.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023. doi: 10.1109/CVPR52729.2023.01499.

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/2506.09985.

Jiaxin Bai and Jiaxuan Xiong. Temporal-distance JEPA: Plan-aware representation learning for latent world model predictive control, 2026. URL https://arxiv.org/abs/2607.25337.

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

Ershad Banijamali, Rui Shu, Mohammad Ghavamzadeh, Hung Bui, and Ali Ghodsi. Robust locally-linear controllable embedding. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1751–1759. PMLR, 09–11 Apr 2018. URL https://proceedings.mlr.press/v84/banijamali18a.html.

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.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 277–286. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/bharadhwaj20a.html.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018. doi: 10.1214/17-AAP1326. URL https://projecteuclid.org/euclid.aoap/1520564481.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin C. M. Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/3de568f8597b94bda53149c7d7f5958c-Paper.pdf.

Andre Costa, Owen Dafydd Jones, and Dirk Kroese. Convergence properties of the cross-entropy method for discrete optimization. *Operations Research Letters*, 35(5):573–580, 2007. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2006.11.005. URL https://www.sciencedirect.com/science/article/pii/S0167637706001313.

Brandon Cui, Yinlam Chow, and Mohammad Ghavamzadeh. Control-aware representations for model-based reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dgd4EJqsbW5.

Pieter-Tjerk de Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

Matthieu Destrade, Oumayma Bounou, Quentin Le Lidec, Jean Ponce, and Yann LeCun. Value-guided action planning with jepa world models, 2025. URL https://arxiv.org/abs/2601.00844.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Benjamin Eysenbach, Vivek Myers, Ruslan Salakhutdinov, and Sergey Levine. Inference via interpolation: Contrastive representations provably enable planning and inference. In *Advances in Neural Information Processing Systems*, volume 37, pp. 58901–58928. Curran Associates, Inc., 2024. doi: 10.52202/079017-1878. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6c49d2ad55e50c5ebc1002fdc50e48e5-Paper-Conference.pdf.

Mahyar Fazlyab, Sina Sharifi, and Jiarui Wang. Model predictive path integral control as preconditioned gradient descent. *IEEE Control Systems Letters*, 10:2089–2094, 2026. doi: 10.1109/LCSYS.2026.3708745.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2170–2179. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/gelada19a.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

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

Nicklas Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Jiaming Hu, Yan Zheng, and Tian Wang. SCALE: State-calibrated latent embeddings for JEPA planning in the right geometry, 2026. URL https://arxiv.org/abs/2608.16287.

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

Hilbert J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11011, 2005. doi: 10.1088/1742-5468/2005/11/P11011.

Dirk P. Kroese, Sergey Porotsky, and Reuven Y. Rubinstein. The cross-entropy method for continuous multi-extremal optimization. *Method. Comput. Appl. Prob.*, 8(3):383–407, September 2006. ISSN 1387-5841. doi: 10.1007/s11009-006-9753-0. URL https://doi.org/10.1007/s11009-006-9753-0.

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

Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal InfoGAN. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/08aac6ac98e59e523995c161e57875f5-Paper.pdf.

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

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJxG_0EtDS.

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

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.

Lev Margolin. On the convergence of the cross-entropy method. *Annals of Operations Research*, 134 (1):201–214, 2005. doi: 10.1007/s10479-005-5731-0.

Nikolai Matni and Stephen Tu. A tutorial on concentration bounds for system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 3741–3749. IEEE Press, 2019. doi: 10.1109/CDC40024.2019.9029621. URL https://doi.org/10.1109/CDC40024.2019.9029621.

Heejeong Nam, Quentin Le Lidec, Lucas Maes, Yann LeCun, and Randall Balestriero. Causal-JEPA: Learning world models through object-level latent masking. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://arxiv.org/abs/2602.11389.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3962–3971, 2019. doi: 10.1109/CVPR.2019.00409.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1049–1065. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/pinneri21a.html.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Tankred Saanum, Peter Dayan, and Eric Schulz. Simplifying latent dynamics with softly state-invariant world models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 38355–38382. Curran Associates, Inc., 2024. doi: 10.52202/079017-1212. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/43ba0466af2b1ac76aa85d8fbec714e3-Paper-Conference.pdf.

Rui Shu, Tung Nguyen, Yinlam Chow, Tuan Pham, Khoat Than, Mohammad Ghavamzadeh, Stefano Ermon, and Hung Bui. Predictive coding for locally-linear control. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8862–8871. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/shu20a.html.

Uladzislau Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. In *Advances in Neural Information Processing Systems*, volume

38, Main Conference, pp. 43905–43941. Curran Associates, Inc., 2025. doi: 10.52202/085713-1465. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3e7cf447f21cd11c846463affefce665-Paper-Conference.pdf.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4732–4741. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/srinivas18b.html.

Junhan Sun, Hao Zhao, and Guofeng Zhang. INTACT: Isomorphic intent-to-action learning for search-free world models, 2026. URL https://arxiv.org/abs/2607.26056.

Basile Terver, Tsung-Yen Yang, Jean Ponce, Adrien Bardes, and Yann LeCun. What drives success in physical planning with joint-embedding predictive world models? *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=cHZn5Gdh8e. Reproducibility Certification.

Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. Learning policy improvements with path integrals. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 828–835, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/theodorou10a.html.

Leonardo F. Toso, Davit Shadunts, Yunyang Lu, Nihal Sharma, Donglin Zhan, Nam H. Nguyen, and James Anderson. Learning invariant visual representations for planning with joint-embedding predictive world models, 2026. URL https://arxiv.org/abs/2602.18639.

Elia Trevisan and Javier Alonso-Mora. Biased-MPPI: Informing sampling-based model predictive control by fusing ancillary controllers. *IEEE Robotics and Automation Letters*, 9(6):5871–5878, 2024. doi: 10.1109/LRA.2024.3397083.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1365–1374, 2019. doi: 10.1109/ICCV.2019.00145.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy P. Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. doi: 10.1016/j.simpa.2020.100022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36411–36430. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wang23al.html.

Ying Wang, Oumayma Bounou, Yann LeCun, and Mengye Ren. AdaJEPA: An adaptive latent world model, 2026a. URL https://arxiv.org/abs/2606.32026.

Ying Wang, Oumayma Bounou, Gaoyue Zhou, Randall Balestriero, Tim G. J. Rudner, Yann LeCun, and Mengye Ren. Temporal straightening for latent planning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026b.

Yuhai Wang, Jiawei Xia, Rongxuan Zhou, Xiao Hu, Yongliang Shi, Jing Du, and Yang Ye. PRISM: PRior-guided Imagination Sampling in world Models, 2026c. URL https://arxiv.org/abs/2606.07974.

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/a1afc58c6ca9540d057299ec3016d726-Abstract.html.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1714–1721, 2017. doi: 10.1109/ICRA.2017.7989202.

Ge Yang, Amy Zhang, Ari Morcos, Joelle Pineau, Pieter Abbeel, and Roberto Calandra. Plan2Vec: Unsupervised representation learning by latent plans. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pp. 935–946. PMLR, 10–11 Jun 2020. URL https://proceedings.mlr.press/v120/yang20b.html.

Zeji Yi, Chaoyi Pan, Guanqi He, Guannan Qu, and Guanya Shi. CoVO-MPC: Theoretical analysis of sampling-based MPC and optimal covariance design. In *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, volume 242 of *Proceedings of Machine Learning Research*, pp. 1122–1135. PMLR, 15–17 Jul 2024. URL https://proceedings.mlr.press/v242/yi24b.html.

Ming Yin, Yaqi Duan, Mengdi Wang, and Yu-Xiang Wang. Near-optimal offline reinforcement learning with linear representation: Leveraging variance information with pessimism. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KLaDXLAzzFT.

Hyung-Jin Yoon and Hunmin Kim. Finite-sample closed-loop stability of model predictive path integral control for linear time-invariant systems, 2026. URL https://arxiv.org/abs/2607.04006.

Hyung-Jin Yoon, Chuyuan Tao, Hunmin Kim, Naira Hovakimyan, and Petros Voulgaris. Sampling complexity of path integral methods for trajectory optimization. In *2022 American Control Conference (ACC)*, pp. 3482–3487, 2022. doi: 10.23919/ACC53348.2022.9867607.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, and Sergey Levine. SOLAR: Deep structured representations for model-based reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7444–7453. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19m.html.

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

Zhenghao Zhang, Yuanxiang Wang, Zhenyu Guan, Yujia Yang, Bingkang Shi, Tianyu Zong, Hongzhu Yi, Guoqing Chao, Xingchen Chen, Tiankun Yang, Chenxi Bao, Tao Yu, Jingjing Zhou, and Jungang Xu. Delta-JEPA: Learning action-sensitive world models via latent difference decoding, 2026b. URL https://arxiv.org/abs/2606.31232.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79115–79135. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/zhou25t.html.

APPENDIX CONTENTS

A Extended Related Work **17**

A.1 Latent world models and JEPA 17

A.2 Planner-friendly representations 17

A.3 Sampling-based MPC 18

A.4 Relational and Similarity-Preserving Knowledge Distillation 18

B Referenced Methods and Planning Algorithms **19**

B.1 Planning algorithms 19

B.2 Temporal Straightening 20

C Experiment Details and Additional Experimental Results **21**

C.1 Main-experiment implementation and evaluation 21

C.2 Visualizing Action–Effect Geometry and Planning 26

C.3 Full Action–Representation Probing 29

C.4 Mechanism-Figure Protocol and Additional Results 32

C.5 DINO-WM Implementation and Additional Results 33

C.6 Training Ablations 35

C.7 Shared MPPI Temperature Selection 35

C.8 Refinement Efficiency and Matched BC Priors on Other Environments 36

D Proofs and Extended Theory **39**

D.1 From CGS Loss to Transition Geometry 39

D.2 Planning Hessian and Dimension Cases 41

D.3 Quadratic Planning and Gaussian Proposals 44

D.4 MPPI: Mean Update and Planning Cost 46

D.5 CEM: Local Mean Update 49

D.6 Gradient Descent: Planning Cost Convergence 51

D.7 Geometry under Other Training Objectives 53

A Extended Related Work

A.1 Latent world models and JEPA

Latent world models compress visual observations into representations that support prediction, planning, and imagination. Reconstruction-based models combine learned dynamics with pixel decoders, while task-oriented models also learn rewards, values, or policies (Ha & Schmidhuber, 2018; Hafner et al., 2019; 2020; Hansen et al., 2022; 2024; Hafner et al., 2025). JEPAs learn through prediction in embedding space, removing the need to reconstruct every visual detail (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024). DINO-WM uses frozen pretrained image features, and V-JEPA 2-AC adapts video representations for action-conditioned prediction (Oquab et al., 2024; Zhou et al., 2025; Assran et al., 2025). PLDM and LeWM instead learn visual representations and dynamics jointly from offline control trajectories (Sobal et al., 2025; Maes et al., 2026).

Stable prediction and representation collapse. A constant encoder can trivially satisfy an embedding prediction loss, making collapse prevention a central design concern. Target encoders and stop-gradient operations stabilize predictive targets; variance–covariance regularization maintains feature diversity; and Gaussian distribution matching provides an explicit distributional constraint (Assran et al., 2023; Bardes et al., 2024; 2022; Balestriero & LeCun, 2025; Maes et al., 2026). Further architectures investigate object-level latent masking and sparse codes, while AdaJEPA adapts a trained world model at test time (Nam et al., 2026; Kuang et al., 2026; Wang et al., 2026a). These choices address representation content, training stability, and adaptation. The geometry exposed to downstream action optimization remains an additional concern: accurate transition prediction and diverse features alone do not ensure that the planning objective is easy to optimize. Empirical studies of JEPA planning and reachability-aware training examine this gap directly (Terver et al., 2026; Li et al., 2026). CGS addresses this concern through an auxiliary geometry objective applicable to end-to-end learned and pretrained representations.

A.2 Planner-friendly representations

Latent dynamics and optimization. Embed-to-Control, Robust Controllable Embedding, SOLAR, and predictive-coding approaches learn representations suited to locally linear control (Watter et al., 2015; Banijamali et al., 2018; Zhang et al., 2019; Levine et al., 2020; Shu et al., 2020). Here, *locally linear latent dynamics* means that transitions in a neighborhood or around a nominal trajectory admit a linear model whose coefficients may vary across states or time. This structure supports local trajectory optimization, including iLQR. Universal Planning Networks backpropagate through an unrolled planner; CARL couples representations to actor–critic updates and value-weighted model fitting (Srinivas et al., 2018; Cui et al., 2021). DeepMDP and bisimulation-based representations preserve reward and transition structure, while softly state-invariant models simplify action effects across states (Gelada et al., 2019; Zhang et al., 2021; Saanum et al., 2024). Other approaches learn feasible latent paths or distances reflecting directed reachability (Kurutach et al., 2018; Yang et al., 2020; Eysenbach et al., 2024; Wang et al., 2023).

Goal-derived and inverse-control supervision. Value-Guided JEPA learns goal-conditioned values; RC-aux adds multi-horizon prediction and budget-conditioned reachability labels; and Temporal-Distance-JEPA encodes directed temporal progress (Destrade et al., 2025; Li et al., 2026; Bai & Xiong, 2026). INTACT uses physical transition intent and future-goal intent to learn an action distribution that can be deployed directly (Sun et al., 2026). Future observations and temporal offsets in offline trajectories supply pseudo-goals or reachability targets for these objectives; these are constructed training signals rather than externally annotated goals. SCALE uses task-relevant state differences to calibrate latent distance (Hu et al., 2026). Sensorimotor World Models and Delta-JEPA use inverse dynamics to retain action information in endpoint embeddings or latent displacements (Ivashkov et al., 2026; Zhang et al., 2026b). Related work studies control-irrelevant visual invariance and the action excitation needed to identify controlled dynamics (Toso et al., 2026; Zhang et al., 2026a). The CGS world-model objective uses local transitions from pixel–action pairs to match action and latent-difference cosines, without future-goal targets or an inverse action decoder.

Temporal Straightening. Temporal Straightening aligns consecutive latent displacements by penalizing their angular deviation, reducing local trajectory curvature (Wang et al., 2026b). Straighter latent paths can improve planning-objective conditioning while preserving a nonlinear predictor. The

regularizer operates directly on consecutive observations, without pseudo-goal selection, value targets, reachability labels, or contrastive negative mining. TS constrains successive displacements along observed paths. CGS uses the corresponding actions as references for pairwise angular relations among latent differences. When consecutive actions align, the adjacent CGS term reduces to the squared angular TS penalty; changing action directions provides references for latent turns. Under the assumptions of our linear-dynamics analysis, CGS promotes temporal straightening, balanced action effects, and reduced drift–control coupling. These properties connect local transition geometry to more balanced terminal-cost curvature across the full planning horizon. Appendix B defines the TS objective, and Appendix C.5 describes its encoder instantiations and our DINO-WM comparisons.

A.3 Sampling-based MPC

Sampling-based MPC evaluates candidate action sequences through model rollouts and repeatedly updates a proposal distribution. CEM fits a distribution to low-cost elites, while MPPI uses cost-dependent weights over sampled sequences (Rubinstein, 1999; de Boer et al., 2005; Kroese et al., 2006; Kappen, 2005; Theodorou et al., 2010; Williams et al., 2017). These planners have been combined with probabilistic ensembles, stochastic latent dynamics, and predictive feature-space models (Chua et al., 2018; Hafner et al., 2019; Zhou et al., 2025; Sobal et al., 2025; Maes et al., 2026).

Proposals and sample reuse. iCEM combines temporally correlated sampling with elite memory and sample reuse; hybrid methods interleave CEM and gradient updates (Pinneri et al., 2021; Bharadhwaj et al., 2020). Biased-MPPI incorporates ancillary controllers, CoVO-MPC adapts covariance to local cost curvature, and PRISM learns a state–goal-conditioned proposal prior (Trevisan & Alonso-Mora, 2024; Yi et al., 2024; Wang et al., 2026c). These approaches use temporal structure, local cost information, or learned behavior to concentrate evaluations on promising action sequences. CGS complements proposal design by shaping the control geometry of the world model during training.

Finite-budget optimization. Classical CEM analyses study distribution updates and convergence (Margolin, 2005; Costa et al., 2007); recent MPPI analyses connect updates to preconditioned gradient descent and examine closed-loop stability (Fazlyab et al., 2026; Yoon & Kim, 2026). Importance-sampling results clarify how mismatch between proposals and high-weight regions affects the sample budget needed for reliable estimates (Agapiou et al., 2017; Chatterjee & Diaconis, 2018). In MPC, the horizon, effective action dimension, cost landscape, and proposal design jointly determine how informative a finite candidate set will be (Yoon et al., 2022). Our linear-dynamics analysis connects local representation learning to finite-horizon terminal-cost curvature, yielding finite-budget guarantees for MPPI, local contraction results for CEM, and convergence bounds for gradient descent.

A.4 Relational and Similarity-Preserving Knowledge Distillation

Relational Knowledge Distillation transfers inter-example distances and angles, while similarity-preserving knowledge distillation preserves pairwise similarities across teacher and student representations (Park et al., 2019; Tung & Mori, 2019). They are relevant to CGS through the general idea of matching structure across different representation spaces. CGS uses a related pairwise matching form, but observed actions provide the reference geometry rather than a teacher representation. Specifically, it matches pairwise action cosines to those among corresponding latent differences. CGS therefore uses structural matching to organize control geometry for multi-step planning; our analysis connects that geometry to more balanced terminal-cost curvature and sampling-efficient optimization.

B Referenced Methods and Planning Algorithms

This section defines the planning algorithms and the TS objective using the notation of Section 3. Experimental settings and the DINO-WM representation comparison are collected in Appendix C.

B.1 Planning algorithms

We write the planner updates using $C(\boldsymbol{a})$ from Eq. (3). In implementation, the cost and MPPI temperature use the matched normalizations specified in Appendix C.1.1; DINO-WM-specific reductions and proprioceptive terms are specified in Appendix C.5. Below, K is the number of evaluated candidates per optimization step, I the number of optimization steps (iterations), and $n_H = H \times d_a$ the flattened action-sequence dimension. The Gaussian proposal has mean μ_i and coordinatewise standard deviation $\sigma_{\mathrm{p},i}$. Sequences are flattened for Gaussian updates and reshaped for model rollout. Following the released LeWM planners, the candidate set contains the current mean and $K-1$ Gaussian draws (Maes et al., 2026).

Cross-Entropy Method (CEM). CEM refits a diagonal Gaussian to the lowest-cost elite set $\mathcal{E}_i$ (Rubinstein, 1999; de Boer et al., 2005). For $M = |\mathcal{E}_i| \geq 2$, the implementation uses the elite mean and sample variance,

$$\mu_{i+1} = \frac{1}{M} \sum_{k \in \mathcal{E}_i} \boldsymbol{a}^{(k)}, \qquad \sigma^2_{\mathrm{p},i+1} = \frac{1}{M-1} \sum_{k \in \mathcal{E}_i} (\boldsymbol{a}^{(k)} - \mu_{i+1})^2. \tag{13}$$

The square is coordinatewise. CEM starts from zero mean and unit standard deviation and returns the final mean. For $M = 1$, CEM updates the mean and retains the previous standard deviation, $\sigma_{\mathrm{p},i+1} = \sigma_{\mathrm{p},i}$; this keeps the proposal finite and allows continued sampling. Algorithm 1 includes this case.

Algorithm 1 CEM and prior-guided CEM

Require: Goal cost C, candidates K, optimization steps I, elites $1 \leq M \leq K$, optional prior mean μ_p

$\mu \leftarrow \mu_p$ for PG-CEM or 0 for CEM; $\sigma_{\mathrm{p}} \leftarrow \mathbf{1}$
for $i = 1, \ldots, I$ **do**
  $\boldsymbol{a}^{(1)} \leftarrow \mu$
  for $k = 2, \ldots, K$ **do**
    $\epsilon^{(k)} \sim \mathcal{N}(0, \mathrm{I}_{n_H})$; $\boldsymbol{a}^{(k)} \leftarrow \mu + \sigma_{\mathrm{p}} \odot \epsilon^{(k)}$
  $c_k \leftarrow C(\boldsymbol{a}^{(k)})$ for $k = 1, \ldots, K$
  $\mathcal{E} \leftarrow$ indices of the M smallest costs
  $\mu \leftarrow M^{-1} \sum_{k \in \mathcal{E}} \boldsymbol{a}^{(k)}$
  if $M > 1$ **then**
    $\sigma^2_{\mathrm{p}} \leftarrow (M-1)^{-1} \sum_{k \in \mathcal{E}} (\boldsymbol{a}^{(k)} - \mu)^2$
return μ

Model Predictive Path Integral (MPPI) control. We use the MPPI cost-weighted update (Kappen, 2005; Theodorou et al., 2010; Williams et al., 2017),

$$w_k = \frac{\exp[-C(\boldsymbol{a}^{(k)})/\tau]}{\sum_{j=1}^{K} \exp[-C(\boldsymbol{a}^{(j)})/\tau]}, \qquad \mu_{i+1} = \sum_{k=1}^{K} w_k \boldsymbol{a}^{(k)}. \tag{14}$$

All candidates contribute to the mean update, and the sampling standard deviation stays fixed at one. Smaller τ concentrates weight on lower-cost candidates. We share τ across training objectives and budgets within each task and backbone family; Appendix C.7 reports the temperature analysis.

Prior-guided CEM and MPPI (PG-CEM and PG-MPPI). A goal-conditioned behavior-cloning head π_ψ, trained for each frozen representation, predicts an action-sequence mean,

$$\mu_p = \pi_\psi(z_t, z_g) \in \mathbb{R}^{H \times d_a}. \tag{15}$$

The PG variants initialize $\mu_0 = \mu_p$ and retain the corresponding planner's unit initial standard deviation and refinement rule. This tests how a representation supports both a learned proposal

mean and subsequent model-based optimization. An accurate prior mean alone does not ensure a well-suited action-sequence distribution; CEM and MPPI continue to refine this distribution through model-based sampling, with CEM updating its mean and variance and MPPI updating its mean at fixed variance. Prior training is specified in Appendix C.1.1.

Algorithm 2 MPPI and prior-guided MPPI

Require: C, K, I, τ; optional prior mean μ_p
$\mu \leftarrow \mu_p$ for PG-MPPI or 0 for MPPI; $\sigma_{\mathrm{p}} \leftarrow \mathbf{1}$
for $i = 1, \ldots, I$ **do**
  $\boldsymbol{a}^{(1)} \leftarrow \mu$
  for $k = 2, \ldots, K$ **do**
    $\epsilon^{(k)} \sim \mathcal{N}(0, \mathrm{I}_{n_H})$; $\boldsymbol{a}^{(k)} \leftarrow \mu + \sigma_{\mathrm{p}} \odot \epsilon^{(k)}$
  $c_k \leftarrow C(\boldsymbol{a}^{(k)})$ for $k = 1, \ldots, K$
  $c_{\min} \leftarrow \min_k c_k$; $\widetilde{w}_k \leftarrow \exp[-(c_k - c_{\min})/\tau]$
  $w_k \leftarrow \widetilde{w}_k / \sum_j \widetilde{w}_j$
  $\mu \leftarrow \sum_{k=1}^{K} w_k \boldsymbol{a}^{(k)}$
return μ

Gradient-based planner (GD). Following the TS planning implementation (Wang et al., 2026b), GD directly optimizes the action sequence through the frozen differentiable world model. Starting from an initial action sequence $\boldsymbol{a}_0$, the rollout cost is minimized by iterative gradient-based updates,

$$\boldsymbol{a}_{i+1} = \boldsymbol{a}_i - \eta_i \nabla_{\boldsymbol{a}} C(\boldsymbol{a}_i), \tag{16}$$

where η_i denotes the optimization step size. In practice, we use Adam with cosine learning-rate decay, initialize from the standardized zero-control sequence, and inject no action noise. GD optimizes a single action trajectory and is independent with sampling budget K. Experimental settings and the per-solve cost construction are given in Appendix C.1.1.

B.2 Temporal Straightening

Temporal Straightening (TS) regularizes the curvature of latent trajectories by maximizing the cosine similarity between consecutive latent displacements (Wang et al., 2026b). Using the main context notation for global visual latents, define

$$\Delta z_t = z_{t+1} - z_t, \qquad v_t = \frac{\Delta z_t}{\|\Delta z_t\|_2}. \tag{17}$$

In implementation, we exclude adjacent pairs for which either latent displacement has norm below 10^{-6}. Let $\mathcal{T}$ denote the remaining valid pairs. TS uses the average cosine-based curvature penalty,

$$\mathcal{L}_{\mathrm{TS}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} (1 - v_t^\top v_{t+1}), \qquad \mathcal{L}_{\mathrm{LeWM+TS}} = \mathcal{L}_{\mathrm{LeWM}} + \lambda_{\mathrm{TS}} \mathcal{L}_{\mathrm{TS}}, \tag{18}$$

with $\lambda_{\mathrm{TS}} \geq 0$. Our LeWM adaptation retains the prediction and SIGReg terms in Eq. (7), with gradients through both branches of the prediction loss (Maes et al., 2026). The original TS implementation uses stop-gradient prediction targets (Wang et al., 2026b); its DINO-based instantiations and our CLS/Patch implementations are described in Appendix C.5. Although LeWM exhibits implicit temporal straightening on PushT (Maes et al., 2026, Appendix H), explicit TS can further reduce latent trajectory curvature and improve planning, with clearer planning gains on Cube and TwoRooms; see Appendix C.6.1.

C Experiment Details and Additional Experimental Results

C.1 Main-experiment implementation and evaluation

C.1.1 Experiment details

Environments and training. We evaluate LeWM on PushT, Cube, TwoRooms, and Reacher using offline pixel–action trajectories, 224×224 images, with 5 frame skips, and the original benchmark success predicates. The LeWM models and the TS and CGS variants follow the same LeWM (Maes et al., 2026) training setup and retain its backbones and architecture of ViT-Tiny/14 encoder, 192-dimensional projected CLS latent, and action-conditioned transformer predictor. Training uses observation encodings as predictor inputs and computes all N next-step predictions in one temporally masked forward pass using Eq. (2). One model per environment and method is fixed across all planners and planning budgets. For CGS, action-reference or latent-difference vectors with ℓ_2 norm below 10^{-6} are excluded from cosine normalization, and off-diagonal pairs involving such vectors are omitted from the loss average. On Cube, the CGS term uses the XYZ displacement components of the action as its geometry reference when the grasp command is active, while the full action is retained in the LeWM prediction objective; see Appendix C.1.3 for more discussions.

Regularization coefficients. We keep the base LeWM hyperparameters fixed across all LeWM-based methods, including $\lambda_{\mathrm{SIGReg}} = 0.09$, and vary only the coefficient introduced by each additional regularizer. For LeWM+TS, we sweep $\lambda_{\mathrm{TS}} \in \{0.01, 0.05, 0.1, 0.5\}$ and use 0.01 on PushT, 0.1 on Cube and Reacher, and 0.5 on TwoRooms. CGS uses $\lambda_{\mathrm{CGS}} = 0.5$ by default and 0.01 on PushT. All selected coefficients are fixed across planners and planning budgets. Ablations of the CGS ramp, history length, $\lambda_{\mathrm{SIGReg}}$ and λ_{CGS} are reported in Appendix C.6.

Planning and evaluation. During planning, the world model is frozen and predicted latents are fed back into the history for autoregressive rollout (Maes et al., 2026). Actions are optimized in the standardized coordinates used during training and converted to environment coordinates for execution. Each model step spans five environment steps. Each solve uses $H = 5$ model steps, corresponding to 25 environment steps, executes the resulting 25-step action sequence, and then replans from the new observation. Episodes contain at most 50 executed environment actions, giving two closed-loop solves. Initial observations are sampled from trajectories in the offline dataset, and goals are the observations 25 environment steps later in the same trajectory, $o_g = o_{t+25}$. This horizon, goal construction, and execution cadence are shared by all planners.

MPPI and CEM use $I = 30$ optimization steps and $K \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ candidates. MPPI weights all candidates, while CEM retains $\max(1, \lfloor K/4 \rfloor)$ elites. Each candidate set includes the current proposal mean; thus $K = 1$ evaluates only the zero or prior initialization. The single-elite CEM case follows Algorithm 1, updating the mean while retaining the previous sampling standard deviation. LeWM sampling costs use the squared terminal latent error. The MPPI temperatures are 4, 64, 128, and 64 on PushT, Cube, TwoRooms, and Reacher, respectively, and are fixed across training objectives and planning budgets. Appendix C.7 reports these sweeps; for each environment, we select a shared value that balances performance across the three methods. GD follows the gradient-based planning setup of Wang et al. (2026b), using 100 Adam steps, an initial learning rate of 0.1, cosine decay, zero-control initialization, and no injected action noise, to minimize terminal latent MSE.

Matched behavior-cloning priors. We train a separate lightweight three-layer MLP (Wang et al., 2026c) for each frozen representation to map (z_t, z_g) to the mean of a 25-step action sequence. Trajectory future observations provide the goals, which are the same as planning goals in evaluation, and checkpoints are selected by validation loss. The prior is trained only after freezing the representation and does not affect world-model training. PG-MPPI and PG-CEM initialize their proposal mean with this prediction and otherwise retain the corresponding planner's unit initial variance, cost, and refinement rule. Thus each representation is evaluated with its own matched prior.

C.1.2 Additional results

Statistics and sample-budget sweep results. All results report mean success rate and sample standard deviation over seeds $\{0, 1, 42\}$, with 50 episodes per seed and shared start indices across

methods. Planner-only timing includes both solves and excludes model loading, environment stepping, and rendering; the timing protocol is given in Appendix C.5. Tables 3–6 report the candidate sample-budget K results underlying Table 1 and Figure 4.

Table 3: **PushT sample-budget sweep.** Success rate (%, mean ± sample SD over three evaluation seeds, 50 episodes each). Bold marks the best representation for each planner and budget, including ties. GD is independent of K.

World model	Planner	$K=1$	$K=2$	$K=4$	$K=8$	$K=16$	$K=32$	$K=64$	$K=128$	$K=256$
LeWM	MPPI	**0.0** ±0.0	30.7 ±6.1	42.7 ±9.0	56.7 ±1.1	58.7 ±10.1	61.3 ±1.1	59.3 ±10.1	64.0 ±5.3	60.7 ±4.2
	PG-MPPI	84.7 ±6.4	82.0 ±4.0	82.7 ±6.1	77.3 ±5.0	76.0 ±6.0	78.7 ±6.4	72.7 ±8.3	72.0 ±6.0	62.0 ±3.5
	CEM	**0.0** ±0.0	**35.3** ±3.1	37.3 ±6.4	56.0 ±5.3	74.0 ±7.2	83.3 ±4.2	83.3 ±5.0	89.3 ±6.1	87.3 ±4.2
	PG-CEM	85.3 ±6.1	84.0 ±3.5	80.7 ±5.8	91.3 ±1.1	89.3 ±1.1	93.3 ±1.1	94.0 ±4.0	92.0 ±5.3	92.0 ±2.0
	GD					85.3 ±5.0				
LeWM + TS	MPPI	**0.0** ±0.0	38.0 ±5.3	40.7 ±6.4	50.0 ±5.3	51.3 ±1.1	56.7 ±5.0	65.3 ±3.1	64.7 ±8.3	66.7 ±7.0
	PG-MPPI	84.0 ±5.3	84.0 ±6.0	82.7 ±5.8	78.0 ±2.0	76.7 ±3.1	78.7 ±3.1	72.7 ±7.6	69.3 ±9.0	70.7 ±3.1
	CEM	**0.0** ±0.0	28.0 ±2.0	34.7 ±10.1	55.3 ±6.4	73.3 ±6.1	85.3 ±3.1	**89.3** ±5.0	86.7 ±5.0	89.3 ±3.1
	PG-CEM	84.0 ±5.3	89.3 ±1.1	86.0 ±3.5	89.3 ±7.6	**94.0** ±4.0	**94.7** ±3.1	92.0 ±2.0	90.0 ±5.3	91.3 ±4.2
	GD					82.7 ±5.8				
LeWM + CGS	MPPI	**0.0** ±0.0	**40.0** ±5.3	**52.0** ±2.0	**60.7** ±8.1	**68.7** ±4.2	**67.3** ±6.1	**72.0** ±2.0	**77.3** ±6.4	**75.3** ±7.6
	PG-MPPI	**88.0** ±7.2	**89.3** ±4.6	**88.0** ±5.3	**87.3** ±6.1	**86.7** ±1.1	**84.7** ±5.8	**85.3** ±1.1	**80.0** ±2.0	**84.0** ±9.2
	CEM	**0.0** ±0.0	**35.3** ±1.1	**42.7** ±1.1	**68.7** ±5.0	**83.3** ±4.2	**86.0** ±4.0	88.7 ±4.6	**92.0** ±3.5	**90.7** ±5.8
	PG-CEM	**88.0** ±7.2	**90.7** ±6.1	**86.7** ±5.0	**96.0** ±4.0	**94.0** ±2.0	94.0 ±5.3	**96.0** ±2.0	**96.0** ±0.0	**96.0** ±4.0
	GD					**89.3** ±6.4				

Table 4: **Cube sample-budget sweep.** Success rate (%, mean ± sample SD over three evaluation seeds, 50 episodes each). Bold marks the best representation for each planner and budget, including ties. GD is independent of K.

World model	Planner	$K=1$	$K=2$	$K=4$	$K=8$	$K=16$	$K=32$	$K=64$	$K=128$	$K=256$
LeWM	MPPI	**44.7** ±7.0	46.0 ±5.3	44.7 ±11.0	46.7 ±6.4	51.3 ±4.2	51.3 ±5.0	56.7 ±5.0	59.3 ±4.6	64.0 ±9.2
	PG-MPPI	**100.0** ±0.0	**75.3** ±8.3	**70.0** ±5.3	60.7 ±4.2	61.3 ±6.1	76.7 ±2.3	82.7 ±6.4	89.3 ±4.2	90.0 ±2.0
	CEM	**44.7** ±7.0	**50.0** ±3.5	45.3 ±10.1	**55.3** ±9.0	57.3 ±7.0	62.7 ±10.1	64.0 ±6.0	63.3 ±2.3	64.0 ±11.1
	PG-CEM	**100.0** ±0.0	94.0 ±0.0	88.0 ±5.3	91.3 ±3.1	**94.7** ±1.1	89.3 ±4.2	88.0 ±4.0	90.0 ±2.0	89.3 ±1.1
	GD					59.3 ±8.1				
LeWM + TS	MPPI	**44.7** ±7.0	**48.0** ±5.3	**46.7** ±3.1	48.0 ±2.0	46.0 ±8.7	56.7 ±9.0	68.0 ±5.3	75.3 ±5.0	75.3 ±6.1
	PG-MPPI	98.0 ±2.0	65.3 ±5.0	65.3 ±10.3	62.7 ±14.5	**74.7** ±5.0	84.7 ±7.6	87.3 ±2.3	92.7 ±3.1	93.3 ±1.1
	CEM	**44.7** ±7.0	48.7 ±4.2	47.3 ±1.1	54.0 ±5.3	61.3 ±4.6	64.7 ±7.6	71.3 ±6.4	73.3 ±10.3	73.3 ±9.9
	PG-CEM	98.0 ±2.0	92.7 ±4.6	86.7 ±5.0	88.7 ±4.2	89.3 ±3.1	88.7 ±3.1	88.7 ±1.1	90.0 ±2.0	90.7 ±3.1
	GD					65.3 ±4.6				
LeWM + CGS	MPPI	**44.7** ±7.0	44.0 ±4.0	**46.7** ±9.4	**50.0** ±7.2	**54.0** ±8.0	**64.0** ±6.9	**69.3** ±9.9	**79.3** ±4.6	**82.7** ±6.1
	PG-MPPI	**100.0** ±0.0	60.0 ±4.0	62.7 ±7.6	**64.7** ±9.9	72.7 ±3.1	**86.7** ±3.1	**92.7** ±2.3	**96.7** ±3.1	**98.7** ±1.1
	CEM	**44.7** ±7.0	48.7 ±4.2	**48.7** ±2.3	**55.3** ±4.2	**64.0** ±10.4	**68.7** ±5.8	**75.3** ±11.6	**76.7** ±9.5	**82.0** ±6.0
	PG-CEM	**100.0** ±0.0	**96.0** ±2.0	**95.3** ±3.1	**92.0** ±4.0	90.7 ±5.0	**94.7** ±2.3	**93.3** ±1.1	**96.0** ±4.0	**95.3** ±4.2
	GD					**71.3** ±4.6				

Table 5: **TwoRooms sample-budget sweep.** Success rate (%, mean ± sample SD over three evaluation seeds, 50 episodes each). Bold marks the best representation for each planner and budget, including ties. GD is independent of K.

World model	Planner	$K=1$	$K=2$	$K=4$	$K=8$	$K=16$	$K=32$	$K=64$	$K=128$	$K=256$
LeWM	MPPI	**13.3** ±1.1	40.7 ±7.0	53.3 ±7.0	60.0 ±10.6	70.7 ±11.7	82.0 ±10.0	88.7 ±4.2	92.0 ±3.5	94.7 ±1.1
	PG-MPPI	**100.0** ±0.0	77.3 ±9.9	68.7 ±4.2	79.3 ±1.1	87.3 ±4.6	91.3 ±6.4	**98.7** ±1.1	98.0 ±2.0	**100.0** ±0.0
	CEM	**13.3** ±1.1	55.3 ±9.0	57.3 ±6.4	56.7 ±3.1	68.0 ±5.3	79.3 ±4.2	79.3 ±9.0	76.7 ±10.1	78.0 ±5.3
	PG-CEM	**100.0** ±0.0	98.0 ±3.5	97.3 ±2.3	99.3 ±1.1	96.0 ±4.0	96.7 ±2.3	96.0 ±3.5	93.3 ±3.1	94.7 ±2.3
	GD					41.3 ±7.0				
LeWM + TS	MPPI	**13.3** ±1.1	**56.0** ±3.5	**64.0** ±9.2	74.7 ±1.1	81.3 ±1.1	88.0 ±5.3	95.3 ±4.2	**97.3** ±3.1	96.0 ±2.0
	PG-MPPI	**100.0** ±0.0	78.7 ±4.2	69.3 ±6.4	78.0 ±4.0	84.7 ±7.0	92.0 ±2.0	95.3 ±5.0	98.0 ±2.0	98.0 ±2.0
	CEM	**13.3** ±1.1	**68.7** ±8.1	68.7 ±7.6	**78.0** ±2.0	**88.7** ±5.0	**95.3** ±3.1	**96.0** ±2.0	**97.3** ±1.2	**96.7** ±2.3
	PG-CEM	**100.0** ±0.0	**98.7** ±1.1	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
	GD					65.3 ±9.9				
LeWM + CGS	MPPI	**13.3** ±1.1	54.0 ±2.0	62.0 ±11.1	**75.3** ±9.4	**82.0** ±2.0	**92.7** ±6.1	**96.0** ±2.0	**97.3** ±1.1	**98.0** ±2.0
	PG-MPPI	**100.0** ±0.0	**86.7** ±6.1	**78.7** ±7.6	**80.0** ±4.0	**90.7** ±2.3	**94.0** ±3.5	98.0 ±3.5	**100.0** ±0.0	**100.0** ±0.0
	CEM	**13.3** ±1.1	64.0 ±6.0	**71.3** ±8.1	76.0 ±5.3	88.0 ±2.0	93.3 ±2.3	94.0 ±2.0	95.3 ±1.1	96.0 ±2.0
	PG-CEM	**100.0** ±0.0	98.0 ±2.0	98.7 ±1.1	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0	**100.0** ±0.0
	GD					**70.7** ±3.1				

Table 6: **Reacher sample-budget sweep.** Success rate (%, mean ± sample SD over three evaluation seeds, 50 episodes each). Bold marks the best representation for each planner and budget, including ties. GD is independent of K.

World model	Planner	$K=1$	$K=2$	$K=4$	$K=8$	$K=16$	$K=32$	$K=64$	$K=128$	$K=256$
LeWM	MPPI	**2.0** ±2.0	46.0 ±7.2	**40.7** ±8.3	**44.7** ±3.1	36.7 ±9.0	36.7 ±6.1	49.3 ±11.4	65.3 ±4.2	**73.3** ±4.6
	PG-MPPI	**78.7** ±2.3	**67.3** ±5.0	**49.3** ±1.1	46.0 ±7.2	42.7 ±5.0	46.0 ±5.3	46.7 ±10.1	62.7 ±11.0	**72.0** ±7.2
	CEM	**2.0** ±2.0	**44.0** ±10.0	**41.3** ±4.2	65.3 ±5.8	**72.0** ±5.3	74.7 ±9.2	**83.3** ±7.0	**83.3** ±8.3	85.3 ±3.1
	PG-CEM	**78.7** ±2.3	**82.0** ±2.0	**73.3** ±6.1	**79.3** ±4.2	**85.3** ±3.1	80.0 ±3.5	81.3 ±9.4	**86.7** ±6.1	86.7 ±10.1
	GD					76.7 ±4.2				
LeWM + TS	MPPI	**2.0** ±2.0	**47.3** ±8.1	40.0 ±10.6	36.7 ±9.9	**44.7** ±4.2	40.0 ±9.2	53.3 ±6.1	64.0 ±2.0	68.7 ±6.4
	PG-MPPI	66.7 ±11.0	58.0 ±6.0	46.7 ±9.9	41.3 ±7.6	38.7 ±9.0	44.7 ±3.1	52.7 ±5.0	66.7 ±2.3	67.3 ±7.6
	CEM	**2.0** ±2.0	42.7 ±5.0	37.3 ±1.1	62.7 ±3.1	70.7 ±3.1	80.0 ±8.7	77.3 ±5.0	81.3 ±8.1	81.3 ±7.0
	PG-CEM	66.7 ±11.0	74.7 ±9.9	66.7 ±5.0	78.7 ±13.0	80.0 ±7.2	79.3 ±3.1	**82.0** ±8.7	78.0 ±2.0	84.7 ±5.0
	GD					81.3 ±5.0				
LeWM + CGS	MPPI	**2.0** ±2.0	46.0 ±2.0	39.3 ±3.1	38.0 ±13.1	41.3 ±4.2	**42.7** ±12.2	**56.0** ±10.0	**66.7** ±2.3	70.7 ±2.3
	PG-MPPI	58.7 ±6.1	64.7 ±1.1	47.3 ±7.0	**46.7** ±7.0	**46.0** ±2.0	**46.7** ±4.2	**53.3** ±12.9	**69.3** ±3.1	67.3 ±7.0
	CEM	**2.0** ±2.0	39.3 ±1.1	38.7 ±9.4	**66.7** ±3.1	71.3 ±15.0	**81.3** ±4.6	82.0 ±2.0	80.7 ±3.1	**86.7** ±4.2
	PG-CEM	58.7 ±6.1	66.7 ±7.0	60.0 ±5.3	74.0 ±6.0	80.7 ±6.4	**80.7** ±2.3	79.3 ±10.1	81.3 ±3.1	**88.0** ±6.9
	GD					**82.0** ±5.3				

Aggregate performance across sample budgets. In order to comprehensively compare planning performance across sample-budget sweep result, we use an integration metric to summarize success rate all across sample-budget K's. Normalized AUC (nAUC) uses trapezoidal integration divided by the horizontal span:

$$\mathrm{nAUC}_x = \frac{\sum_{j=1}^{m-1}(x_{j+1}-x_j)(s_j+s_{j+1})/2}{x_m - x_1}.$$

Here s_j is SR in percent. We use $x = \log_2 K$ for sample budgets, nAUC ranges from 0 to 100. In Appendix C.8 and C.7, we also show nAUC results for refinement SR curve and temperature selection results, with $x = I$ being planning optimization steps, and $x = \log_2 \tau$ being temperature. Integration is linear, so integrating the mean curve equals averaging per-seed integrals. Updated Reacher entries use the reported rounded mean curves.

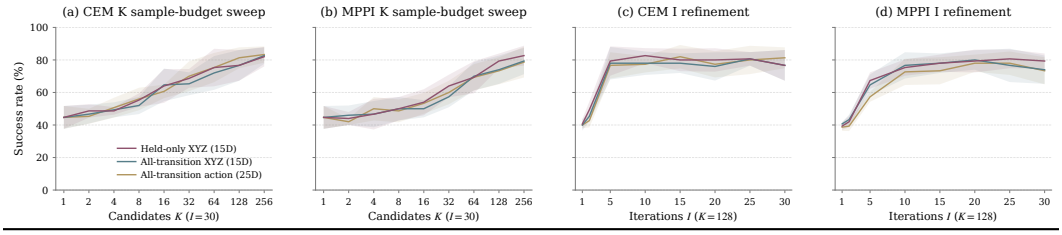
Table 7: **Complete sample-budget sweep nAUC.** $K \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$; sampling planners use $I = 30$. nAUC uses $\log_2 K$. Parentheses give percentage-point differences from LeWM for the same environment and planner. Bold marks the best representation for each planner and environment. Results average three evaluation seeds (50 episodes each); GD repeats its K-independent SR.

Planner	Model	PushT	Cube	TwoRooms	Reacher
CEM	LeWM	62.79	56.54	64.79	**63.46**
	LeWM + TS	62.17 (−0.62)	59.96 (+3.42)	**80.96** (+16.17)	61.71 (−1.75)
	LeWM + CGS	**67.75** (+4.96)	**62.58** (+6.04)	79.58 (+14.79)	63.04 (−0.42)
PG-CEM	LeWM	89.17	91.25	96.75	**81.33**
	LeWM + TS	90.38 (+1.21)	89.88 (−1.38)	**99.83** (+3.08)	76.88 (−4.46)
	LeWM + CGS	**93.17** (+4.00)	**94.46** (+3.21)	99.58 (+2.83)	74.50 (−6.83)
MPPI	LeWM	50.46	51.29	67.67	44.63
	LeWM + TS	50.00 (−0.46)	56.08 (+4.79)	76.42 (+8.75)	45.17 (+0.54)
	LeWM + CGS	**59.46** (+9.00)	**58.88** (+7.58)	**76.88** (+9.21)	**45.79** (+1.17)
PG-MPPI	LeWM	76.83	76.38	87.58	54.50
	LeWM + TS	77.42 (+0.58)	78.54 (+2.17)	86.88 (−0.71)	51.96 (−2.54)
	LeWM + CGS	**85.92** (+9.08)	**79.42** (+3.04)	**91.00** (+3.42)	**54.62** (+0.12)
GD	LeWM	85.33	59.33	41.33	76.67
	LeWM + TS	82.67 (−2.67)	65.33 (+6.00)	65.33 (+24.00)	81.33 (+4.67)
	LeWM + CGS	**89.33** (+4.00)	**71.33** (+12.00)	**70.67** (+29.33)	**82.00** (+5.33)

C.1.3 Cube ablation of reference geometry in CGS.

Each Cube environment action is five-dimensional, consisting of XYZ end-effector displacement, a grasp command, and an arm-angle command. Because one model step concatenates five environment actions, the predictor receives a 25D action block, while the corresponding XYZ translation channels form a 15D reference. We compare three CGS reference geometries: *grasp-active XYZ* (15D), which

uses only transitions in which the grasp command is active; *all-transition XYZ* (15D), which uses the same translation channels on all transitions; and *all-transition full-action* (25D), which uses all five action coordinates. In the main context (Section 5.1), we use grasp-active XYZ because XYZ commands most directly specify cube motion while the cube is grasped, whereas ungrasped transitions and the grasp/arm-angle channels need not induce comparable object displacement. The full 25D action block is nevertheless retained as predictor input in all variants. The sample-budget sweep uses $I = 30$; the refinement sweep uses $K = 128$. MPPI uses $\tau = 64$ throughout.



Reference	CEM (K)	MPPI (K)	CEM (I)	MPPI (I)
grasp-active XYZ15D (main)	62.58	**58.88**	**77.40**	**73.72**
All-transition XYZ15D	61.29 (−1.29)	57.00 (−1.88)	75.09 (−2.31)	72.67 (−1.06)
All-transition full25D	**62.92** (+0.33)	57.29 (−1.58)	75.82 (−1.59)	69.68 (−4.05)

Figure 7: **Cube reference geometry: sample-budget and refinement curves with nAUC summaries.** Lines show mean success and bands show sample SD over three evaluation seeds (50 episodes each). The table reports mean normalized AUC (nAUC) over the complete sample-budget and refinement sweeps: $K = 1$–256 on a $\log_2 K$ axis and $I = 1$–30 on a linear I axis. Higher is better; bold marks the best reference for each planner and sweep, including ties. Parentheses give percentage-point differences from grasp-active XYZ15D for the same sweep and planner; negative values indicate decreases. Both planners use no action prior.

grasp-active XYZ has the highest nAUC for MPPI in sweeps and for CEM in the refinement sweep. The all-transition full-action reference has the highest nAUC for CEM in the sample-budget sweep. On Cube, the grasp-active mask and XYZ reference are derived entirely from the observed action vector and require no additional state or task labels. Our full-action ablation shows that CGS remains effective without this reference selection.

C.1.4 State-informed reference geometry and observation history on Reacher

We separate two factors that can limit an action-only reference on Reacher: raw torque geometry does not fully describe the state-dependent physical effect of an action, and a single image does not fully specify the dynamical state. We compare LeWM, the main CGS model using the action reference (action-CGS), and a transition-reference CGS variant (transition-CGS). Transition-CGS uses a state-informed reference constructed from observed joint-configuration displacements, capturing the realized transition under the executed action and current state. Each checkpoint is frozen and evaluated under the same three observation-history protocols.

Transition-reference training. Action-CGS uses the executed action block a_t to define reference geometry. Transition-CGS instead uses the observed joint-configuration displacement, which reflects the transition produced by that action in its starting state:

$$\psi(q_t) = [\sin q_{t,1}, \sin q_{t,2}, \cos q_{t,1}, \cos q_{t,2}]^\top, \qquad r_t = W[\psi(q_{t+1}) - \psi(q_t)], \tag{19}$$

where q_t denotes the two joint angles, i.e., the configuration component of the Reacher probe state s_t. The index t follows model steps, so consecutive model-step observations are separated by five environment steps. W is a fixed diagonal normalization computed from training displacements; the sine and cosine coordinates of each joint share an inverse RMS scale. The reference r_t therefore reflects the realized state-dependent transition rather than the executed torque vector alone, with no learned aggregation head or encoder teacher. Only the reference construction changes: the architecture, prediction and SIGReg losses, the default $N = 3$ history setting, the cosine-Gram CGS objective, and all other training settings are unchanged. Joint-angle labels are used only to construct the training reference; planning uses images and actions without access to these labels.

Observation-history protocols. *Single-frame context (native)* supplies only the current image at every replanning call, with no previous observations or executed actions. Predicted latents and candidate actions still accumulate causally within each imagined rollout; we use this historical strategy in most of our experiments unless explained.

Dataset-initialized history (privileged) supplies $[o_{t-2}, o_{t-1}, o_t]$ and the two connecting executed-action blocks, $[a_{t-2}, a_{t-1}]$. At the first solve, the preceding observations and actions come from the same dataset episode used to initialize the environment. Missing episode-start observations are padded with the first frame and missing actions with raw zero actions before normalization; history never crosses episode boundaries. All later history comes from the actual online trajectory. These initial observations and actions are unavailable to a planner starting from an arbitrary reset with only one image, so we treat dataset-initialized history as a privileged-context diagnostic rather than a deployable cold-start protocol.

Online history reads no dataset prehistory and starts through the native single-frame path. It records observations and executed actions, enabling the same three-frame context after at least two model steps have been collected. In our evaluation, each solve executes five action blocks, so the second solve uses $[o_3, o_4, o_5]$ and $[a_3, a_4]$, with time measured from the reset. Both the native and online-history protocols are feasible under the single-image cold-start setting. In either history-based protocol, past actions are aligned with the observed transitions and candidate actions begin at the current time; the predictor retains its causal mask and at most three context tokens during rollout. No model parameters are updated online.

Evaluation and results. We follow the shared sampling-based and GD evaluation protocols in Appendices C.1.1, using matched evaluation starts for seeds $\{0, 1, 42\}$ and 50 episodes per seed. Sampling-based evaluations use no action prior. Across this ablation, only the CGS reference and observation-history protocol vary.

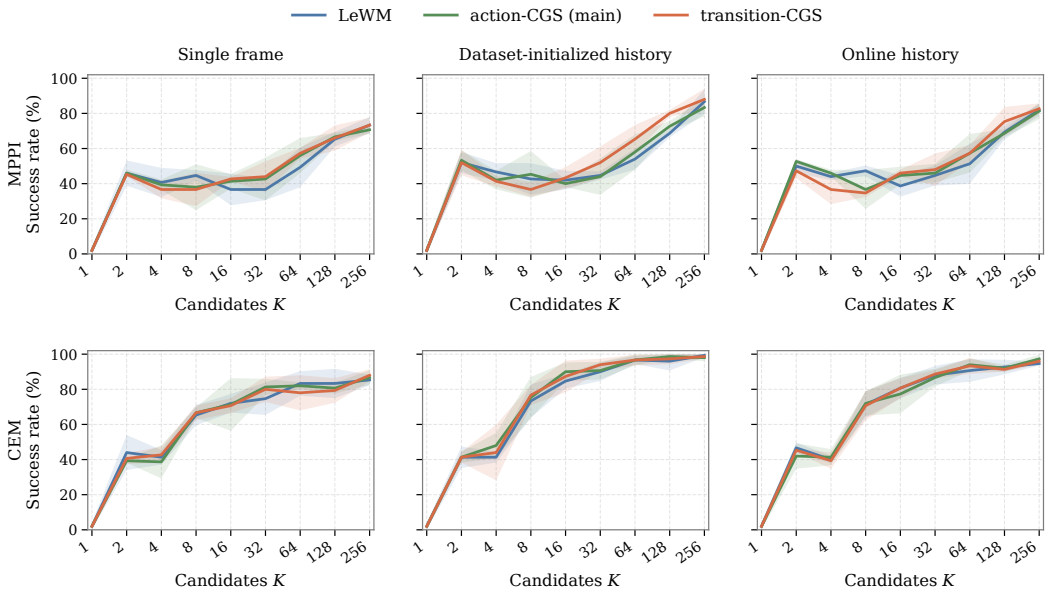


Figure 8: **Reacher sample-budget sweeps under three observation-history protocols.** Top: MPPI; bottom: CEM. Columns show single-frame input, dataset-initialized history, and online history, respectively. All curves use $I = 30$ without an action prior; bands show $\pm$ one sample SD across three evaluation seeds, 50 episodes each. MPPI uses $\tau = 64$. State labels are used only to train transition-CGS.

Figure 8 and Table 8 separate the effects of reference geometry from those of observation history. For MPPI, the three models are similar under the native single-frame protocol: at $K = 128$, LeWM, action-CGS, and transition-CGS reach 65.3%, 66.7%, and 66.0%, respectively. The state-informed reference becomes more useful when recent observation–action history provides dynamical context.

With dataset-initialized history, transition-CGS reaches 80.0% at $K = 128$, compared with 68.7% for LeWM and 72.7% for action-CGS, and also gives the highest sample-budget nAUC (51.96 versus 49.38 and 49.75). With online history, its advantage becomes clearer at larger candidate budgets, reaching 75.3% at $K = 128$, compared with 69.3% for LeWM and 68.7% for action-CGS. Thus, for MPPI, the benefit of a state-dependent transition reference becomes clearer when short transition history makes the current dynamics more observable, particularly at larger sampling budgets.

CEM shows a different pattern. Adding history substantially improves the LeWM baseline itself: its $K = 128$ success rises from 83.3% with a single frame to 96.0% with dataset-initialized history and 92.7% with online history. With history available, CEM approaches saturation at larger candidate budgets (Figure 8), leaving limited headroom for either reference variant to provide further gains. Accordingly, the additional differences among LeWM, action-CGS, and transition-CGS are small and mixed in both $K = 128$ success and sample-budget nAUC. GD exhibits a more consistent reference effect. Action-CGS improves over LeWM in all three history settings, and transition-CGS further increases mean success to 85.3%, 94.0%, and 92.7% under single-frame, dataset-initialized, and online history, respectively, compared with 76.7%, 84.0%, and 81.3% for LeWM.

Together, these results suggest that observation history and reference geometry address distinct limitations on Reacher. History provides information about recent dynamics that is absent from a single image, whereas transition-CGS supervises latent geometry using the realized, state-dependent effect of an action. Their combination is especially useful for MPPI and GD, while CEM already benefits strongly from history alone. Transition-CGS, however, requires joint-state information to construct its training reference. Such labels are privileged and may be unavailable in the pixel–action datasets targeted by our main setting. We therefore retain action-CGS as the main method and use transition-CGS as a diagnostic of the potential benefit of richer state-dependent reference geometries.

Table 8: **Reacher reference and history ablation: success rate and sample-budget nAUC.** SR is mean ± sample SD (%) over three evaluation seeds, 50 episodes each. MPPI/CEM SR uses $K = 128$, $I = 30$; GD uses 100 Adam steps and is independent of K. nAUC summarizes the complete sample-budget sweep. Parentheses give percentage-point differences from LeWM for both CGS models, within the same history, planner, and metric. Bold marks the highest value within each history/metric group. MPPI uses $\tau = 64$.

History	Model	MPPI		CEM		GD
		SR	nAUC	SR	nAUC	SR
Single frame	LeWM	65.3 ±4.2	44.63	**83.3** ±8.3	**63.46**	76.7 ±4.2
	action-CGS	**66.7** ±2.3 (+1.3)	**45.79** (+1.17)	80.7 ±3.1 (−2.7)	63.04 (−0.42)	82.0 ±5.3 (+5.3)
	transition-CGS	66.0 ±7.2 (+0.7)	**45.79** (+1.17)	79.3 ±7.0 (−4.0)	62.88 (−0.58)	**85.3** ±7.0 (+8.7)
Dataset-initialized history	LeWM	68.7 ±2.3	49.38	96.0 ±5.3	71.75	84.0 ±3.5
	action-CGS	72.7 ±3.1 (+4.0)	49.75 (+0.37)	**98.7** ±1.1 (+2.7)	**73.83** (+2.08)	90.7 ±4.2 (+6.7)
	transition-CGS	**80.0** ±2.0 (+11.3)	**51.96** (+2.58)	97.3 ±3.1 (+1.3)	73.46 (+1.71)	**94.0** ±3.5 (+10.0)
Online history	LeWM	69.3 ±4.2	48.42	**92.7** ±4.2	**69.79**	81.3 ±2.3
	action-CGS	68.7 ±2.3 (−0.7)	**49.21** (+0.79)	92.0 ±2.0 (−0.7)	69.37 (−0.42)	89.3 ±4.2 (+8.0)
	transition-CGS	**75.3** ±8.3 (+6.0)	48.46 (+0.04)	91.3 ±1.1 (−1.3)	**69.79** (+0.00)	**92.7** ±5.0 (+11.3)

C.2 Visualizing Action–Effect Geometry and Planning

Figure 9 shows two selected examples per environment, complementing the planning results with two views of control geometry: encoded transition trajectories and predicted endpoint responses to action perturbations.

C.2.1 Action and latent transition trajectories

Construction. For PushT, Cube, and Reacher, we sample candidate starts from recorded trajectories and select two illustrative 30-observation windows per benchmark at a five-step stride, retaining all 29 transitions. Because recorded TwoRooms episodes are shorter than this 145-step span, we use seeded native rollouts. Its two examples share a zigzag control family; a local encoder-geometry probe selects initial regions, and seeds determine control orientation and size. All are selected qualitative examples, not unfiltered random samples.

For each window, we form standardized five-step action blocks a_t and latent differences $\Delta z_t = f_\theta(o_{t+1}) - f_\theta(o_t)$, then normalize each vector in its original space as in Eq. (4), and cumulatively sum the unit directions into action and latent paths. We align latent to action directions (using a rigid cumulative-path fit for Cube) and project both paths onto action-path principal components: two for PushT, TwoRooms, and Reacher and three for Cube. Cube uses the XYZ action channels. The basis is shared across models; alignment has no scale, and projected steps are not renormalized.

Interpretation and results. The purple dashed curve is the action reference; colored curves are encoded transition paths. The key criterion is preservation of turns and relative directions: under CGS, $K_Z \approx K_A$ means that similarly directed actions induce similarly directed latent changes. Because the objective matches pairwise angles rather than coordinates or step lengths, exact projected overlap is a stronger criterion than the training loss imposes.

On **PushT and Cube**, CGS follows the action-reference shape more faithfully than LeWM or TS, particularly around changes of direction. TS is partially improved or comparable to LeWM but does not consistently recover these turns. On simpler **TwoRooms**, TS and CGS are similar and both align much better than LeWM, consistent with their near-ceiling MPPI SR (96.0% and 98.0% at $K = 256$; Table 5). On Reacher, CGS preserves several turns more clearly than LeWM or TS, though less consistently than on PushT or Cube, suggesting partial improvement in the action-related organization of this more state-dependent setting.

C.2.2 Action-perturbation heatmaps and planning isotropy

Construction and connection to G_H. At a shared observation and base action sequence $\mathbf{a}$, let $\widehat{F}_H(\mathbf{a})$ be the model's predicted terminal latent for $H = 5$ action blocks. Let $Q \in \mathbb{R}^{(Hd_a)\times 2}$ have orthonormal columns spanning the fixed horizon-constant plane of the first two primitive control coordinates. We sample 32 equally spaced unit-circle directions $q_j = (\cos\theta_j, \sin\theta_j)^\top$, with $\theta_j = 2\pi j/32$, and apply equal-norm perturbations $\epsilon Q q_j$ in standardized action coordinates, using $\epsilon = 0.05$. The heatmap records pairwise cosines of the full-dimensional terminal responses

$$r_j = \widehat{F}_H(\mathbf{a} + \epsilon Q q_j) - \widehat{F}_H(\mathbf{a}), \qquad K^{\mathrm{resp}}_{jk} = \frac{r_j^\top r_k}{\|r_j\|_2 \, \|r_k\|_2}. \tag{20}$$

The first PushT example uses frame 13 as its response anchor; every other example uses frame 1. Each comparison uses the same anchor, base actions, and perturbations for all three models.

Writing $J_H = \partial \widehat{F}_H / \partial \mathbf{a}$, local linearization gives $r_j \approx \epsilon J_H Q q_j$. Consequently,

$$K^{\mathrm{resp}}_{jk} \approx \frac{q_j^\top M_Q q_k}{\sqrt{q_j^\top M_Q q_j}\sqrt{q_k^\top M_Q q_k}}, \qquad M_Q = Q^\top J_H^\top J_H Q. \tag{21}$$

For the linear dynamics in Section 4.2, $J_H = \Gamma_H$ and hence $M_Q = Q^\top G_H Q$; for a nonlinear predictor, $J_H^\top J_H$ is the Gauss–Newton component of the planning Hessian (Eq. (69)). Under ideal CGS geometry, $G_H = cHP_H$ and $P_H Q = Q$, giving $M_Q = cHI_2$. Equal curvature in the tested plane then preserves the circle's angular relations: $K^{\mathrm{resp}}_{jk} \approx q_j^\top q_k = \cos(\theta_j - \theta_k)$. Thus similarity to the ideal heatmap is a local diagnostic of planning isotropy in this plane, rather than a measurement of full-space isotropy or of the nonlinear residual Hessian.

Interpretation and results. The *Ideal* tile is the cosine Gram matrix of a standard unit circle: nearby directions have cosine close to 1, orthogonal directions have cosine 0, and opposite directions have cosine -1. Its regular diagonal bands provide the reference for every model, with a common color range $[-1, 1]$. On **PushT and Cube**, CGS retains these bands more faithfully than LeWM and TS, whose distorted bands reveal uneven angular sensitivity. On **TwoRooms**, TS and CGS are close to one another and closer to the circle reference than LeWM, consistent with their similar trajectory geometry and near-ceiling SR. On Reacher, the response patterns remain less regular than on PushT or Cube, but the displayed CGS heatmaps recover the ideal band structure more closely than LeWM or TS. This indicates partial improvement in local planning sensitivity despite the stronger state dependence of the task, consistent with the smaller and more planner-dependent gains observed in the planning results.

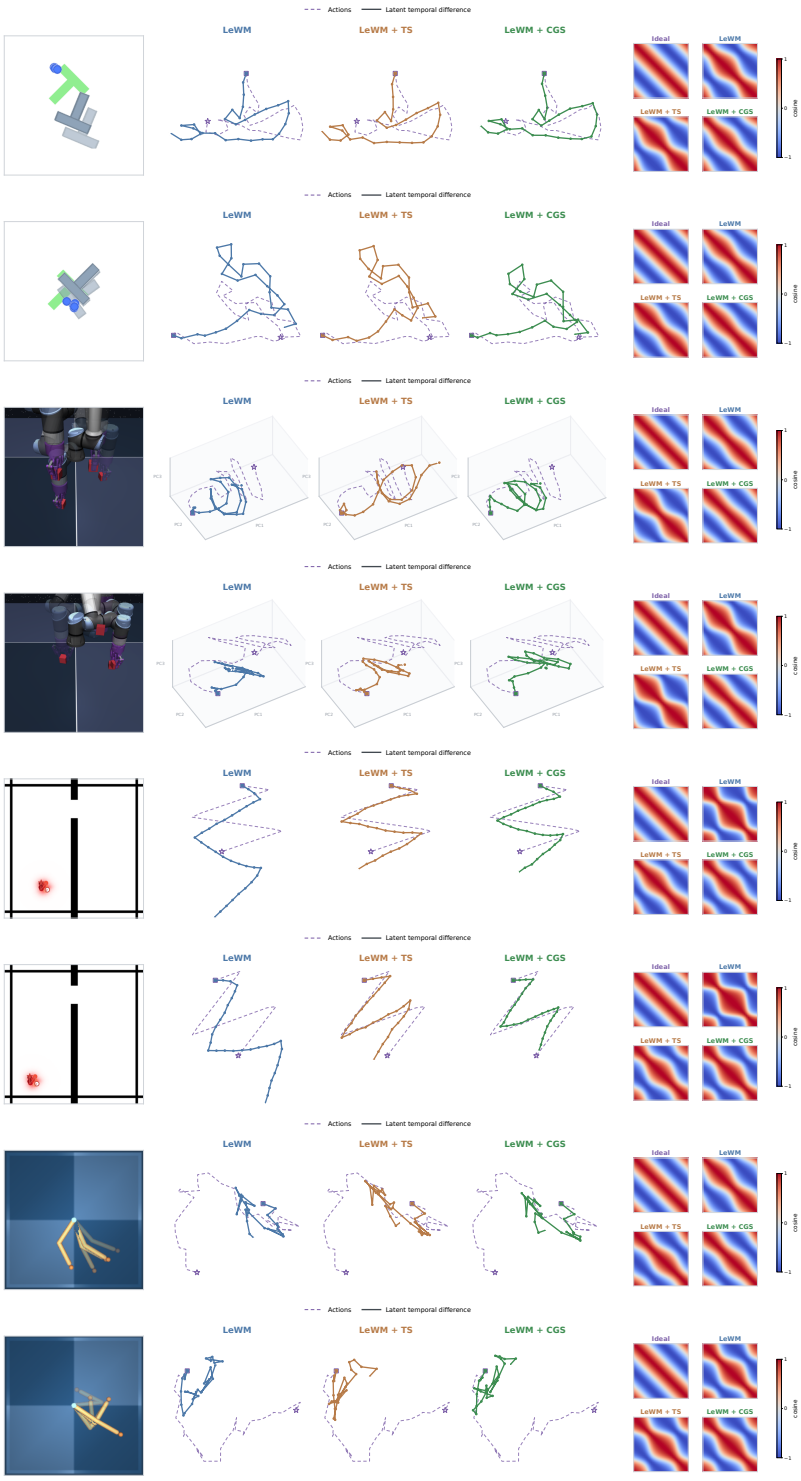


Figure 9: **Action–effect geometry across four environments.** From top to bottom: two examples each of PushT, Cube, TwoRooms, and Reacher. **Left:** environment illustrations. **Middle:** complete 30-frame action references (purple dashed) and encoded transition paths for LeWM (blue), TS (brown), and CGS (green), with shared action PCA in 2D or 3D; squares and stars mark starts and ends. Shape agreement indicates preservation of action-related directions. **Right:** cosine Grams of predicted endpoint responses to circular action perturbations; *Ideal* is the unit-circle reference. Protocol and interpretation are given in Appendix C.2.

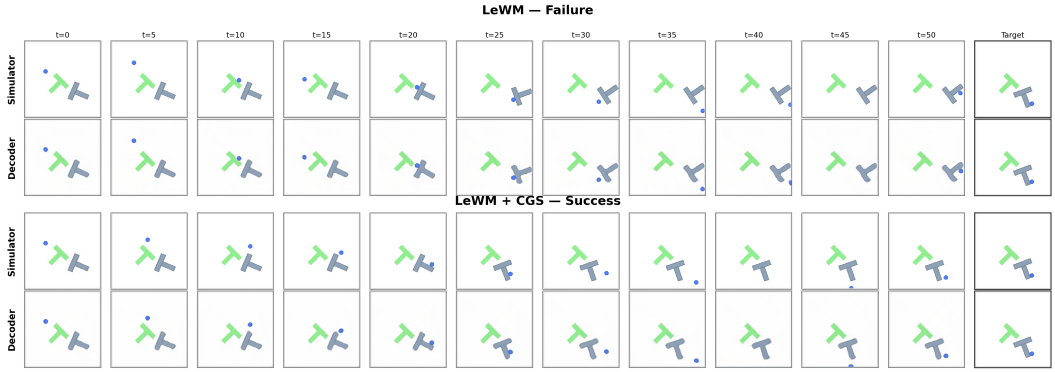


Figure 10: **Selected qualitative PushT planning rollout.** Simulator and decoded trajectories for a LeWM failure (top) and a LeWM+CGS success (bottom). Columns show the rollout over environment steps with the target configuration. The CGS example reaches the target while its decoded trajectory remains qualitatively aligned with the simulator rollout, illustrating that the learned representation retains sufficient task-relevant information for successful planning in this example.

C.2.3 Qualitative PushT Planning Rollout

Figure 10 shows a PushT planning example under LeWM and LeWM+CGS. With LeWM, the planned trajectory fails to reach the target configuration, whereas CGS produces a successful rollout. The corresponding decoded trajectories (decoder trained a posteriori) remain qualitatively consistent with the simulator evolution, indicating that the CGS representation retains sufficient task-relevant information to support successful planning in this example.

C.3 Full Action–Representation Probing

Protocol and probe definitions. We use 200 episodes per environment, split by episode into 140 fit, 20 validation, and 40 test episodes with split seed 20260814. The probe time index follows the frame-skipped sequences used by the world model, with frame skip 5. Thus, consecutive indexed observations o_t and o_{t+1} are separated by five raw data frames, and $a_t \in \mathbb{R}^{d_a}$ denotes the concatenated action vector associated with that interval; d_a is the resulting action dimension after frame-skip concatenation. We define

$$\Delta z_t = z_{t+1} - z_t, \qquad \Delta s_t = s_{t+1} - s_t, \tag{22}$$

with wrapped differences for angular targets.

Here s_t denotes the physical-state features supplied to the probe, rather than necessarily the complete simulator state. State-conditioned probes use agent and object pose on PushT, joint positions and velocities on Cube and Reacher, and agent position on TwoRooms. Encoders are frozen, and z_t is the global encoder token after the learned projector, evaluated on real dataset images. Linear probes are ridge regressions with coefficient 10^{-4}. Nonlinear probes are two-layer width-128 GELU MLPs selected by validation loss; reported MLP values use the ensemble prediction from seeds 7, 17, and 27. Input and target normalizers are fit on the fit episodes only. Unless otherwise stated, scores are pooled held-out $R^2 = 1 - \mathrm{SSE}/\mathrm{SST}$, where $\mathrm{SSE} = \sum_i \|y_i - \hat{y}_i\|_2^2$ is the sum of squared prediction errors and $\mathrm{SST} = \sum_i \|y_i - \bar{y}\|_2^2$ is the total sum of squares around the held-out target mean $\bar{y}$.

Physical identification. We first ask how predictable the physical displacement is from action alone, state alone, or their combination. Let

$$\begin{aligned} \widehat{\Delta s}^{\,g}_{A,t} &= p^g_A(a_t), \\ \widehat{\Delta s}^{\,g}_{S,t} &= p^g_S(s_t), \\ \widehat{\Delta s}^{\,g}_{AS,t} &= p^g_{AS}(a_t, s_t), \end{aligned} \tag{23}$$

where $g \in \{\text{lin}, \text{MLP}\}$ denotes the probe class. We denote their held-out R^2 scores by P_A^g, P_S^g, and P_{AS}^g, respectively. These are environment-level diagnostics and are therefore shared across representations.

The partial action score

$$R^2_{A|S} = 1 - \frac{\text{SSE}\big(P_{AS}^{\text{MLP}}\big)}{\text{SSE}\big(P_S^{\text{MLP}}\big)} \tag{24}$$

measures the additional predictable effect supplied by action after conditioning on the state features. These quantities are empirical held-out predictability diagnostics under the specified probe classes, rather than Bayes-optimal physical ceilings.

Motion decoding. For each frozen representation, we next fit decoders

$$d_{\text{lin}}(\Delta z_t), \qquad d_{\text{MLP}}(\Delta z_t) \tag{25}$$

to predict the true physical displacement Δs_t. Their held-out R^2 scores are denoted D_{lin} and D_{MLP}. Thus D_{MLP} tests whether the evaluated motion information is accessible to the nonlinear probe, whereas D_{lin} measures its linear accessibility from the latent difference.

Control-effect accessibility and agreement. We finally test whether Δz_t makes physically predicted control effects linearly accessible. Two ridge readouts are fit from Δz_t:

$$\begin{aligned} c_A(\Delta z_t) &\approx \widehat{\Delta s}^{\,\text{lin}}_{A,t}, \\ c_{AS}(\Delta z_t) &\approx \widehat{\Delta s}^{\,\text{MLP}}_{AS,t}. \end{aligned} \tag{26}$$

Their held-out R^2 scores are denoted C_A and C_{AS}, respectively. Hence C_A measures accessibility of the linear action-only physical effect, whereas C_{AS} uses the nonlinear state-conditioned physical estimate as its target.

The bridge scores measure agreement in physical displacement coordinates without fitting an additional bridge probe. We reuse the ordinary linear motion decoder d_{lin} and define

$$B_P = 1 - \frac{\sum_t \left\| d_{\text{lin}}(\Delta z_t) - \widehat{\Delta s}_{P,t} \right\|_2^2}{\sum_t \left\| \Delta s_t - \overline{\Delta s} \right\|_2^2}, \qquad P \in \{A, AS\}, \tag{27}$$

where

$$\widehat{\Delta s}_{A,t} := \widehat{\Delta s}^{\,\text{lin}}_{A,t}, \qquad \widehat{\Delta s}_{AS,t} := \widehat{\Delta s}^{\,\text{MLP}}_{AS,t}. \tag{28}$$

We denote the resulting scores by B_A and B_{AS}. Unlike C_A and C_{AS}, these are normalized agreement scores rather than R^2 values of newly fitted probes.

To match the notation from the main context, the six quantities reported there correspond to

$$P_A = P_A^{\text{MLP}}, \qquad P_{AS} = P_{AS}^{\text{MLP}}, \qquad D_{\text{lin}}, \qquad D_{\text{MLP}}, \qquad C_A, \qquad B_{AS}. \tag{29}$$

The additional quantities reported here provide complementary diagnostics. In particular, a high D_{MLP} together with a substantially lower D_{lin} indicates that motion information is retained but is less linearly accessible under these probe classes; this pattern is consistent with a more nonlinear organization of the latent representation.

Physical action-identifiability diagnostics. The physical-prediction results distinguish action dependence from state dependence under the specified probe classes. PushT agent displacement and TwoRooms agent displacement are almost completely predictable from action alone. Reacher fingertip displacement is qualitatively different: both the linear and nonlinear action-only probes obtain $R^2 = -0.006$, whereas the state-conditioned MLP reaches $P_{AS}^{\text{MLP}} = 0.983$. Cube joint, end-effector, and yaw changes are also highly predictable from action, while gripper and block effects show larger linear-to-nonlinear gaps. These physical diagnostics provide the reference needed to interpret the representation probes below.

Table 9: Physical-motion identification. A, S, and AS use the action, current state, and both. $R^2_{A|S}$ uses the MLP probes and the state-only residual as its denominator. Negative values denote performance below the held-out mean predictor.

Environment	Physical change	P_A^{lin}	P_A^{MLP}	P_S^{lin}	P_S^{MLP}	P_{AS}^{lin}	P_{AS}^{MLP}	$R^2_{A\|S}$
PushT	Agent position	0.999	0.999	0.122	0.266	0.999	0.999	0.999
	Block position	0.121	0.317	0.092	0.372	0.175	0.613	0.384
	Block angle	-0.012	0.074	0.045	0.271	0.044	0.543	0.372
Cube	Joint position	0.946	0.966	0.718	0.834	0.960	0.997	0.981
	End-effector position	0.992	0.994	0.753	0.870	0.993	0.999	0.990
	End-effector yaw	0.991	0.992	0.528	0.628	0.992	0.998	0.995
	Gripper opening	0.250	0.961	0.541	0.886	0.853	0.992	0.934
	Block position	0.492	0.991	0.797	0.863	0.876	0.997	0.976
	Block yaw	0.294	0.916	0.523	0.595	0.569	0.913	0.785
Reacher	Joint position	0.847	0.847	0.093	0.110	0.948	0.993	0.992
	Fingertip position	-0.006	-0.006	-0.001	0.251	-0.005	0.983	0.977
TwoRoom	Agent position	0.952	0.953	0.047	0.057	0.952	0.965	0.963

Table 10: Complete PushT representation probes. D denotes dynamic decoding, C control-effect accessibility, and B bridge agreement. Subscripts A and AS denote action-only and state-conditioned physical targets.

Model	Target	D_{lin}	D_{MLP}	C_A	C_{AS}	B_A	B_{AS}
LeWM	Agent position	0.809	0.888	0.807	0.808	0.807	0.808
TS	Agent position	0.831	0.891	0.829	0.830	0.829	0.830
CGS	Agent position	0.903	0.959	0.901	0.902	0.901	0.902
LeWM	Block position	0.813	0.864	0.791	0.602	0.256	0.603
TS	Block position	0.816	0.830	0.817	0.670	0.314	0.678
CGS	Block position	0.847	0.873	0.871	0.655	0.276	0.664
LeWM	Block angle	0.151	0.604	0.773	0.158	0.423	0.308
TS	Block angle	0.357	0.574	0.808	0.287	0.386	0.404
CGS	Block angle	0.227	0.601	0.843	0.143	0.450	0.296

Table 11: Complete Cube representation probes. CGS uses grasp-active transitions and XYZ controls for its geometry objective. All three models share the dataset, episode split, horizon, latent dimension, and probe protocol.

Model	Target	D_{lin}	D_{MLP}	C_A	C_{AS}	B_A	B_{AS}
LeWM	Joint position	0.742	0.747	0.741	0.743	0.723	0.743
TS	Joint position	0.622	0.643	0.629	0.622	0.631	0.621
CGS	Joint position	0.743	0.895	0.748	0.741	0.729	0.741
LeWM	End-effector position	0.895	0.969	0.893	0.893	0.891	0.893
TS	End-effector position	0.710	0.776	0.706	0.708	0.700	0.709
CGS	End-effector position	0.931	0.982	0.930	0.930	0.927	0.930
LeWM	End-effector yaw	-0.039	-0.023	-0.036	-0.040	-0.026	-0.042
TS	End-effector yaw	-0.025	-0.020	-0.024	-0.026	-0.010	-0.026
CGS	End-effector yaw	-0.078	0.830	-0.080	-0.078	-0.063	-0.078
LeWM	Gripper opening	0.490	0.905	0.205	0.494	0.502	0.498
TS	Gripper opening	0.528	0.905	0.296	0.533	0.543	0.537
CGS	Gripper opening	0.898	0.945	0.381	0.900	0.307	0.901
LeWM	Block position	0.941	0.970	0.926	0.939	0.517	0.939
TS	Block position	0.946	0.976	0.813	0.943	0.513	0.944
CGS	Block position	0.918	0.985	0.948	0.916	0.526	0.916
LeWM	Block yaw	-0.013	-0.009	-0.054	-0.041	0.757	0.101
TS	Block yaw	-0.018	-0.020	-0.029	-0.050	0.750	0.081
CGS	Block yaw	-0.009	0.426	-0.102	-0.032	0.751	0.096

Table 12: Complete Reacher and TwoRoom representation probes, using the same notation and protocol as Tables 10 and 11.

Environment	Model	Target	D_{lin}	D_{MLP}	C_A	C_{AS}	B_A	B_{AS}
Reacher	LeWM	Joint position	0.355	0.988	0.368	0.357	0.447	0.364
	TS	Joint position	0.347	0.988	0.371	0.348	0.447	0.353
	CGS	Joint position	0.123	0.988	0.135	0.127	0.248	0.134
	LeWM	Fingertip position	0.951	0.994	0.155	0.932	0.028	0.932
	TS	Fingertip position	0.974	0.994	0.157	0.960	0.011	0.959
	CGS	Fingertip position	0.899	0.993	0.041	0.883	0.074	0.884
TwoRoom	LeWM	Agent position	0.862	0.956	0.815	0.829	0.819	0.827
	TS	Agent position	0.970	0.984	0.933	0.946	0.928	0.939
	CGS	Agent position	0.903	0.976	0.866	0.878	0.867	0.873

C.3.1 Representation probes among environments.

PushT: a targeted positive result. For agent displacement, action-only and state-conditioned physical prediction both reach $R^2 = 0.999$. Relative to LeWM, CGS raises D_{lin} from 0.809 to 0.903, D_{MLP} from 0.888 to 0.959, and C_A from 0.807 to 0.901, while also giving the strongest $B_{AS} = 0.902$; TS is intermediate on these agent-motion metrics. CGS is therefore strongest for the directly action-predictable agent effect and also improves linear motion decoding and action-effect accessibility for block translation. The state-conditioned object results are less uniform: TS gives the strongest block-position bridge score (0.678 versus 0.664 for CGS) and block-angle bridge score (0.404 versus 0.296). These results should be read together with the lower action-only physical-prediction scores for the object targets: accessibility of a predictable component does not imply that action alone determines the full object displacement.

Cube. All scores below are evaluated on the full held-out test split, although the CGS geometry objective was applied only to grasp-active transitions during representation training. CGS gives the strongest nonlinear motion decoding on all six evaluated targets. For end-effector position, D_{lin} rises from 0.895 with LeWM to 0.931 with CGS; for gripper opening, it rises from 0.490 to 0.898, while B_{AS} rises from 0.498 to 0.901. TS retains the strongest block-position agreement ($B_{AS} = 0.944$). Both yaw targets remain difficult to decode linearly, despite substantially stronger nonlinear readout for CGS.

Reacher: motion is retained but strongly state dependent. For fingertip displacement, both action-only physical probes remain at $P_A = -0.006$, whereas conditioning on joint positions and velocities raises the nonlinear score to $P_{AS} = 0.983$. Thus the evaluated fingertip displacement is strongly configuration dependent under these probe classes. Increasing from the linear action-only predictor to our nonlinear action-only MLP does not repair this mismatch, while the state-conditioned predictor does.

At the representation level, direct action-effect accessibility remains low for fingertip motion ($C_A = 0.041$–0.157), and CGS obtains $B_{AS} = 0.884$, below LeWM (0.932) and TS (0.959). At the same time, all three representations retain nearly all evaluated motion information under nonlinear decoding: $D_{\text{MLP}} \approx 0.99$ for both joint and fingertip targets. The main difference is therefore how linearly accessible that information is from Δz_t, rather than whether it is present at all. TS gives the strongest state-conditioned accessibility and agreement on the fingertip target, while CGS retains the motion information with weaker linear accessibility.

TwoRooms: TS is strongest despite highly identifiable action effects. TwoRooms has high action-only and state-conditioned physical-prediction scores, yet TS is strongest across the reported representation probes, including $D_{\text{lin}} = 0.970$, $C_{AS} = 0.946$, and $B_{AS} = 0.939$. CGS improves over LeWM but remains below TS on these diagnostics. High action identifiability alone is therefore not sufficient for CGS to outperform temporal straightening.

C.4 Mechanism-Figure Protocol and Additional Results

The illustration in Figure 3 links measured Cube planning geometry to one-step MPPI progress. We use 32 shared grasp-active anchors from distinct episodes and perturb the XYZ commands uniformly across the 25-action horizon. The orthonormal basis $U = \mathbf{1}_{25} \otimes [I_3, 0_{3\times 2}]^\top / \sqrt{25}$ maps three-dimensional perturbations into the full action sequence. At each recorded action sequence, we

differentiate the terminal predictor and form

$$G_{H,\mathrm{XYZ}} := (J_{\widehat{F}_H} U)^\top (J_{\widehat{F}_H} U),$$
$$\bar{G}_{H,\mathrm{XYZ}} := \frac{G_{H,\mathrm{XYZ}}}{\mathrm{tr}(G_{H,\mathrm{XYZ}})/3}. \tag{30}$$

The response Gram matrix $G_{H,\mathrm{XYZ}}$ measures local sensitivity to XYZ actions. Trace normalization fixes the average eigenvalue at one, allowing spectral-shape comparisons at a common scale. Below, write $\bar{G} := \bar{G}_{H,\mathrm{XYZ}}$. Since $\bar{G}$ is positive semidefinite, $\sigma_{\min}(\bar{G}) = \lambda_{\min}(\bar{G})$ measures its weakest normalized XYZ curvature.

Let V contain the eigenvectors of $\bar{G}$ in descending eigenvalue order. We place the target along its weakest direction, $u^\star = (0, 0, 1)^\top$, and evaluate the quadratic cost

$$\widehat{C}(u) = (u - u^\star)^\top V^\top \bar{G} V (u - u^\star), \qquad q_K = \frac{\|\widehat{\mu}_K - u^\star\|_2}{\|u^\star\|_2}. \tag{31}$$

MPPI starts from $\mathcal{N}(0, I_3)$ with temperature $\tau = 2$; $\widehat{\mu}_K$ is its weighted sample mean. We call q_K the K-sample contraction factor: the ratio of post-update to initial mean-error norms, with initial mean zero. It is a random error ratio, not an empirical Jacobian norm, and can exceed one. Each anchor uses 512 Gaussian batches, with shared sample coordinates across models and prefix budgets $K \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. Thus $\Pr(q_K < 1)$ is the probability that one update reduces target error in this quadratic problem. Curves average over anchors, with 95% bootstrap intervals.

The quadratic Hessian is $2V^\top \bar{G} V$, so unit proposal covariance and $\tau = 2$ give the population error map $(I_3 + V^\top \bar{G} V)^{-1}$. Because the initial error lies along the weakest-curvature direction, the population counterpart of q_K is

$$q_\infty = \frac{1}{1 + \sigma_{\min}(\bar{G}_{H,\mathrm{XYZ}})}. \tag{32}$$

For full-rank $\bar{G}$, this equals the contraction factor q_{M} in Theorem 4.6 applied to this calibrated three-dimensional quadratic probe, rather than the full nonlinear planning problem. Thus, larger curvature along the weakest direction produces greater population progress. Median q_∞ is 0.841, 0.835, and 0.685 for LeWM, TS, and CGS. At $K = 32$, their one-update progress probabilities are 75.1%, 80.3%, and 91.3%. This illustration shows how more balanced action sensitivity can improve MPPI updates at a fixed sample budget.

C.5 DINO-WM Implementation and Additional Results

This section provides the implementation details for the DINO-WM comparisons in Figure 5 and reports additional results on the DINO-WM benchmark settings used by Temporal Straightening (Wang et al., 2026b). Following that setup, we evaluate Base (w/o TS or CGS), TS, and CGS on PushT, Wall, PointMaze–UMaze, and PointMaze–Medium, using both global CLS and spatial Patch representations. Within each representation type, all three objectives share the same frozen DINOv2 backbone, data, predictor architecture, and planning protocol.

Representations and geometry objectives. Temporal Straightening was evaluated with frozen DINOv2 features and trainable projectors, as well as with ResNet encoders learned from scratch (Wang et al., 2026b). Our DINO-based experiments follow the spatial DINO-WM setting of DINO-WM and TS (Zhou et al., 2025; Wang et al., 2026b): a frozen DINOv2-S/14 backbone (Oquab et al., 2024) produces a 14×14 grid of 384-dimensional patch tokens, which are projected to eight dimensions per token. The predictor rolls out this projected spatial grid. TS and CGS regularize a learned 128-dimensional aggregation of patch-token differences, while prediction and planning retain the full spatial representation.

Our CLS variant instead uses the backbone's 384-dimensional CLS token, followed by an identity-initialized trainable adapter, and applies the geometry objective directly to the resulting global latent differences. The global DINO variant in the original TS study obtains a single vector by projecting patch features (Wang et al., 2026b); our CLS variant therefore differs in the backbone output to which the geometry objective is applied. In comparison, LeWM learns its visual encoder end to end and predicts a compact 192-dimensional projected CLS latent (Maes et al., 2026).

Training protocol. Prediction targets use stop-gradient in the DINO-WM variants. The projector or CLS adapter, predictor, action encoder, and geometry head are trainable, while the DINOv2 backbone remains frozen. On PushT, we use a 90/10 trajectory split, two training epochs, and unit gradient clipping. Representation learning rates are 10^{-6} for Base and 10^{-5} for TS and CGS; the predictor and action encoder use a learning rate of 5×10^{-4} with weight decay 10^{-2}. Models condition on three observations and predict five-step transitions. The PushT CGS weights are 0.1 for Patch and CLS.

Planning protocol. DINO-WM sampling costs average squared visual prediction errors over tokens and channels. On PushT, we additionally predict proprioception and add its MSE to the visual cost with unit weight; proprioception is not included in either TS or CGS regularization. The GD comparison uses the optimization schedule in Appendix C.1.1, with per-step error given by visual MSE plus proprioceptive MSE.

Sampling-based planners use $I = 30$ optimization steps, and CEM retains an elite fraction of 0.25. GD uses 100 Adam steps. Within each environment, a single MPPI temperature is shared across CLS/Patch and Base/TS/CGS: 0.015 on PushT, 5×10^{-6} on Wall, 10^{-3} on PointMaze–UMaze, and 3×10^{-3} on PointMaze–Medium. These temperatures are fixed across sampling budgets and representation objectives. Evaluation uses three evaluation seeds and 50 episodes per seed.

Table 13: **Additional DINO-WM benchmark results.** Goal-reaching success rate (%) for Base, TS, and CGS using frozen DINOv2 CLS or Patch representations on the DINO-WM benchmark settings used by Temporal Straightening. CEM and MPPI use $I = 30$ optimization steps, and GD uses 100 Adam steps. A single environment-level MPPI temperature is shared across representation types and training objectives. Values are means ± sample SD over three evaluation seeds with 50 evaluation episodes per seed; PushT additionally uses proprioception. Bold marks the best objective for a fixed representation type, planner, environment, and sampling budget.

World model / objective	Planning method	Wall			PointMaze–UMaze			PointMaze–Medium			PushT		
		Sampling budget			Sampling budget			Sampling budget			Sampling budget		
		K=32	K=64	K=128	K=32	K=64	K=128	K=32	K=64	K=128	K=32	K=64	K=128
DINO-WM (CLS)	CEM	74.7±3.1	76.0±5.3	74.7±6.1	88.0±3.5	92.7±3.1	90.7±5.0	85.3±4.2	86.0±2.0	84.7±7.0	47.3±6.4	56.0±8.7	63.3±5.0
	MPPI	84.0±5.3	**86.7±6.4**	**89.3±5.0**	92.0±2.0	93.3±2.3	96.7±4.2	90.7±5.0	90.7±2.3	92.7±3.1	31.3±6.1	35.3±4.6	46.7±3.1
	GD		55.3±2.3			62.0±5.3			56.0±4.0			46.0±2.0	
+TS (CLS)	CEM	79.3±6.1	78.7±4.2	77.3±8.3	86.0±2.0	89.3±5.0	90.0±2.0	87.3±1.2	84.7±3.1	89.3±4.6	41.3±5.0	38.0±2.0	47.3±4.2
	MPPI	**88.7±5.8**	83.3±7.6	84.0±4.0	91.3±4.2	94.7±1.2	95.3±3.1	88.7±4.2	88.7±6.1	94.0±2.0	29.3±4.2	29.3±10.3	29.3±5.8
	GD		68.0±4.0			66.7±3.1			65.3±8.1			34.0±14.0	
+CGS (CLS)	CEM	**83.3±8.1**	**81.3±7.0**	**83.3±3.1**	**93.3±3.1**	**96.0±4.0**	**96.0±3.5**	**92.0±2.0**	**92.7±3.1**	**93.3±2.3**	**55.3±4.6**	**65.3±6.1**	**66.7±6.4**
	MPPI	86.0±6.9	84.7±5.8	81.3±6.1	**96.7±3.1**	**96.0±2.0**	**97.3±3.1**	**94.7±1.2**	**95.3±2.3**	**95.3±3.1**	**40.7±4.2**	**46.7±2.3**	**56.7±4.6**
	GD		**81.3±7.6**			**81.3±4.2**			**79.3±6.1**			**57.3±8.1**	
DINO-WM (Patch)	CEM	90.7±2.3	81.3±8.1	84.7±6.4	**98.0±2.0**	**98.7±1.2**	**99.3±1.2**	**94.7±1.2**	**92.7±2.3**	**98.7±1.2**	75.3±2.3	81.3±4.2	83.3±1.2
	MPPI	**90.0±2.0**	89.3±2.3	86.0±5.3	**99.3±1.2**	96.7±1.2	98.7±1.2	94.0±4.0	**97.3±3.1**	**97.3±2.3**	68.0±4.0	62.0±4.0	75.3±5.0
	GD		83.3±3.1			94.0±4.0			**88.7±4.2**			78.0±5.3	
+TS (Patch)	CEM	**91.3±1.2**	87.3±2.3	**86.0±2.0**	94.7±3.1	97.3±3.1	95.3±1.2	88.7±3.1	90.7±1.2	92.0±3.5	76.7±5.0	**84.7±4.6**	**87.3±4.2**
	MPPI	89.3±1.2	**91.3±2.3**	90.7±3.1	97.3±3.1	96.7±1.2	98.0±2.0	94.0±2.0	93.3±3.1	95.3±2.3	**75.3±4.2**	**80.0±8.7**	78.7±4.6
	GD		86.7±2.3			**96.0±2.0**			87.3±4.6			**84.0±5.3**	
+CGS (Patch)	CEM	88.0±4.0	**88.7±5.0**	85.3±6.4	96.0±2.0	96.0±2.0	98.0±0.0	94.0±2.0	86.0±2.0	92.0±3.5	**80.0±3.5**	81.3±2.3	84.7±5.0
	MPPI	88.0±7.2	88.7±1.2	**92.7±3.1**	98.7±1.2	**99.3±1.2**	**100.0±0.0**	**94.7±1.2**	95.3±1.2	96.0±2.0	64.7±4.2	72.7±6.4	**80.7±2.3**
	GD		**88.7±4.2**			**96.0±3.5**			88.0±2.0			76.7±1.2	

Additional DINO-WM benchmark results. Table 13 shows that the effect of geometry shaping depends on both the underlying representation and the planner. CGS gives particularly consistent improvements with the compact CLS representation. At $K = 128$, relative to Base CLS, CGS improves CEM on all four environments, improves GD on all four, and improves MPPI on PointMaze–UMaze, PointMaze–Medium, and PushT. Wall MPPI is the exception, where the Base representation performs better. The Patch representation begins from substantially stronger planning performance and leaves less headroom for geometry regularization. Its results are correspondingly more mixed: CGS improves several planner–task combinations, including PushT CEM and MPPI, Wall MPPI, and PointMaze–UMaze MPPI, while Base or TS remains stronger in other settings. These additional experiments therefore show that CGS can be applied to both compact global and spatial DINO-WM representations, while its benefit is not uniform across every representation–planner combination.

Cross-architecture planning time. We measure DINO-WM (Patch)+CGS and LeWM+CGS on the same GPU, using episode batch size 50 for GD. Times sum the two planner solves and are averaged per episode; model loading, environment execution, and rendering are excluded. Sampling uses $K = 128$ and $I = 30$, while GD uses 100 optimization steps. The DINO-WM PushT planner uses proprioception and MPPI temperature 0.015; LeWM uses image latents and temperature 4. Figure 5 combines these timing measurements with the success rates from the main evaluations.

Table 14: **PushT planner-solve time at $K = 128$.** Seconds per episode measured in a timing experiment. Times include both planner solves and exclude loading, environment execution, and rendering. Ratios characterize the stated implementations and batching settings.

Model	CEM	MPPI	GD
DINO-WM (Patch)+CGS	86.852	86.959	6.516
LeWM+CGS	5.163	1.850	0.161
DINO/LeWM	16.8×	47.0×	40.5×

C.6 Training Ablations

We vary the ramp fraction, history length, SIGReg weight and CGS weight on PushT, evaluating each checkpoint with MPPI, CEM, and GD (Figure 11). Sampling uses $K = 128$, $I = 30$, MPPI temperature 4, and 32 CEM elites; GD uses 100 steps. The reference uses $\lambda_{\text{CGS}} = 0.01$, a 5% linear ramp, $N = 3$, and $\lambda_{\text{SIGReg}} = 0.09$.

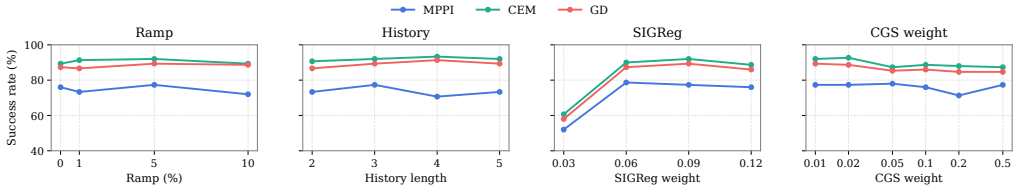


Figure 11: **PushT training ablations.** Ramp fraction, history length, SIGReg weight, and CGS weight, evaluated with MPPI, CEM, and GD. Means use three evaluation seeds with 50 episodes each; GD is independent of K. Reference points reuse the main PushT checkpoint with $\lambda_{\text{CGS}} = 0.01$, a 5% ramp, $N = 3$, and $\lambda_{\text{SIGReg}} = 0.09$.

The 5% ramp gives the strongest overall performance across the three planners, with modest differences across ramp choices. Following LeWM, $N = 3$ provides an economical history setting: CEM and GD gain slightly at $N = 4$, while MPPI is strongest at $N = 3$. For SIGReg, weight 0.03 substantially reduces success, indicating potential representation collapse with small SIGReg strength, while 0.06–0.12 forms a comparatively stable range around the default 0.09. The CGS-weight sweep is broadly stable for all three planners over the tested range, and goes down when CGS-weight becomes larger, suggesting conflicts with representation geometry induced by other loss terms, so we choose a conservative weight $\lambda_{\text{CGS}} = 0.01$ on PushT.

C.6.1 Can Explicit Temporal Straightening Further Benefit LeWM?

Maes et al. (2026, Appendix H) report that LeWM develops temporal straightening on PushT without an explicit TS loss. Our diagnostic shows that adding TS further raises the final cosine of consecutive latent displacements from 0.548 to 0.613 on PushT slightly, with a small change in planning success rate in Table 3. Cube and TwoRooms show clearer geometric changes ($0.262 \rightarrow 0.666$ and $-0.341 \rightarrow -0.168$) and planning gains (MPPI: $59.3\% \rightarrow 75.3\%$ and $92.0\% \rightarrow 97.3\%$). Reacher remains near zero cosine ($-0.049 \rightarrow -0.003$), indicating that explicit TS produces only a small additional straightening effect in this environment. These results support explicit TS as a useful addition to LeWM, with the clearest gains on Cube and TwoRooms. Figure 12 tracks the cosine similarity of consecutive latent displacements during training on fixed validation windows for the LeWM and LeWM+TS variants in Section 5.

C.7 Shared MPPI Temperature Selection

Figure 13 sweeps the MPPI temperature from 0.25 to 512 at $K = 128$ and $I = 30$ on all four environments, and summarizes planning stability across wide temperature range in the bottom table. Temperature controls the concentration of the MPPI importance weights. At very small values, the weights collapse onto one or a few candidates, producing an effectively single-mode update that is high-variance and brittle. At very large values, the weights approach uniformity, so low-quality candidates contribute substantially and dilute the improvement signal. For fair selection, we examine

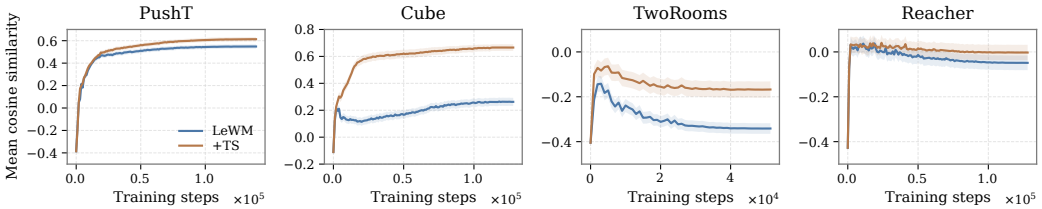


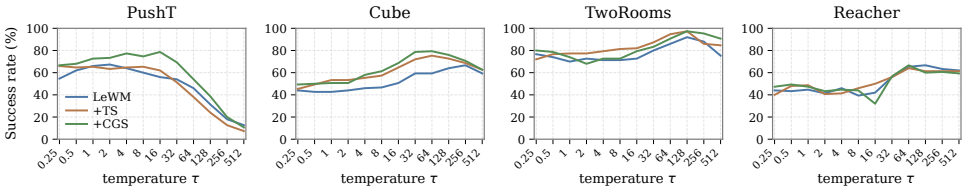
Figure 12: **Temporal straightening in LeWM.** Mean cosine similarity of consecutive latent displacements on fixed validation windows. Lines follow one training run per method and environment; bands show ± 1.96 standard errors across probe episodes. TS produces higher final mean cosine similarity on all four environments.

the LeWM, LeWM+TS, and LeWM+CGS curves jointly within each environment and choose one shared temperature from a broad stable region where all three methods achieve competitive success: 4 on PushT, 64 on Cube, 128 on TwoRooms, and 64 on Reacher. Each selected value is fixed across representations and all sampling budgets in the main results and Tables 3–6. nAUC results summarized in the bottom table also shows the stable improvement of CGS on pushT and Cube, comparable stable improvement of CGS with TS on TwoRooms, also similar planning performance across temperature of all three methods on Reacher.

C.8 Refinement Efficiency and Matched BC Priors on Other Environments

Refinement efficiency. We evaluate refinement speed at fixed candidate budget $K = 128$, using 32 CEM elites and the environment-specific MPPI temperatures from Appendix C.1.1. We vary the number of optimization steps over $I \in \{1, 2, 5, 10, 15, 20, 25, 30\}$, with each replanning solve initialized independently. Each run evaluates the final proposal, so success need not increase monotonically with I. We also summarize the complete no-prior CEM and MPPI refinement curves using nAUC in Table 15.

On Cube, CGS improves both CEM and MPPI within the first few optimization steps, consistent with its stronger finite-budget planning performance. On TwoRooms, TS and CGS both reach high success rapidly, with little room for further separation as the task approaches saturation. On Reacher, refinement differences are smaller than on Cube and TwoRooms, but CGS attains the highest aggregate refinement nAUC for both CEM and MPPI. The gains are modest and not monotonic across iterations, consistent with a more planner- and budget-dependent effect.



Model	PushT	Cube	TwoRooms	Reacher
LeWM	50.76	52.15	77.64	51.00
LeWM + TS	49.82 (−0.94)	61.45 (+9.30)	**83.42** (+5.79)	**51.64** (+0.64)
LeWM + CGS	**60.48** (+9.73)	**63.64** (+11.48)	81.58 (+3.94)	50.73 (−0.27)

Figure 13: **MPPI temperature sensitivity and temperature-selection nAUC at $K = 128$. Top:** Mean success over three evaluation seeds (50 episodes each, $I = 30$), using one shared temperature per environment and a 0–100% axis. **Bottom:** nAUC over the displayed τ values on a $\log_2$ scale; parentheses give percentage-point differences from LeWM and bold marks the best representation.

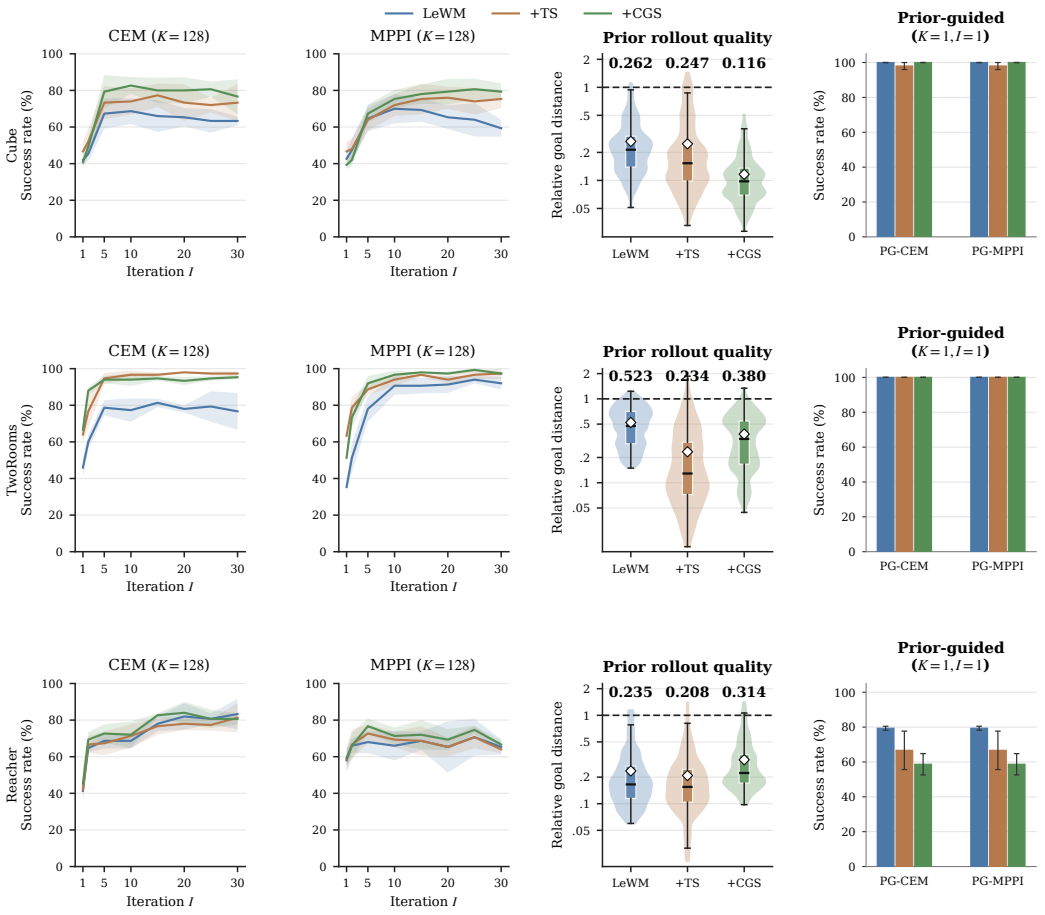


Figure 14: **Refinement efficiency and matched BC prior quality on Cube, TwoRooms, and Reacher.** Left: CEM and MPPI success over I (iterations) at $K = 128$; bands show sample SD over three evaluation seeds. Middle: first-solve prior endpoint distance divided by zero-action endpoint distance, on the same 150 starts; diamonds and labels show means. Right: prior-only success at $K = 1$, $I = 1$, with the prior mean as the sole candidate; error bars show sample SD.

Table 15: **Refinement-iteration nAUC at $K = 128$.** $I \in \{1, 2, 5, 10, 15, 20, 25, 30\}$. nAUC integrates over linear I. Parentheses give percentage-point differences from the matching LeWM baseline. Bold marks the best representation for each planner and environment. All entries summarize three evaluation seeds.

Planner	Model	PushT	Cube	TwoRooms	Reacher
CEM	LeWM	73.93	64.00	76.87	75.16
	LeWM + TS	72.23 (−1.70)	71.98 (+7.98)	**94.85** (+17.98)	73.92 (−1.24)
	LeWM + CGS	**80.25** (+6.32)	**77.40** (+13.40)	93.34 (+16.47)	**77.60** (+2.44)
MPPI	LeWM	64.85	64.36	86.06	67.23
	LeWM + TS	67.06 (+2.21)	70.70 (+6.34)	92.89 (+6.83)	68.39 (+1.16)
	LeWM + CGS	**75.71** (+10.86)	**73.72** (+9.37)	**94.49** (+8.44)	**71.44** (+4.21)

Matched BC-prior quality. We separately evaluate the quality of the matched BC prior before sampling-based refinement. At the first solve on 150 shared starts, we measure

$$\rho = \frac{\|\widehat{F}_H(\mu_p) - z_g\|_2}{\|\widehat{F}_H(0) - z_g\|_2}, \tag{33}$$

where μ_p is the prior-predicted action sequence. Lower values indicate greater predicted goal progress relative to that representation's own zero-action baseline. Prior-only runs execute μ_p as the sole candidate at $K = 1$, $I = 1$, so PG-CEM and PG-MPPI are identical in this setting. Each prior is trained separately on its frozen representation, as described in Appendix C.1.1.

On Cube, CGS yields the strongest prior rollout quality, reducing the mean normalized goal-distance ratio from 0.262 for LeWM to 0.116, while prior-only success is already near saturation. On TwoRooms, all three matched priors reach 100% prior-only success, so initialization quality has little room to affect the final result. On Reacher, the TS- and CGS-matched priors are weaker than the LeWM prior at initialization. However, across the sample-budget sweep, these differences narrow as sampling-based refinement incorporates more model-evaluated candidates. Thus, weaker prior initialization can be partially corrected by subsequent refinement, separating prior quality from the representation's performance after model-based search.

D Proofs and Extended Theory

Sections D.1–D.2 derive transition geometry and the planning Hessian from CGS, including the square and rectangular action cases. Section D.3 fixes the quadratic planning problem and Gaussian proposal coordinates. Sections D.4–D.6 then analyze MPPI, CEM, and gradient descent, respectively. Section D.7 discusses other training objectives. The protocol for Figure 3 is reported with the additional experimental results in Appendix C.4.

Throughout this appendix, we use the main context convention $C(\mathbf{a}) = \frac{1}{2}\|z_H - z_g\|_2^2$, so $G_H = \nabla_{\mathbf{a}}^2 C = \Gamma_H^\top \Gamma_H$ for linear rollouts. The corresponding MPPI scale is $s = \sigma_{\mathrm{p}}^2 cH/\tau$. Multiplying the cost and temperature by the same positive constant preserves MPPI weights; positive cost rescaling leaves CEM elite selection unchanged.

D.1 From CGS Loss to Transition Geometry

The population statements below use two independent copies of a transition; they do not assume that successive transitions within a training window are independent. Applying the identities to temporally dependent pairs requires a separate dependence argument. We also use the same local matrices A, B for encoded transitions and predictor rollouts. Transferring an encoder-side CGS bound to a separately parameterized predictor requires agreement of their local transition maps; it does not follow from the CGS loss alone.

Let (z, a) and (z', a') be independent draws from the same transition distribution, with $x = [z^\top, a^\top]^\top$, $x' = [z'^\top, a'^\top]^\top$, and the corresponding differences $\Delta z = Dz + Ba$ and $\Delta z' = Dz' + Ba'$. The population version of the practical cosine CGS objective is

$$\mathcal{L}_{\mathrm{CGS}}^{\mathrm{pop}} = \mathbb{E}\left[\left(\frac{\Delta z^\top \Delta z'}{\|\Delta z\|_2 \|\Delta z'\|_2} - \frac{a^\top a'}{\|a\|_2 \|a'\|_2}\right)^2\right]. \tag{34}$$

Assume the fixed-radius idealization

$$\|\Delta z_t\|_2 = r_\Delta, \qquad \|a_t\|_2 = r_a, \qquad r_\Delta, r_a > 0. \tag{35}$$

With $c = r_\Delta^2 / r_a^2$, recall the Gram loss from the main context

$$\mathcal{L}_G(c) = \mathbb{E}\left[(\Delta z^\top \Delta z' - c\, a^\top a')^2\right]. \tag{36}$$

Then

$$\mathcal{L}_{\mathrm{CGS}}^{\mathrm{pop}} = r_\Delta^{-4} \mathcal{L}_G(c). \tag{37}$$

Thus the population cosine objective is exactly a scaled Gram-matching objective under Eq. (35). Away from this idealization, an approximate relation can be justified under quantitatively small relative fluctuations of the transition and action norms around r_Δ and r_a, respectively; mere boundedness of the norms does not ensure a small approximation error. The analysis below uses the fixed-radius model.

D.1.1 Proof of Proposition 4.2

We give the calculation for general d, d_a under Assumption 4.1. The feature coverage condition gives $\rho I_{d+d_a} \preceq \Sigma_x$. For any $v \in \mathbb{R}^{d+d_a}$, bounded features and Cauchy–Schwarz give

$$v^\top \Sigma_x v = \mathbb{E}[(v^\top x_t)^2] \le \|v\|_2^2 \mathbb{E}\|x_t\|_2^2 \le \bar{\rho}\|v\|_2^2.$$

Thus $\rho I_{d+d_a} \preceq \Sigma_x \preceq \bar{\rho} I_{d+d_a}$. The analysis in the main context specializes to $d = d_a$ with B invertible; the Gram identity and spectral equivalence below do not require a square B.

Let

$$M_\Delta := [D \quad B] \in \mathbb{R}^{d \times (d+d_a)}, \qquad P_c := \begin{bmatrix} 0_{d\times d} & 0_{d\times d_a} \\ 0_{d_a \times d} & cI_{d_a} \end{bmatrix}. \tag{38}$$

Because $\Delta z_t = M_\Delta x_t$,

$$\Delta z^\top \Delta z' - c\, a^\top a' = x^\top Q x', \qquad Q := M_\Delta^\top M_\Delta - P_c. \tag{39}$$

For independent samples x, x',

$$\mathcal{L}_G(c) = \mathbb{E}[(x^\top Q x')^2] = \operatorname{tr}(Q\Sigma_x Q\Sigma_x) = \left\|\Sigma_x^{1/2} Q \Sigma_x^{1/2}\right\|_F^2. \tag{40}$$

Moreover,

$$Q = \begin{bmatrix} D^\top D & D^\top B \\ B^\top D & B^\top B - cI_{d_a} \end{bmatrix}. \tag{41}$$

Since the singular values of the linear map $Q \mapsto \Sigma_x^{1/2} Q \Sigma_x^{1/2}$ lie between ρ and $\bar{\rho}$,

$$\rho^2 \|Q\|_F^2 \le \left\|\Sigma_x^{1/2} Q \Sigma_x^{1/2}\right\|_F^2 \le \bar{\rho}^2 \|Q\|_F^2. \tag{42}$$

Finally,

$$\|Q\|_F^2 = \|D^\top D\|_F^2 + 2\|D^\top B\|_F^2 + \|B^\top B - cI_{d_a}\|_F^2 = \mathcal{R}(A, B). \tag{43}$$

Together with Eq. (40), this proves the spectral equivalence in Proposition 4.2. Under positive definite joint coverage, $\mathcal{L}_G(c) = 0$ if and only if $Q = 0$. The upper-left block gives $D^\top D = 0$, hence $D = 0$ and $A = I_d$; the lower-right block gives $B^\top B = cI_{d_a}$. Conversely, these two conditions make every block of Q zero. When $d \ge d_a$, such a full-column-rank B exists, so zero loss is attainable.

D.1.2 Proof of Corollary 4.3

Proposition 4.2 gives $\mathcal{R}(A, B) \le \ell/\rho^2$. It first yields the Frobenius-norm defect bounds

$$\|D^\top D\|_F \le \frac{\sqrt{\ell}}{\rho}, \quad \|D^\top B\|_F \le \frac{\sqrt{\ell}}{\sqrt{2}\,\rho}, \quad \|B^\top B - cI_{d_a}\|_F \le \frac{\sqrt{\ell}}{\rho}. \tag{44}$$

Using $\|D\|_{\mathrm{op}}^2 = \|D^\top D\|_{\mathrm{op}} \le \|D^\top D\|_F$ and $\|\cdot\|_{\mathrm{op}} \le \|\cdot\|_F$ gives the spectral-norm bounds

$$\begin{aligned} \|D\|_{\mathrm{op}}^2 &\le \frac{\sqrt{\ell}}{\rho}, \\ \|D^\top B\|_{\mathrm{op}} &\le \frac{\sqrt{\ell}}{\sqrt{2}\,\rho}, \\ \|B^\top B - cI_{d_a}\|_{\mathrm{op}} &\le \frac{\sqrt{\ell}}{\rho}. \end{aligned} \tag{45}$$

Setting

$$\varepsilon := \frac{\sqrt{\ell}}{\rho} \max\{1, (2c)^{-1/2}, c^{-1}\}. \tag{46}$$

gives the three scaled conditions in Corollary 4.3. Finally, when $\varepsilon < 1$,

$$c(1-\varepsilon) I_{d_a} \preceq B^\top B \preceq c(1+\varepsilon) I_{d_a}, \tag{47}$$

so B has full column rank and

$$\kappa(B^\top B) \le \frac{1+\varepsilon}{1-\varepsilon}. \tag{48}$$

The mixed defect is stronger than a bound obtained only from $\|D\|_{\mathrm{op}}^2 \le \varepsilon$. It directly controls $\|D^\top B\|_{\mathrm{op}} = O(\sqrt{c}\,\varepsilon)$, suppressing the first-order term $B^\top D B$ in G_H. This is what produces the $O_H(\varepsilon)$ leading term in Theorem 4.4.

D.1.3 Coverage and dimension assumptions

Joint coverage and state–action correlation. Assumption 4.1 adapts the bounded-feature and feature coverage conditions of Yin et al. (2022, Definition 2.1 and Assumption 2.2) to the concatenated state–action feature $x_t = [z_t^\top, a_t^\top]^\top$. Our boundedness condition is $\|x_t\|_2^2 \le \bar{\rho}$ almost surely, which directly implies the second-moment upper bound $\Sigma_x \preceq \bar{\rho} I_{d+d_a}$. The bound $\rho I_{d+d_a} \preceq \Sigma_x \preceq \bar{\rho} I_{d+d_a}$ means that $\rho\|v\|_2^2 \le \mathbb{E}[(v^\top x_t)^2] \le \bar{\rho}\|v\|_2^2$ for every joint state–action direction v. Its lower bound requires variation in every such direction; its upper bound controls the second moment. This is a population excitation condition of the kind used in linear regression and system identification, where

lower bounds on covariate Gram matrices control estimation error (Matni & Tu, 2019). It is not a claim of empirical excitation for a dependent trajectory.

The proof requires neither independence of z_t and a_t nor zero cross-covariance. Define $\Sigma_z = \mathbb{E}[z_t z_t^\top]$, $\Sigma_a = \mathbb{E}[a_t a_t^\top]$, and $\Sigma_{za} = \mathbb{E}[z_t a_t^\top] = \Sigma_{az}^\top$. When $\Sigma_z \succ 0$, the Schur complement gives $\Sigma_x \succ 0$ if and only if the linear action innovation has positive definite second-moment matrix,

$$\Sigma_{a|z}^{\mathrm{lin}} := \Sigma_a - \Sigma_{az}\Sigma_z^{-1}\Sigma_{za} \succ 0. \tag{49}$$

Here the innovation $a_t - \Sigma_{az}\Sigma_z^{-1} z_t$ is the residual after the best linear prediction of the action from the state. This uncentered second moment need not equal a conditional covariance. Thus state-dependent actions are allowed. For standardized scalar z_t, a_t with correlation γ, the eigenvalues of Σ_x are $1 - |\gamma|$ and $1 + |\gamma|$. Strong correlation is compatible with coverage, but makes ρ small and weakens the loss-to-geometry bounds. A fixed positive lower bound is therefore a quantitative assumption, not merely the exclusion of exact linear dependence. Neither $d = d_a$ nor invertibility of B implies it.

Singular joint second moment and the data subspace. For a transition distribution with $\mathbb{E}\|x_t\|_2^2 < \infty$, define $\mathcal{S}_x := \mathrm{range}(\Sigma_x) = (\ker \Sigma_x)^\perp$. This is the smallest linear subspace containing x_t almost surely: $v \in \ker \Sigma_x$ if and only if $\mathbb{E}[(v^\top x_t)^2] = 0$, or $v^\top x_t = 0$ almost surely. It is a population subspace, distinct from a finite sample span. Let $U_x \in \mathbb{R}^{(d+d_a)\times r_x}$ have orthonormal columns spanning $\mathcal{S}_x$, where $r_x = \mathrm{rank}(\Sigma_x) > 0$, and set $\widetilde{x}_t = U_x^\top x_t$, $\widetilde{\Sigma}_x = U_x^\top \Sigma_x U_x$, and $\Pi_x = U_x U_x^\top$. Coverage on this subspace means $\rho I_{r_x} \preceq \widetilde{\Sigma}_x \preceq \bar{\rho} I_{r_x}$, equivalently $\rho \Pi_x \preceq \Sigma_x \preceq \bar{\rho}\Pi_x$. For a fixed finite-dimensional distribution, some such constants always exist on its nonzero support; a useful uniform lower bound is additional.

Writing $\widetilde{Q} = U_x^\top Q U_x$, the trace identity gives $\mathcal{L}_G(c) = \|\widetilde{\Sigma}_x^{1/2}\widetilde{Q}\widetilde{\Sigma}_x^{1/2}\|_F^2$, hence $\rho^2\|\widetilde{Q}\|_F^2 \le \mathcal{L}_G(c) \le \bar{\rho}^2\|\widetilde{Q}\|_F^2$. In particular, zero loss identifies $U_x^\top Q U_x = 0$, not necessarily the separate blocks of Q, because U_x can mix states and actions. For example, take $d = d_a = 1$, $c = 1$, and $z_t = a_t$ uniformly in $\{-1, 1\}$. With $A = -1/2$ and $B = 1/2$, we have $\Delta z_t = -a_t$ and $\mathcal{L}_G(1) = 0$, although $A \neq I_1$ and $B^\top B \neq I_1$; even B is invertible in this example. Thus arbitrary projection onto $\mathcal{S}_x$ does not preserve the full straightening or rollout conclusions. Assumption 4.1 retains positive definite joint coverage in the original state–action coordinates.

Dimension requirement. Under joint coverage, the zero-loss geometry $B^\top B = cI_{d_a}$ is feasible whenever $d \ge d_a$. For $d = d_a$, $B/\sqrt{c}$ is orthogonal; for $d > d_a$, its columns are orthonormal and its image has dimension d_a. If $d < d_a$, rank deficiency gives $\|B^\top B - cI_{d_a}\|_F^2 \ge c^2(d_a - d)$, so $\mathcal{L}_G(c) \ge \rho^2 c^2 (d_a - d) > 0$. Likewise, the ε-CGS regime with $\varepsilon < 1$ requires $d \ge d_a$, since $B^\top B \succeq c(1-\varepsilon) I_{d_a}$. This dimensional restriction concerns the action map B, independently of whether Σ_x is singular.

D.2 Planning Hessian and Dimension Cases

Recall $\Gamma_H = [A^{H-1}B, \ldots, B]$, $G_H = \Gamma_H^\top \Gamma_H$, and $P_H = H^{-1}(\mathbf{1}_H \mathbf{1}_H^\top) \otimes I_{d_a}$, with $n_H = H \times d_a$. For an integer $H \ge 1$ and $0 \le \varepsilon < 1$, define

$$\begin{aligned} s_H(\varepsilon) &:= (1+\sqrt{\varepsilon})^{H-1} - 1, \\ r_H(\varepsilon) &:= s_H(\varepsilon) - (H-1)\sqrt{\varepsilon}, \\ \xi_H(\varepsilon) &:= \varepsilon + 2(H-1)\sqrt{1+\varepsilon}\,\varepsilon \\ &\qquad + (1+\varepsilon)\left(2r_H(\varepsilon) + s_H(\varepsilon)^2\right). \end{aligned} \tag{50}$$

D.2.1 Proof of Theorem 4.4

The ideal projector. The averaging matrix $H^{-1}\mathbf{1}_H\mathbf{1}_H^\top$ is symmetric, idempotent, and rank one. Hence P_H is an orthogonal projector of rank d_a, with range $\{\mathbf{1}_H \otimes u : u \in \mathbb{R}^{d_a}\}$ and kernel $\{\mathbf{a} : \sum_{t=0}^{H-1} a_t = 0\}$. When $A = I_d$ and $B^\top B = cI_{d_a}$, every block of G_H is cI_{d_a}, so $G_H = cHP_H$. This gives the reference geometry with the correct rank: equal curvature on cumulative-action directions and zero curvature on cancelling directions. When CGS is approximate, the positive eigenspace of G_H need not equal $\mathrm{range}(P_H)$; the norm bound below compares the matrices without assuming their eigenspaces coincide.

The CGS perturbation bound. Write $A = I_d + D$. For every integer $k \geq 0$,

$$A^k = I_d + kD + R_k, \qquad R_k := \sum_{j=2}^{k} \binom{k}{j} D^j. \tag{51}$$

Since $\|D\|_{\mathrm{op}} \leq \sqrt{\varepsilon}$, for $0 \leq k \leq H-1$,

$$\begin{aligned} \|A^k - I_d\|_{\mathrm{op}} &\leq (1+\sqrt{\varepsilon})^k - 1 \leq s_H(\varepsilon), \\ \|R_k\|_{\mathrm{op}} &\leq (1+\sqrt{\varepsilon})^k - 1 - k\sqrt{\varepsilon} \leq r_H(\varepsilon). \end{aligned} \tag{52}$$

The third ε-CGS condition gives

$$\|B\|_{\mathrm{op}}^2 = \lambda_{\max}(B^\top B) \leq c(1+\varepsilon). \tag{53}$$

Index the block rows and columns of G_H by action times $t, s \in \{0, \ldots, H-1\}$, and set $p = H-1-t$ and $q = H-1-s$. This preserves the ordering used in the main context $\Gamma_H = [A^{H-1}B, \ldots, B]$, so

$$[G_H]_{ts} = B^\top (A^p)^\top A^q B. \tag{54}$$

Let $F_k := A^k - I_d = kD + R_k$. Since every block of cHP_H is cI_{d_a},

$$[G_H]_{ts} - cI_{d_a} = (B^\top B - cI_{d_a}) + B^\top F_p^\top B + B^\top F_q B + B^\top F_p^\top F_q B. \tag{55}$$

Using Corollary 4.3 and Eq. (53),

$$\begin{aligned} \|B^\top F_p^\top B\|_{\mathrm{op}} &\leq p\|B\|_{\mathrm{op}}\|D^\top B\|_{\mathrm{op}} + \|B\|_{\mathrm{op}}^2\|R_p\|_{\mathrm{op}}, \\ \|B^\top F_q B\|_{\mathrm{op}} &\leq q\|B\|_{\mathrm{op}}\|D^\top B\|_{\mathrm{op}} + \|B\|_{\mathrm{op}}^2\|R_q\|_{\mathrm{op}}, \\ \|B^\top F_p^\top F_q B\|_{\mathrm{op}} &\leq \|B\|_{\mathrm{op}}^2\|F_p\|_{\mathrm{op}}\|F_q\|_{\mathrm{op}}. \end{aligned} \tag{56}$$

Therefore, uniformly in t, s,

$$\|[G_H]_{ts} - cI_{d_a}\|_{\mathrm{op}} \leq c\varepsilon + 2(H-1)c\sqrt{1+\varepsilon}\,\varepsilon + c(1+\varepsilon)\left(2r_H(\varepsilon) + s_H(\varepsilon)^2\right) = c\,\xi_H(\varepsilon). \tag{57}$$

Let $E_H := G_H - cHP_H$. For a block vector $v = [v_0^\top, \ldots, v_{H-1}^\top]^\top \in \mathbb{R}^{n_H}$,

$$\|E_H v\|_2^2 \leq \sum_{t=0}^{H-1} \left(c\xi_H(\varepsilon) \sum_{s=0}^{H-1} \|v_s\|_2 \right)^2 \leq c^2 H^2 \xi_H(\varepsilon)^2 \|v\|_2^2. \tag{58}$$

Hence $\|G_H - cHP_H\|_{\mathrm{op}} \leq cH\xi_H(\varepsilon)$. To verify the leading coefficient, put $h = H-1$. For fixed H, $s_H = h\sqrt{\varepsilon} + \frac{1}{2}h(h-1)\varepsilon + O_H(\varepsilon^{3/2})$ and $r_H = \frac{1}{2}h(h-1)\varepsilon + O_H(\varepsilon^{3/2})$. The coefficient of ε in ξ_H is therefore $1 + 2h + h(h-1) + h^2 = 2H^2 - 3H + 2$, proving the stated expansion. This establishes the bound on $\|G_H/(cH) - P_H\|_{\mathrm{op}}$ for any $d \geq d_a$.

Comparison with temporal straightening and control isotropy. Retain $\|D\|_{\mathrm{op}}^2 \leq \varepsilon$ and $\|B^\top B - cI_{d_a}\|_{\mathrm{op}} \leq c\varepsilon$. These conditions already control the mixed drift–control defect indirectly:

$$\|D^\top B\|_{\mathrm{op}} \leq \|D\|_{\mathrm{op}}\|B\|_{\mathrm{op}} \leq \sqrt{c(1+\varepsilon)\varepsilon} = O(\sqrt{\varepsilon}). \tag{59}$$

Using this bound in Eq. (55) gives

$$\begin{aligned} \left\| \frac{G_H}{cH} - P_H \right\|_{\mathrm{op}} &\leq \varepsilon + (1+\varepsilon)(2s_H + s_H^2) \\ &= 2(H-1)\sqrt{\varepsilon} + O_H(\varepsilon). \end{aligned} \tag{60}$$

For $H \geq 2$, this square-root order is generally unavoidable under these two conditions alone.

CGS directly controls the same mixed defect at the geometric error scale,

$$\|D^\top B\|_{\mathrm{op}} \leq \sqrt{c}\,\varepsilon, \tag{61}$$

and hence

$$\|B^\top D B\|_{\mathrm{op}} \leq \|D^\top B\|_{\mathrm{op}}\|B\|_{\mathrm{op}} \leq c\sqrt{1+\varepsilon}\,\varepsilon.$$

The linear drift–control terms are therefore $O(\varepsilon)$, yielding the $O_H(\varepsilon)$ Hessian perturbation bound in Theorem 4.4. This compares the geometric consequences of the two bounds under matched drift and control-isotropy tolerances; it does not identify the CGS and temporal-straightening training objectives. At $H = 1$, both bounds reduce to the control-isotropy error ε.

Controllability Gramian and effective conditioning. Recall the finite-horizon controllability Gramian and extend the effective condition number from the main context to any nonzero matrix:

$$W_H = \Gamma_H \Gamma_H^\top = \sum_{k=0}^{H-1} A^k B B^\top (A^\top)^k, \qquad \kappa_{\text{eff}}(M) := \sigma_{\max}(M)/\sigma_{\min}^+(M). \tag{62}$$

Here $M \neq 0$, $\sigma_{\min}^+(M)$ is its smallest positive singular value, and κ denotes the Euclidean condition number on a nonsingular restriction. For symmetric positive semidefinite M, this ratio also equals $\lambda_{\max}(M)/\lambda_{\min}^+(M)$. The matrices $G_H = \Gamma_H^\top \Gamma_H$ and $W_H = \Gamma_H \Gamma_H^\top$ have the same positive eigenvalues and singular values, namely the squared positive singular values of Γ_H. Hence $\kappa_{\text{eff}}(G_H) = \kappa_{\text{eff}}(W_H)$. In particular, $\kappa_{\text{eff}}(G_H) = \kappa_{\text{eff}}(\Gamma_H)^2$; conditioning the horizon map itself therefore takes the square root of the Gram-matrix condition-number bound.

D.2.2 Square and rectangular action maps

We distinguish the dimension cases used by all three planners below, extending Remark 4.5. Throughout this discussion, assume $d \geq d_a$, the three ε-CGS bounds with $0 \leq \varepsilon < 1$, and $\xi_H < 1$. In particular, $B^\top B \succeq c(1-\varepsilon) I_{d_a}$, so B has full column rank.

State and action subspaces. Define the finite-horizon controllable state subspace and the action subspace that affects the terminal state by

$$\mathcal{C}_H := \text{range}(W_H) = \text{span}\{\text{range}(A^k B) : 0 \leq k < H\}, \qquad \mathcal{T}_H := \text{range}(G_H) = \text{range}(\Gamma_H^\top). \tag{63}$$

Their common dimension is $r_{\text{ctrl}} := \text{rank}(\Gamma_H)$, with $d_a \leq r_{\text{ctrl}} \leq \min\{d, n_H\}$. These subspaces are determined by the dynamics and horizon, and differ from the data subspace $\mathcal{S}_x$. A thin singular value decomposition $\Gamma_H = U_\Gamma \Sigma_\Gamma V_\Gamma^\top$, where $\Sigma_\Gamma = \text{diag}(\sigma_1(\Gamma_H), \ldots, \sigma_{r_{\text{ctrl}}}(\Gamma_H))$ retains its positive singular values, gives $W_H = U_\Gamma \Sigma_\Gamma^2 U_\Gamma^\top$ and $G_H = V_\Gamma \Sigma_\Gamma^2 V_\Gamma^\top$. Thus both restrictions are positive definite and $\kappa_{\text{eff}}(G_H) = \kappa(G_H|_{\mathcal{T}_H}) = \kappa(W_H|_{\mathcal{C}_H})$. Restricting to these subspaces removes the exact null directions.

If $d = d_a$, full column rank makes B invertible, and the last block of Γ_H gives $r_{\text{ctrl}} = d_a$. If $d > d_a$, r_{ctrl} can exceed d_a because propagation through A can reach state directions outside $\text{range}(B)$.

Bounds inherited from CGS. The proof of Theorem 4.4 applies to rectangular B, giving $\|G_H/(cH) - P_H\|_{\text{op}} \leq \xi_H$. To extend the notation from the main context to rectangular action maps, let $\mathcal{S}_H$ be a subspace spanned by orthonormal eigenvectors associated with the largest $r = d_a$ eigenvalues of G_H. In the square case this is exactly the subspace used in the main context $\mathcal{S}_H = \text{range}(G_H) = \mathcal{T}_H$; in the rectangular case $\mathcal{S}_H$ can be a strict subspace of $\mathcal{T}_H$. In orthonormal coordinates on $\mathcal{S}_H$, Theorem 4.4 and Weyl's inequality give

$$cH(1-\xi_H) I_r \preceq G_H|_{\mathcal{S}_H} \preceq cH(1+\xi_H) I_r, \qquad \kappa(G_H|_{\mathcal{S}_H}) \leq \frac{1+\xi_H}{1-\xi_H}. \tag{64}$$

Let Π_H denote the orthogonal projector onto $\mathcal{S}_H$. Unlike the fixed ideal projector P_H, Π_H depends on the learned dynamics; the two agree at ideal CGS geometry. These symbols are used throughout the planner analysis. When $r_{\text{ctrl}} = r$, $\mathcal{S}_H = \mathcal{T}_H$ and every direction that contributes to the planning cost is included. When $r_{\text{ctrl}} > r$, the spectral bound above covers only the leading r directions. The corresponding leading eigenspace of W_H has the same spectral bounds, by the singular value decomposition above. This proves the geometric assertions of Remark 4.5.

Bounds on all controllable directions. The following conditions extend the bound from $\mathcal{S}_H$ to $\mathcal{T}_H$. Let $\Pi_{\mathcal{C}_H}$ denote the orthogonal projector onto $\mathcal{C}_H$.

Corollary D.1 (Conditioning on the full controllable subspace)**.** *Under the conditions of this discussion, if $r_{\text{ctrl}} = d_a$, then*

$$\kappa_{\text{eff}}(G_H) = \kappa(W_H|_{\mathcal{C}_H}) \leq \frac{1+\xi_H}{1-\xi_H}. \tag{65}$$

A sufficient condition is $A\ \text{range}(B) \subseteq \text{range}(B)$. More generally, if a quantitative controllability bound $W_H \succeq \gamma_H cH \Pi_{\mathcal{C}_H}$ holds with a specified $\gamma_H > 0$, then $\kappa_{\text{eff}}(G_H) \leq (1+\xi_H)/\gamma_H$.

Proof. If $r_{\mathrm{ctrl}} = d_a$, the leading d_a eigenvalues are all of the positive eigenvalues, so Eq. (64) proves Eq. (65). Under the stated invariance condition, induction gives $\mathrm{range}(A^k B) \subseteq \mathrm{range}(B)$ for all k. Since Γ_H contains B as its last block and B has full column rank, $r_{\mathrm{ctrl}} = d_a$. For the general bound, quantitative controllability gives $\lambda^+_{\min}(W_H) \geq \gamma_H c H$, while Theorem 4.4 and Weyl's inequality give $\lambda_{\max}(W_H) = \lambda_1(G_H) \leq cH(1 + \xi_H)$. Taking the ratio proves the claim. □

Remark D.2 (Additional controllable directions)**.** When $r_{\mathrm{ctrl}} > d_a$, the remaining positive eigenvalues satisfy $0 < \lambda_j(G_H) \leq cH\xi_H$ for $d_a < j \leq r_{\mathrm{ctrl}}$. Controllability makes these eigenvalues positive, but a uniform lower bound requires further information, even after restriction to $\mathcal{C}_H$. For example, let $d = 2$, $d_a = 1$, $H = 2$, $c = 1$, and $0 < \delta < 1/4$. With e_1, e_2 the standard basis, take $B = e_1$ and $A = I_2 + \delta e_2 e_1^\top$. Then $\|D\|^2_{\mathrm{op}} = \delta^2$, $D^\top B = 0$, and $B^\top B = 1$, so all three CGS bounds hold with $\varepsilon = \delta^2$ and $\xi_2(\varepsilon) = \delta^2(1 + \sqrt{1 + \delta^2})^2 < 1$. Direct calculation gives

$$\Gamma_2 = \begin{bmatrix} 1 & 1 \\ \delta & 0 \end{bmatrix}, \qquad G_2 = \begin{bmatrix} 1 + \delta^2 & 1 \\ 1 & 1 \end{bmatrix}, \qquad W_2 = \begin{bmatrix} 2 & \delta \\ \delta & \delta^2 \end{bmatrix}. \tag{66}$$

For every $\delta > 0$, both G_2 and W_2 are positive definite, so $\mathcal{C}_2 = \mathbb{R}^2$. Their eigenvalues are $\lambda_\pm = (2 + \delta^2 \pm \sqrt{4 + \delta^4})/2$. As $\delta \downarrow 0$, $\lambda_+ \to 2$ and $\lambda_- \sim \delta^2/2$, hence $\kappa_{\mathrm{eff}}(G_2) \sim 4/\delta^2$, while $\|G_2/2 - P_2\|_{\mathrm{op}} = \delta^2/2$. The additional reachable direction becomes weak as the dynamics approach ideal CGS geometry, consistently with the dominant-spectrum bound.

Relation to temporal straightening. The controllable-subspace formulation follows Wang et al. (2026b, Remark 4.5 and Appendix C.2). Their Remark C.6 also notes that a lower bound on the smallest positive Gramian eigenvalue requires additional controllability assumptions. In the square case, their lower-bound argument uses $\|B^\top u\|_2 \geq \sigma_{\min}(B)\|u\|_2$ for every state vector u. For rectangular B, this inequality holds on $\mathrm{range}(B)$, which may be smaller than $\mathcal{C}_H$. Thus the shared nonzero-spectrum identity extends to all dimensions, while quantitative conditioning on the full controllable subspace uses conditions such as those in Corollary D.1.

Implications for the planner guarantees. For either dimension case, the MPPI, CEM, and gradient-descent statements below use $r = d_a$ and the leading eigenspace $\mathcal{S}_H$ defined above. Their respective hypotheses are unchanged; Eq. (64) supplies the spectral bounds on these directions. CEM additionally requires the perturbation and mean neighborhood conditions in Proposition D.7. When $r_{\mathrm{ctrl}} = d_a$, $\mathcal{S}_H = \mathcal{T}_H$, so these guarantees cover all action directions that affect the terminal state. Under the more general quantitative controllability condition, the exact MPPI map in Proposition D.4 also gives $\|J_{\mathrm{M}}|_{\mathcal{T}_H}\|_{\mathrm{op}} \leq (1 + s\gamma_H)^{-1}$, where $s = \sigma_{\mathrm{p}}^2 cH/\tau$: its eigenvalues on $\mathcal{T}_H$ are $(1 + s\lambda_j(G_H)/(cH))^{-1}$.

D.3 Quadratic Planning and Gaussian Proposals

D.3.1 Planning cost and nonlinear rollouts

Exact quadratic expansion. Let $\mathbf{a}^\star$ be any unconstrained minimizer of $C(\mathbf{a}) = \frac{1}{2}\|A^H z_0 + \Gamma_H \mathbf{a} - z_g\|_2^2$. The normal equation gives

$$\Gamma_H^\top (A^H z_0 + \Gamma_H \mathbf{a}^\star - z_g) = 0. \tag{67}$$

Therefore

$$C(\mathbf{a}) = C(\mathbf{a}^\star) + \tfrac{1}{2}(\mathbf{a} - \mathbf{a}^\star)^\top G_H (\mathbf{a} - \mathbf{a}^\star). \tag{68}$$

This identity is exact even when the goal is not reachable.

Planning cost gap. For a candidate sequence $\mathbf{a}$, define $\Delta(\mathbf{a}) := C(\mathbf{a}) - C(\mathbf{a}^\star)$. Equation (68) gives $\Delta(\mathbf{a}) = \frac{1}{2}\|G_H^{1/2}(\mathbf{a} - \mathbf{a}^\star)\|_2^2$. For a proposal mean μ_i, write $\Delta_i := \Delta(\mu_i)$. The minimum $C(\mathbf{a}^\star)$ need not be zero.

Nonlinear rollout. In practice, for a nonlinear terminal predictor $z_H = \widehat{F}_H(\mathbf{a})$,

$$\nabla^2_{\mathbf{a}} C = J_{\widehat{F}_H}^\top J_{\widehat{F}_H} + \sum_{j=1}^{d} (\widehat{F}_H(\mathbf{a}) - z_g)_j \nabla^2_{\mathbf{a}} (\widehat{F}_H)_j. \tag{69}$$

Nonlinear rollouts introduce an additional residual-weighted second-order term beyond $J_{\widehat{F}_H}^\top J_{\widehat{F}_H}$. This term need not be positive semidefinite and can alter the Hessian spectrum or introduce negative curvature, complicating the relationship between control geometry and sampling-based planner updates. Our analysis focuses on linear dynamical models; extending it to nonlinear rollouts is an exciting direction for future work.

Spectral assumptions for the planner bounds. The CGS-dependent results below assume the three ε-CGS bounds with $0 \le \varepsilon < 1$ from Corollary 4.3. We use $M_H = G_H/(cH)$, $r = d_a$, and the subspaces of Section D.2.2; each result states its additional requirement on ξ_H.

D.3.2 Gaussian proposals and update rules

Let $\mathsf{P}_i := \mathcal{N}(\mu_i, \Sigma_i)$ be the proposal distribution at iteration i. Draw K independent action sequences $\mathbf{a}^{(1)}, \dots, \mathbf{a}^{(K)} \sim \mathsf{P}_i$.

Proposal whitening. For a general positive-definite covariance Σ, set

$$X := \Sigma^{-1/2}(\mathbf{a} - \mathbf{a}^\star), \qquad \nu := \Sigma^{-1/2}(\mu - \mathbf{a}^\star). \tag{70}$$

In these proposal-whitened coordinates, the planning Hessian becomes

$$\Sigma^{1/2} G_H \Sigma^{1/2}. \tag{71}$$

For a general proposal, isotropy relevant to the sampling update is therefore measured by this proposal-whitened Hessian, not by G_H alone. Even an isotropic planning cost can become anisotropic in these coordinates if the proposal scales directions unequally. The MPPI and CEM bounds in Sections D.4 and D.5 use $\Sigma = \sigma_{\mathrm{p}}^2 I_{n_H}$, where $\sigma_{\mathrm{p}} > 0$ is the sampling standard deviation, as in the main text, so that Theorem 4.4 applies directly.

MPPI. Following path-integral control and MPPI (Kappen, 2005; Theodorou et al., 2010; Williams et al., 2017), for temperature $\tau > 0$, MPPI assigns soft weights

$$w_k := \exp\!\left(-\frac{C(\mathbf{a}^{(k)})}{\tau}\right) \tag{72}$$

and updates the proposal mean by the self-normalized estimator

$$\widehat{\mu}_{i+1}^{\mathrm{M}} := \frac{\sum_{k=1}^K w_k \mathbf{a}^{(k)}}{\sum_{k=1}^K w_k}. \tag{73}$$

Adding a constant to C does not change the normalized weights. The population map is

$$\mu_{i+1}^{\mathrm{M}} := \frac{\mathbb{E}_{\mathsf{P}_i}[w(\mathbf{a})\mathbf{a}]}{\mathbb{E}_{\mathsf{P}_i}[w(\mathbf{a})]}. \tag{74}$$

The analysis below conditions on the proposal covariance.

CEM. Following the cross-entropy method (Rubinstein, 1999; de Boer et al., 2005), for an elite fraction $\alpha \in (0,1)$, let $K_{\mathrm{elite}} = \alpha K \in \mathbb{N}$ be the number of samples with lowest cost. The empirical mean and covariance updates are

$$\begin{aligned} \widehat{\mu}_{i+1}^{\mathrm{C}} &:= \frac{1}{K_{\mathrm{elite}}} \sum_{j \in \mathcal{E}_i} \mathbf{a}^{(j)}, \\ \widehat{\Sigma}_{i+1}^{\mathrm{C}} &:= \frac{1}{K_{\mathrm{elite}}} \sum_{j \in \mathcal{E}_i} (\mathbf{a}^{(j)} - \widehat{\mu}_{i+1}^{\mathrm{C}})(\mathbf{a}^{(j)} - \widehat{\mu}_{i+1}^{\mathrm{C}})^\top. \end{aligned} \tag{75}$$

where $\mathcal{E}_i$ is the elite index set. The theory conditions on the current covariance and analyzes the population mean map. A full coupled mean–covariance convergence result is not claimed.

D.3.3 A SHARED BOUND FOR REPEATED MEAN UPDATES

Corollary D.3 (Mean error with finite samples and updates)**.** *Fix A, B, z_0, z_g, the proposal covariance Σ, and the projector Π_H. Let μ_i be the proposal mean after i updates and define $\nu_i := \Sigma^{-1/2}(\mu_i - \mathbf{a}^\star)$ and $e_i := \Pi_H \nu_i$. For fixed $I \geq 1$ and $\zeta \in (0,1)$, define*

$$\mathcal{D} := \{\nu : \|\nu\|_2 \leq R_\nu,\ \|\Pi_H \nu\|_2 \leq R_e\}, \qquad T_{\text{exit}} := \inf\{i \geq 0 : \nu_i \notin \mathcal{D}\}, \tag{76}$$

with $\inf \varnothing = \infty$. Let $\mathcal{F}_i$ be the history before batch i is drawn, so ν_i is $\mathcal{F}_i$-measurable. Let the population map T satisfy $\|\Pi_H T(\nu)\|_2 \leq q\|\Pi_H \nu\|_2$, with $0 < q < 1$, for $\nu \in \mathcal{D}$. Suppose each fresh K-sample update obeys $\|\Pi_H(\widehat{T}_{K,i}(\nu) - T(\nu))\|_2 \leq \epsilon_K(\zeta/I)$ with conditional probability at least $1 - \zeta/I$ given $\mathcal{F}_i$, uniformly for the current mean in $\mathcal{D}$. The one-step sampling estimates used below take $R = R_\nu$. Assume $\|e_0\|_2 \leq R_e$ and $\epsilon_K(\zeta/I) \leq (1-q)R_e$. There is an event of probability at least $1 - \zeta$ on which every update with $0 \leq i < \min(I, T_{\text{exit}})$ satisfies $\|e_{i+1}\|_2 \leq q\|e_i\|_2 + \epsilon_K(\zeta/I)$, and, whenever $I \leq T_{\text{exit}}$,

$$\|e_I\|_2 \leq q^I \|e_0\|_2 + \frac{1-q^I}{1-q}\epsilon_K(\zeta/I). \tag{77}$$

For the budget statement, let $0 < \epsilon_{\text{mean}} < \|e_0\|_2$ be a mean-error tolerance, distinct from the planning-cost tolerance in the main context ϵ_{tar}. If $\epsilon_K(\eta) \leq C_{\text{samp}}\sqrt{[r + \log(1/\eta)]/K}$, then $\|e_I\|_2 \leq \epsilon_{\text{mean}}$ is ensured on the same event whenever $I \leq T_{\text{exit}}$, up to logarithmic factors, by

$$I \gtrsim \frac{\log(2\|e_0\|_2/\epsilon_{\text{mean}})}{-\log q}, \qquad K \gtrsim \frac{C_{\text{samp}}^2}{(1-q)^2\epsilon_{\text{mean}}^2}\left(r + \log\frac{I}{\zeta}\right). \tag{78}$$

Proof. Let B_i be the event that the batch-i sampling bound fails. On $\{i < T_{\text{exit}}\}$, the conditional failure probability given $\mathcal{F}_i$ is at most ζ/I. Since this event is $\mathcal{F}_i$-measurable, the tower property and a union bound give

$$\mathbb{P}\left(\bigcup_{i=0}^{I-1}(\{i < T_{\text{exit}}\} \cap B_i)\right) \leq \sum_{i=0}^{I-1}\mathbb{E}\big[\mathbf{1}_{\{i<T_{\text{exit}}\}}\mathbb{P}(B_i \mid \mathcal{F}_i)\big] \leq \zeta. \tag{79}$$

On the complementary event, $\|e_{i+1}\|_2 \leq q\|e_i\|_2 + \epsilon_K(\zeta/I)$ for every $0 \leq i < \min(I, T_{\text{exit}})$. The invariance condition keeps the active iterates inside the radius-R_e ball through these updates, including the output of the last such update. Thus, on this event, exit cannot be caused by the active-radius condition alone; full-mean boundedness is not implied. Unrolling the recursion gives Eq. (77) whenever $I \leq T_{\text{exit}}$. It suffices to require $q^I\|e_0\|_2 \leq \epsilon_{\text{mean}}/2$ and $\epsilon_K(\zeta/I) \leq (1-q)\epsilon_{\text{mean}}/2$; solving these inequalities gives Eq. (78). □

Probability interpretation. Let $E_I := \{I \leq T_{\text{exit}}\}$ and let G_I denote the event that Eq. (77) holds. The guarantee is $\mathbb{P}(E_I \cap G_I^c) \leq \zeta$, not a conditional success probability $\mathbb{P}(G_I \mid E_I) \geq 1 - \zeta$. If $\mathbb{P}(E_I) > 0$, it only implies $\mathbb{P}(G_I^c \mid E_I) \leq \zeta/\mathbb{P}(E_I)$. When full-mean boundedness holds almost surely throughout the input iterations, active-radius invariance yields the unconditional $1 - \zeta$ error guarantee. No exit-probability bound is asserted: bounding only e_i does not control drift in the other coordinates. The condition $I \leq T_{\text{exit}}$ concerns $\nu_0, \ldots, \nu_{I-1}$ and does not require $\nu_I \in \mathcal{D}$. Updating the proposal covariance, changing the dynamics, or replanning from a new initial state requires corresponding uniform bounds.

D.4 MPPI: MEAN UPDATE AND PLANNING COST

We use r, $\mathcal{S}_H$, and Π_H as defined in Section D.2.2; the formulas below apply to both square and rectangular action maps.

D.4.1 POPULATION MEAN UPDATE

Proposition D.4 (Exact MPPI population map)**.** *Let $X \sim \mathcal{N}(\nu, I_{n_H})$, $M_H = G_H/(cH)$, and $s = \sigma_{\text{p}}^2 cH/\tau$. Define the tilted law*

$$\pi_\nu^{\text{M}}(x) \propto \exp\Big(-\frac{s}{2}x^\top M_H x\Big)\mathcal{N}(x; \nu, I_{n_H}). \tag{80}$$

Then

$$T_{\mathrm{M}}(\nu) := \mathbb{E}_{\pi_\nu^{\mathrm{M}}}[X] = (I_{n_H} + sM_H)^{-1}\nu, \qquad J_{\mathrm{M}} = (I_{n_H} + sM_H)^{-1}. \tag{81}$$

If $\xi_H < 1$, then on the top-r eigenspace $\mathcal{S}_H$,

$$q_{\mathrm{M}} := \|J_{\mathrm{M}}|_{\mathcal{S}_H}\|_{\mathrm{op}} = \frac{1}{1 + (\sigma_{\mathrm{p}}^2/\tau)\lambda_r(G_H)} \le \frac{1}{1 + s(1-\xi_H)}, \tag{82}$$

and

$$\kappa(J_{\mathrm{M}}|_{\mathcal{S}_H}) \le \frac{1 + s(1+\xi_H)}{1 + s(1-\xi_H)}. \tag{83}$$

At ideal geometry,

$$J_{\mathrm{M}}^0 = \frac{1}{1+s}P_H + (I_{n_H} - P_H), \tag{84}$$

and

$$\|J_{\mathrm{M}} - J_{\mathrm{M}}^0\|_{\mathrm{op}} \le s\xi_H. \tag{85}$$

Proof. The log unnormalized density in Eq. (80) is

$$-\frac{1}{2}\|x-\nu\|_2^2 - \frac{s}{2}x^\top M_H x. \tag{86}$$

Completing the square gives a Gaussian with precision $I_{n_H} + sM_H$ and mean $(I_{n_H} + sM_H)^{-1}\nu$, proving Eq. (81). Since J_{M} is a matrix function of M_H, they share eigenvectors. On $\mathcal{S}_H$, Theorem 4.4 gives

$$1 - \xi_H \le \lambda_j(M_H) \le 1 + \xi_H. \tag{87}$$

Applying $u \mapsto (1+su)^{-1}$ proves Eqs. (82) and (83); the smallest dominant eigenvalue also gives $q_{\mathrm{M}} = [1 + (\sigma_{\mathrm{p}}^2/\tau)\lambda_r(G_H)]^{-1}$. Substituting $M_H = P_H$ and using $P_H^2 = P_H$ proves Eq. (84). Finally, the resolvent identity gives

$$J_{\mathrm{M}} - J_{\mathrm{M}}^0 = (I_{n_H} + sM_H)^{-1} s(P_H - M_H)(I_{n_H} + sP_H)^{-1}. \tag{88}$$

Both inverse factors have operator norm at most one because $M_H, P_H \succeq 0$, which proves Eq. (85). □

For $d = d_a$, $r = d$ and $\mathcal{S}_H = \mathrm{range}(G_H)$, so this proves Theorem 4.6 with the same s and q_{M} as in the main context.

The contraction depends on both shape and scale. CGS controls spectral spread through ξ_H, but cosine matching does not identify c. Therefore τ and σ_{p}^2 must still be calibrated so that $s = \sigma_{\mathrm{p}}^2 cH/\tau$ is non-negligible.

D.4.2 Sampling error in one update

Assume $\xi_H < 1$ and $\|\nu\|_2 \le R$. All samples in a batch are independent.

Proposition D.5 (Sampling error in one MPPI update)**.** *For MPPI, draw $X_1, \dots, X_K \sim \mathcal{N}(\nu, I_{n_H})$ and define*

$$\widetilde{w}_k := \exp\Big(-\frac{s}{2}X_k^\top M_H X_k\Big), \qquad \widehat{T}_{\mathrm{M},K} := \frac{\sum_{k=1}^K \widetilde{w}_k X_k}{\sum_{k=1}^K \widetilde{w}_k}. \tag{89}$$

These weights use the cost gap Δ in place of C, which leaves normalized weights unchanged. Let

$$\underline{Z}_{\mathrm{M}}(R) := \frac{e^{-R^2/2}}{(1+s(1+\xi_H))^{r/2}(1+s\xi_H)^{(n_H-r)/2}}. \tag{90}$$

There is a universal constant $C_{\mathrm{conc}} > 0$ such that, for every $\zeta \in (0,1)$, if

$$K \ge \frac{8}{\underline{Z}_{\mathrm{M}}(R)^2}\log\frac{4}{\zeta}, \tag{91}$$

then, with probability at least $1-\zeta$,

$$\Big\|\Pi_H(\widehat{T}_{\mathrm{M},K} - T_{\mathrm{M}}(\nu))\Big\|_2 \le \frac{C_{\mathrm{conc}}(1+R)}{\underline{Z}_{\mathrm{M}}(R)}\sqrt{\frac{r\log(4r/\zeta)}{K}}. \tag{92}$$

Proof. Let $Z := \mathbb{E}[\widetilde{w}_1]$. A Gaussian integral gives

$$Z = \det(I_{n_H} + sM_H)^{-1/2} \exp\!\big(-\tfrac{1}{2}\nu^\top sM_H(I_{n_H} + sM_H)^{-1}\nu\big)\,. \tag{93}$$

The top r eigenvalues of M_H are at most $1 + \xi_H$, the remaining eigenvalues are at most ξ_H, and $0 \preceq sM_H(I_{n_H} + sM_H)^{-1} \preceq I_{n_H}$. Therefore $Z \geq \underline{Z}_{\mathrm{M}}(R)$. Since $0 < \widetilde{w}_k \leq 1$, Hoeffding's inequality (Hoeffding, 1963) and Eq. (91) imply, with probability at least $1 - \zeta/2$,

$$\left| K^{-1} \sum_{k=1}^{K} \widetilde{w}_k - Z \right| \leq \sqrt{\frac{\log(4/\zeta)}{2K}} \leq \frac{\underline{Z}_{\mathrm{M}}(R)}{2}. \tag{94}$$

In an orthonormal basis of $\mathcal{S}_H$, each coordinate of $\widetilde{w}_k \Pi_H X_k$ has centered sub-Gaussian norm at most $C_0(1 + R)$. Coordinate-wise concentration and a union bound give, with probability at least $1 - \zeta/2$,

$$\begin{aligned} &\left\| K^{-1} \sum_{k=1}^{K} \widetilde{w}_k \Pi_H X_k - \mathbb{E}[\widetilde{w}_1 \Pi_H X_1] \right\|_2 \\ &\qquad \leq C_1(1 + R)\sqrt{\frac{r \log(4r/\zeta)}{K}}. \end{aligned} \tag{95}$$

On the intersection of these events, the empirical denominator is at least $Z/2$. Since $\mathbb{E}[\widetilde{w}_1 \Pi_H X_1] = Z \Pi_H T_{\mathrm{M}}(\nu)$ and $\|T_{\mathrm{M}}(\nu)\|_2 \leq \|\nu\|_2 \leq R$, the standard ratio decomposition and $Z \geq \underline{Z}_{\mathrm{M}}(R)$ prove Eq. (92) after absorbing numerical constants into C_{conc}. □

The bound controls the sampling error in $\mathcal{S}_H$. The remaining coordinates can still affect the importance weights, which is why $n_H - r$ appears in the normalizer bound. Combining it with Corollary D.3 gives a projected mean-error bound for repeated updates. The next result states the corresponding planning cost bound when all nonzero eigenvalues are covered.

D.4.3 Planning cost gap for MPPI

The result in the main context reports the planning cost at the proposal mean. Assume $\xi_H < 1$ and $r_{\mathrm{ctrl}} = r$, so $\mathcal{S}_H = \mathrm{range}(G_H)$. Keep $A, B, z_0, z_g, \sigma_{\mathrm{p}}$, and τ fixed, and define

$$\nu_i := \sigma_{\mathrm{p}}^{-1}(\mu_i - \mathbf{a}^\star), \qquad \Delta_i := C(\mu_i) - C(\mathbf{a}^\star) = \tfrac{1}{2}\sigma_{\mathrm{p}}^2 \nu_i^\top G_H \nu_i. \tag{96}$$

The identity follows from Eq. (68). The rank condition includes $d = d_a$ and the rectangular case with $r_{\mathrm{ctrl}} = d_a$, as characterized in Section D.2.2. Use a fresh batch of K samples at each update and define T_{exit} as in Eq. (76), here with $\mathcal{D} = \{\nu : \|\nu\|_2 \leq R\}$. Let $\epsilon_K(\zeta/I)$ be the right-hand side of Eq. (92) with failure probability ζ/I. Define the importance-weight factor $\mathcal{A}_{\mathrm{M}}(R) := C_{\mathrm{conc}}(1 + R)/\underline{Z}_{\mathrm{M}}(R)$, with $\underline{Z}_{\mathrm{M}}(R)$ from Eq. (90), and set

$$b_K := \sigma_{\mathrm{p}} \sqrt{cH(1 + \xi_H)/2}\, \epsilon_K(\zeta/I). \tag{97}$$

The minimum sample count in Eq. (91) is also required with failure probability ζ/I.

Corollary D.6 (Planning cost with finite samples and updates)**.** *Under these conditions, there is an event of probability at least $1 - \zeta$ on which the following bounds hold in the indicated ranges:*

$$\begin{aligned} \sqrt{\Delta_{i+1}} &\leq q_{\mathrm{M}} \sqrt{\Delta_i} + b_K, && 0 \leq i < \min(I, T_{\mathrm{exit}}), \\ \Delta_I &\leq \left[q_{\mathrm{M}}^I \sqrt{\Delta_0} + \frac{1 - q_{\mathrm{M}}^I}{1 - q_{\mathrm{M}}} b_K \right]^2, && I \leq T_{\mathrm{exit}}. \end{aligned} \tag{98}$$

For $0 < \epsilon_{\mathrm{tar}} < \Delta_0$, it suffices to choose

$$I \geq \left\lceil \frac{\log(4\Delta_0/\epsilon_{\mathrm{tar}})}{-2 \log q_{\mathrm{M}}} \right\rceil, \qquad b_K \leq \frac{1 - q_{\mathrm{M}}}{2} \sqrt{\epsilon_{\mathrm{tar}}} \tag{99}$$

to obtain $\Delta_I \leq \epsilon_{\mathrm{tar}}$ on the same event whenever $I \leq T_{\mathrm{exit}}$. These guarantees use the same stopped-process interpretation as Corollary D.3, not success probabilities conditional on non-exit. Substituting the sampling bound gives sufficient sample and update budgets for the cost bound in the main context.

Proof. Apply the stopped union-bound argument of Eq. (79) to the full-mean radius-R region. The fresh-batch estimate holds conditionally on the past whenever the input mean is in this region. Thus, on an event of probability at least $1-\zeta$, $\|\Pi_H(\nu_{i+1} - J_{\mathrm{M}}\nu_i)\|_2 \le \epsilon_K(\zeta/I)$ for all $0 \le i < \min(I, T_{\mathrm{exit}})$. Since $\mathcal{S}_H = \mathrm{range}(G_H)$, $G_H^{1/2}(I_{n_H} - \Pi_H) = 0$. Theorem 4.4 gives $\lambda_{\max}(G_H) \le cH(1+\xi_H)$, hence

$$\frac{\sigma_{\mathrm{p}}}{\sqrt{2}}\|G_H^{1/2}(\nu_{i+1} - J_{\mathrm{M}}\nu_i)\|_2 \le b_K. \tag{100}$$

Moreover, J_{M} commutes with $G_H^{1/2}$ and has operator norm q_{M} on $\mathrm{range}(G_H)$, so

$$\frac{\sigma_{\mathrm{p}}}{\sqrt{2}}\|G_H^{1/2}J_{\mathrm{M}}\nu_i\|_2 \le q_{\mathrm{M}}\frac{\sigma_{\mathrm{p}}}{\sqrt{2}}\|G_H^{1/2}\nu_i\|_2 = q_{\mathrm{M}}\sqrt{\Delta_i}. \tag{101}$$

The triangle inequality proves the one-update bound before exit. Summing the geometric series proves the bound after I updates whenever $I \le T_{\mathrm{exit}}$. The choices in Eq. (99) bound each term in the square brackets by $\sqrt{\epsilon_{\mathrm{tar}}}/2$. □

In the rectangular case, if $\mathcal{S}_H$ contains only the leading d_a eigenvectors and additional positive eigenvalues remain outside it, the same argument bounds the cost contribution $\frac{1}{2}\sigma_{\mathrm{p}}^2\nu_i^\top\Pi_H G_H\Pi_H\nu_i$. A bound for the full planning cost gap requires control of all positive eigenvalues, as in Corollary D.1.

D.5 CEM: Local Mean Update

Fix an elite fraction $\alpha \in (0,1)$ and use $r = \mathrm{rank}(P_H) = d_a$, $\mathcal{S}_H$, and Π_H from Section D.2.2.

D.5.1 Population mean update near the optimum

For $X \sim \mathcal{N}(\nu, I_{n_H})$, let $t_\alpha(M,\nu)$ be the α-quantile of $X^\top MX$, and define

$$T_{\mathrm{C}}(\nu; M) := \mathbb{E}[X \mid X^\top MX \le t_\alpha(M,\nu)]\,. \tag{102}$$

Proposition D.7 (Exact local CEM Jacobian and projector limit)**.** *For every nonzero $M \succeq 0$, $T_{\mathrm{C}}(0;M) = 0$, and, for $X \sim \mathcal{N}(0, I_{n_H})$,*

$$\begin{aligned} J_{\mathrm{C}}(M) &:= D_\nu T_{\mathrm{C}}(0;M) \\ &= \frac{1}{\alpha}\mathbb{E}[XX^\top \mathbf{1}\{X^\top MX \le t_\alpha(M,0)\}]\,. \end{aligned} \tag{103}$$

Let $q_{\alpha,r}$ be the α-quantile of χ_r^2, and define

$$\beta_{\alpha,r} := \frac{1}{r}\mathbb{E}\big[\chi_r^2 \mid \chi_r^2 \le q_{\alpha,r}\big] \in (0,1). \tag{104}$$

At $M = P_H$,

$$J_{\mathrm{C}}(P_H) = \beta_{\alpha,r}P_H + (I_{n_H} - P_H). \tag{105}$$

For $0 \le \delta < 1/2$, let $\Pi_r(M)$ project onto the top r eigenvectors of M, and define

$$\omega_{\alpha,n_H,r}(\delta) := \sup_{\substack{M\succeq 0 \\ \|M-P_H\|_{\mathrm{op}}\le\delta}} \big\|\Pi_r(M)J_{\mathrm{C}}(M)\Pi_r(M) - \beta_{\alpha,r}\Pi_r(M)\big\|_{\mathrm{op}}. \tag{106}$$

Then

$$\omega_{\alpha,n_H,r}(\delta) \longrightarrow 0 \qquad \text{as } \delta \downarrow 0. \tag{107}$$

Consequently, if $\xi_H < 1/2$, the top-r subspace $\mathcal{S}_H$ of M_H is invariant under $J_{\mathrm{C}}(M_H)$ and, in orthonormal coordinates on this subspace,

$$\|J_{\mathrm{C}}(M_H)|_{\mathcal{S}_H} - \beta_{\alpha,r}I_r\|_{\mathrm{op}} \le \omega_{\alpha,n_H,r}(\xi_H). \tag{108}$$

If $\beta_{\alpha,r} + \omega_{\alpha,n_H,r}(\xi_H) < 1$, then every

$$q_{\mathrm{C}} \in (\beta_{\alpha,r} + \omega_{\alpha,n_H,r}(\xi_H), 1) \tag{109}$$

satisfies $\|\Pi_H T_{\mathrm{C}}(\nu; M_H)\|_2 \le q_{\mathrm{C}}\|\Pi_H\nu\|_2$ for all sufficiently small full mean vectors ν, where $\Pi_H = \Pi_r(M_H)$. In particular, it is a local contraction factor for means in $\mathcal{S}_H$.

Proof. For fixed nonzero $M \succeq 0$, define

$$F(\nu, t) := \mathbb{P}_{X \sim \mathcal{N}(\nu, I_{n_H})}(X^\top M X \le t). \tag{110}$$

For nonzero $M \succeq 0$, the quadratic form has a positive continuous density for $t > 0$, and its α-quantile is strictly positive. The Gaussian density is smooth in ν, so the implicit function theorem applies to $F(\nu, t_\alpha(M, \nu)) = \alpha$ near $\nu = 0$ and gives a differentiable quantile. By central symmetry,

$$\nabla_\nu F(0, t) = \mathbb{E}[X \mathbf{1}\{X^\top M X \le t\}] = 0, \tag{111}$$

so $D_\nu t_\alpha(M, 0) = 0$. Differentiating the unnormalized elite first moment under the integral sign gives Eq. (103).

For $M = P_H$, decompose $X = P_H X + (I_{n_H} - P_H)X$. The components are independent, and the elite event is $\|P_H X\|_2^2 \le q_{\alpha,r}$. Rotational symmetry on $\mathrm{range}(P_H)$ gives conditional second moment $\beta_{\alpha,r} P_H$; the orthogonal component remains standard Gaussian and the cross moments vanish. This proves Eq. (105). Conditioning a non-degenerate χ^2_r variable on a strict lower-tail event reduces its mean, so $0 < \beta_{\alpha,r} < 1$.

To prove continuity, let $M_n \to P_H$. For $X \sim \mathcal{N}(0, I_{n_H})$, $X^\top M_n X \to X^\top P_H X$ almost surely. The limit has the continuous χ^2_r law, so the corresponding α-quantiles converge to $q_{\alpha,r}$. The elite indicators then converge almost surely outside a null boundary event. Since $\|XX^\top\|_{\mathrm{op}} = \|X\|_2^2$ is integrable, dominated convergence in Eq. (103) gives $J_{\mathrm{C}}(M_n) \to J_{\mathrm{C}}(P_H)$. The spectral gap of P_H also gives $\Pi_r(M_n) \to P_H$ whenever $\|M_n - P_H\|_{\mathrm{op}} < 1/2$. These two facts imply Eq. (107) by contradiction.

In an eigenbasis of M_H, the elite event depends only on squared coordinates. All off-diagonal conditional second moments vanish, so $J_{\mathrm{C}}(M_H)$ commutes with M_H and leaves $\mathcal{S}_H$ invariant. The balance bound follows from the definition of $\omega_{\alpha,n_H,r}$. To obtain the projected contraction, write $\nu = e + \eta$, with $e = \Pi_H \nu$ and $\eta = (I_{n_H} - \Pi_H)\nu$. Reflection symmetry of the active coordinates gives $\Pi_H T_{\mathrm{C}}(\eta; M_H) = 0$. Continuity of the derivative near zero bounds $\|\Pi_H D_\nu T_{\mathrm{C}}(\nu; M_H)\Pi_H\|_{\mathrm{op}}$ by q_{C} on a sufficiently small ball. Integrating along $\eta + te$, $0 \le t \le 1$, proves the claimed contraction. □

Multiplying the cost by a positive scalar does not change the CEM elite set. Thus its ideal contraction depends on α and r, while CGS controls the deviation from a balanced active update through ξ_H. When CEM also updates covariance, repeated application requires a uniform bound on the proposal-whitened matrices $\Sigma_i^{1/2} G_H \Sigma_i^{1/2}$; no full coupled mean–covariance contraction is asserted.

Conditions and dimension cases. The modulus $\omega_{\alpha,n_H,r}$ gives continuity at the ideal projector; no linear rate in ξ_H is asserted. Contraction requires both $\xi_H < 1/2$ and $\beta_{\alpha,r} + \omega_{\alpha,n_H,r}(\xi_H) < 1$, with the full mean inside the neighborhood specified in the proposition. When $r_{\mathrm{ctrl}} = r$, the result covers every direction that affects the cost. When $r_{\mathrm{ctrl}} > r$, it controls the leading r directions; the additional positive eigenvalues still enter the elite event and need not have a comparable contraction factor.

D.5.2 Finite-sample and repeated mean updates

For the one-step sampling bound, assume $\xi_H < 1$ and $\|\nu\|_2 \le R$. All samples in a batch are independent.

Proposition D.8 (Sampling error in one CEM update)**.** *For CEM, draw a fresh batch $X_1, \dots, X_K \sim \mathcal{N}(\nu, I_{n_H})$, assume $K_{\mathrm{elite}} := \alpha K \in \mathbb{N}$, let $\widehat{t}_\alpha$ be the K_{elite}-th order statistic of $Y_j := X_j^\top M_H X_j$, and define*

$$\widehat{T}_{\mathrm{C},K} := \frac{1}{K_{\mathrm{elite}}} \sum_{j=1}^{K} X_j \mathbf{1}\{Y_j \le \widehat{t}_\alpha\}. \tag{112}$$

There is a universal constant $C_{\mathrm{conc}} > 0$ such that, for every $\zeta \in (0,1)$, with probability at least $1 - \zeta$,

$$\begin{aligned} &\left\| \Pi_H (\widehat{T}_{\mathrm{C},K} - T_{\mathrm{C}}(\nu; M_H)) \right\|_2 \\ &\quad \le \frac{C_{\mathrm{conc}}(1+R)}{\alpha} \sqrt{\frac{r \log(6r/\zeta)}{K}} + \frac{R_{\mathrm{C},K}(R,\zeta)}{\alpha} \sqrt{\frac{\log(6/\zeta)}{2K}}, \end{aligned} \tag{113}$$

where

$$R_{\mathrm{C},K}(R,\zeta) := R + \sqrt{r} + \sqrt{2\log(3K/\zeta)}. \tag{114}$$

Thus the projected CEM mean update converges to its population map at the canonical $K^{-1/2}$ rate, up to logarithmic factors.

Proof. Let F_Y and $F_{Y,K}$ be the population and empirical CDFs of $Y = X^\top M_H X$, and let t_α be its population α-quantile. The law of Y is continuous because M_H has at least r positive eigenvalues. The Dvoretzky–Kiefer–Wolfowitz inequality gives, with probability at least $1 - \zeta/3$,

$$\sup_t |F_{Y,K}(t) - F_Y(t)| \le u_K := \sqrt{\frac{\log(6/\zeta)}{2K}}. \tag{115}$$

With probability one there are no ties, so $F_{Y,K}(\widehat{t}_\alpha) = \alpha$. On the event above, $|F_{Y,K}(t_\alpha) - \alpha| \le u_K$; because threshold sets are nested, the two selected index sets differ in at most Ku_K samples.

A Gaussian norm tail bound and a union bound give, with probability at least $1 - \zeta/3$,

$$\max_{1\le j\le K} \|\Pi_H X_j\|_2 \le R_{\mathrm{C},K}(R,\zeta). \tag{116}$$

Thus replacing t_α by $\widehat{t}_\alpha$ changes the projected empirical first moment by at most $R_{\mathrm{C},K}(R,\zeta)u_K$.

At the fixed threshold t_α, every coordinate of $\Pi_H X_j \mathbf{1}\{Y_j \le t_\alpha\}$ is sub-Gaussian with centered norm at most $C_0(1+R)$. Therefore, with probability at least $1 - \zeta/3$,

$$\begin{aligned} &\left\| K^{-1}\sum_{j=1}^{K} \Pi_H X_j \mathbf{1}\{Y_j \le t_\alpha\} - \mathbb{E}[\Pi_H X \mathbf{1}\{Y \le t_\alpha\}] \right\|_2 \\ &\qquad \le C_1(1+R)\sqrt{\frac{r\log(6r/\zeta)}{K}}. \end{aligned} \tag{117}$$

Both empirical and population CEM means divide their first moments by α. Combining the three events proves Eq. (113). □

Repeated updates. Assume the contraction conditions of Proposition D.7, and choose q_{C} and a full-mean neighborhood radius R_ν as in that proposition. Keep the proposal covariance fixed and draw a fresh independent batch at each update. Let $e_i = \Pi_H \nu_i$ and let $\epsilon_K^{\mathrm{C}}(\eta)$ denote the right-hand side of Eq. (113) with $R = R_\nu$ and failure probability η. Assume $\|e_0\|_2 \le R_e$ and $\epsilon_K^{\mathrm{C}}(\zeta/I) \le (1 - q_{\mathrm{C}})R_e$, and define T_{exit} by Eq. (76) with these radii. Corollary D.3 gives an event of probability at least $1 - \zeta$ on which, whenever $I \le T_{\mathrm{exit}}$,

$$\|e_I\|_2 \le q_{\mathrm{C}}^I \|e_0\|_2 + \frac{1 - q_{\mathrm{C}}^I}{1 - q_{\mathrm{C}}} \epsilon_K^{\mathrm{C}}(\zeta/I). \tag{118}$$

This controls the projected mean error, with the additional logarithms from the empirical elite threshold retained in ϵ_K^{C}. As in Corollary D.3, this statement bounds failure before exit; it does not condition on the trajectory remaining in the full-mean neighborhood. The covariance remains fixed throughout the analyzed updates.

D.6 Gradient Descent: Planning Cost Convergence

We consider unconstrained gradient descent on the quadratic cost C of Section D.3. Let $\mathbf{a}_i$ be its iterates, $\Delta_i := C(\mathbf{a}_i) - C(\mathbf{a}^\star)$, and $h > 0$ the step size.

D.6.1 Convergence from the CGS spectral bounds

Proposition D.9 (Gradient descent under CGS geometry)**.** *Assume the three ε-CGS bounds with $0 \le \varepsilon < 1$ and $\xi_H < 1$. Gradient descent with $h = 1/(cH)$ satisfies*

$$\begin{aligned} \mathbf{a}_{i+1} &= \mathbf{a}_i - h\nabla C(\mathbf{a}_i), \\ \mathbf{a}_{i+1} - \mathbf{a}^\star &= \left(I_{n_H} - \frac{G_H}{cH}\right)(\mathbf{a}_i - \mathbf{a}^\star). \end{aligned} \tag{119}$$

On the leading $r = d_a$ eigenspace $\mathcal{S}_H$,

$$\left\| \left(I_{n_H} - \frac{G_H}{cH} \right) \Big|_{\mathcal{S}_H} \right\|_{\mathrm{op}} \le \xi_H. \tag{120}$$

Define the cost contribution of these directions by

$$\Delta_i^{\mathrm{dom}} := \tfrac{1}{2}(\mathbf{a}_i - \mathbf{a}^\star)^\top \Pi_H G_H \Pi_H (\mathbf{a}_i - \mathbf{a}^\star). \tag{121}$$

For every integer $I \ge 1$,

$$\Delta_I^{\mathrm{dom}} \le \xi_H^{2I} \Delta_0^{\mathrm{dom}}. \tag{122}$$

If $r_{\mathrm{ctrl}} = r$, then $\Delta_i^{\mathrm{dom}} = \Delta_i$, and the same bound holds for the full planning cost gap.

Proof. Equation (68) gives $\nabla C(\mathbf{a}) = G_H(\mathbf{a} - \mathbf{a}^\star)$, proving the update formula. The projector Π_H commutes with G_H. In an eigenbasis, each of the leading r error coordinates is multiplied by $1 - \lambda_j(G_H)/(cH)$, whose absolute value is at most ξ_H by Theorem 4.4. Squaring each coordinate and weighting it by $\lambda_j(G_H)/2$ proves the one-step cost bound; iteration gives Eq. (122). When $\mathcal{S}_H = \mathrm{range}(G_H)$, all other directions have zero cost contribution, so $\Delta_i^{\mathrm{dom}} = \Delta_i$. □

Step size and full-spectrum conditioning. More generally, let $L = \lambda_{\max}(G_H)$ and $m = \lambda_{\min}^+(G_H) > 0$. The update on a positive-eigenvalue direction has multiplier $1 - h\lambda_j(G_H)$. Every such direction contracts when $0 < h < 2/L$. Minimizing $\max\{|1 - hm|, |1 - hL|\}$ gives $h_\star = 2/(L + m)$ and contraction factor $q_{\mathrm{GD}} = (\kappa_{\mathrm{eff}}(G_H) - 1)/(\kappa_{\mathrm{eff}}(G_H) + 1)$. Consequently, $\Delta_I \le q_{\mathrm{GD}}^{2I} \Delta_0$. When $r_{\mathrm{ctrl}} = r$, the CGS spectral bound gives $q_{\mathrm{GD}} \le \xi_H$. At ideal geometry, one step with $h = 1/(cH)$ reaches the minimum cost.

Dimension cases. If $d = d_a$, $r_{\mathrm{ctrl}} = r$ and the full cost bound applies. It also applies when $d > d_a$ but $r_{\mathrm{ctrl}} = d_a$, for example under the invariance condition in Corollary D.1. If $r_{\mathrm{ctrl}} > d_a$, the bound with factor ξ_H covers the dominant cost contribution. The remaining positive eigenvalues may be arbitrarily small, so their gradient updates may contract arbitrarily slowly. A quantitative bound on all positive eigenvalues, such as the one in Corollary D.1, instead controls q_{GD} for the full cost. These statements concern ordinary gradient descent on C; the Adam optimizer used in our experiments requires a separate optimizer analysis.

D.6.2 Comparison using temporal straightening and control isotropy

The bound below uses only $\|A - I_d\|_{\mathrm{op}}$ and $\|B^\top B - cI_{d_a}\|_{\mathrm{op}}$. It recovers the temporal-straightening-style conditioning argument in the square-action case. Proposition D.9 above uses the full CGS spectral bound, including drift–control decoupling.

Corollary D.10 (Bound in the square-action case)**.** *Suppose $d_a = d$, and define $\delta_A := \|A - I_d\|_{\mathrm{op}} < 1$ and $\delta_B := c^{-1}\|B^\top B - cI_{d_a}\|_{\mathrm{op}} < 1$. Then B is invertible and*

$$\begin{aligned} \kappa_{\mathrm{eff}}(G_H) &= \kappa(W_H) \\ &\le \frac{1 + \delta_B}{1 - \delta_B} \frac{\sum_{k=0}^{H-1} (1 + \delta_A)^{2k}}{\sum_{k=0}^{H-1} (1 - \delta_A)^{2k}} \\ &\le \frac{1 + \delta_B}{1 - \delta_B} \left(\frac{1 + \delta_A}{1 - \delta_A} \right)^{2(H-1)}. \end{aligned} \tag{123}$$

Under the ε-CGS bounds,

$$\kappa_{\mathrm{eff}}(G_H) \le \frac{1 + \varepsilon}{1 - \varepsilon} \left(\frac{1 + \sqrt{\varepsilon}}{1 - \sqrt{\varepsilon}} \right)^{2(H-1)}. \tag{124}$$

Proof. For every unit vector x,

$$x^\top W_H x = \sum_{k=0}^{H-1} \|B^\top (A^\top)^k x\|_2^2. \tag{125}$$

Thus

$$\lambda_{\max}(W_H) \le \sigma_{\max}(B)^2 \sum_{k=0}^{H-1} \sigma_{\max}(A)^{2k},$$
$$\lambda_{\min}(W_H) \ge \sigma_{\min}(B)^2 \sum_{k=0}^{H-1} \sigma_{\min}(A)^{2k}. \tag{126}$$

The assumptions give $\sigma_{\max}(A) \le 1 + \delta_A$, $\sigma_{\min}(A) \ge 1 - \delta_A$, and $\kappa(B)^2 = \kappa(B^\top B) \le (1 + \delta_B)/(1 - \delta_B)$. Dividing the two bounds proves the first inequality in Eq. (123). The ratio of the two positive sums is at most the largest termwise ratio, proving the second. Under the ε-CGS bounds, $\delta_A \le \sqrt{\varepsilon}$ and $\delta_B \le \varepsilon$, which gives Eq. (124). □

D.7 Geometry under Other Training Objectives

In the trained world model, A and B are jointly induced by encoder and predictor parameters $\vartheta = (\theta, \phi)$. Prediction and other training objectives also shape these parameters, and their preferred representations may differ from those that minimize CGS alone. CGS can still improve control geometry while preserving the structure required by the other training objectives, even when its loss remains nonzero. The presence of another loss does not itself imply a positive loss floor: the objectives may share a zero-CGS solution.

Let $\mathcal{L}_{\text{other}} := \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{aux}}$ collect prediction and all weighted non-CGS auxiliary objectives, with $\mathcal{L}_{\text{aux}} = 0$ when none are present. For LeWM, $\mathcal{L}_{\text{aux}} = \lambda_{\text{sig}}\mathcal{L}_{\text{sig}}$, so $\mathcal{L}_{\text{other}} = \mathcal{L}_{\text{LeWM}}$. Using the same total-loss notation as in Eq. (8), consider

$$\mathcal{L}(\vartheta) = \mathcal{L}_{\text{other}}(\vartheta) + \lambda_{\text{CGS}}\mathcal{L}_{\text{CGS}}(\vartheta), \tag{127}$$

and the nonempty set of models meeting a prescribed tolerance $\ell_{\text{other}} \ge 0$ for these other objectives,

$$\Theta_{\ell_{\text{other}}} := \{\vartheta : \mathcal{L}_{\text{other}}(\vartheta) \le \ell_{\text{other}}\}. \tag{128}$$

A positive loss floor arises if these requirements keep the Gram defect uniformly away from zero. One sufficient condition is that every $\vartheta \in \Theta_{\ell_{\text{other}}}$ must retain temporal drift

$$\|A_\vartheta - I_d\|_{\text{op}} \ge \delta_{\text{drift}} > 0, \tag{129}$$

and Assumption 4.1 holds uniformly on this set with coverage constant at least ρ. Write $D_\vartheta := A_\vartheta - I_d$ and keep $c > 0$ fixed. Then Proposition 4.2 implies

$$\inf_{\vartheta \in \Theta_{\ell_{\text{other}}}} \mathcal{L}_G^\vartheta(c) \ge \rho^2 \delta_{\text{drift}}^4. \tag{130}$$

Indeed, $\|D_\vartheta^\top D_\vartheta\|_F^2 \ge \|D_\vartheta\|_{\text{op}}^4 \ge \delta_{\text{drift}}^4$. This gives a positive CGS Gram-loss floor whenever fitting the other objectives requires irreducible temporal drift. More generally, the same argument applies whenever those objectives keep $\mathcal{R}(A, B)$ uniformly bounded away from zero over $\Theta_{\ell_{\text{other}}}$. A large value of $\|(A - I_d)z_t\|_2$ alone does not establish such a floor: $A - I_d$ may still be reducible without compromising the other objectives.

A positive loss floor limits how closely the model can approach ideal CGS geometry. Feasible adjustments to A and B can still improve control geometry. CGS can favor feasible reductions in drift–control coupling and adjustments to the control map B, even when temporal drift must be retained. The fixed-drift calculation below illustrates this remaining freedom through the Gram-defect surrogate; its optimizer describes the jointly trained model only when compatible with the other objectives.

D.7.1 Effect of irreducible drift on the control map

For fixed $D = A - I_d$, the B-dependent part of $\mathcal{R}(A, B)$ is

$$\mathcal{R}_B(D, B) := 2\|D^\top B\|_F^2 + \|B^\top B - cI_{d_a}\|_F^2. \tag{131}$$

Let $\lambda_k^D := \lambda_{d-k+1}(DD^\top)$, $1 \le k \le d$. Thus $0 \le \lambda_1^D \le \cdots \le \lambda_d^D$ lists the drift eigenvalues in nondecreasing order, while $\lambda_j(M)$ retains the nonincreasing convention from the main context.

Proposition D.11 (Optimal control map under fixed temporal drift)**.** *For fixed D and $d \geq d_a$, a minimizer of Eq.* (131) *selects the left singular directions of B from eigenspaces associated with the d_a smallest eigenvalues of $DD^\top$, with squared singular values*

$$\sigma_k^2(B_\star) = (c - \lambda_k^D)_+, \qquad k = 1, \ldots, d_a, \tag{132}$$

where $(u)_+ := \max\{u, 0\}$. The right singular vectors are arbitrary.

Proof. Let $\upsilon_k := \sigma_k^2(B)$, ordered as $\upsilon_1 \geq \cdots \geq \upsilon_{d_a} \geq 0$. Since

$$\begin{aligned} \|D^\top B\|_F^2 &= \operatorname{tr}(DD^\top BB^\top), \\ \|B^\top B - cI_{d_a}\|_F^2 &= \sum_{k=1}^{d_a} (\upsilon_k - c)^2. \end{aligned} \tag{133}$$

the second term is independent of the singular vectors. For fixed (υ_k), von Neumann's trace inequality pairs the largest υ_k with the smallest eigenvalues of $DD^\top$. Writing U_B for the left singular vectors,

$$\min_{U_B} \mathcal{R}_B(D, B) = \sum_{k=1}^{d_a} \left[2\lambda_k^D \upsilon_k + (\upsilon_k - c)^2\right]. \tag{134}$$

Each term is strictly convex on $\upsilon_k \geq 0$ and is minimized at $\upsilon_k = (c - \lambda_k^D)_+$. Since λ_k^D is nondecreasing, these minimizers are nonincreasing and respect the ordering of (υ_k). □

If $\dim \ker(D^\top) \geq d_a$, the defect objective can place the action map inside $\ker(D^\top)$ while retaining $B^\top B = cI_{d_a}$. Thus, in this surrogate, nonzero temporal drift can coexist with decoupled and balanced action effects. When action-controlled directions overlap with non-removable temporal dynamics, CGS favors weaker drift directions. Unequal $\lambda_k^D < c$ yield unequal positive singular values in $B_\star$, whereas all $\lambda_k^D \geq c$ yield zero singular values, even when the drift eigenvalues differ. A common reduction of all singular values is only a scale change; unequal reductions directly worsen control conditioning. This calculation minimizes the unweighted defect $\mathcal{R}$. For a general correlated Σ_x, spectral equivalence bounds $\mathcal{L}_G$ but does not make its minimizer identical to that of $\mathcal{R}$; the calculation illustrates the tradeoff in this surrogate.

Reacher as an illustrative case. Reacher illustrates a regime in which one-step action geometry is only an indirect reference for visual latent change. Joint torques affect Δz_t through acceleration, velocity, configuration, and momentum, so accurate prediction may require a non-negligible drift term in

$$\Delta z_t = Dz_t + Ba_t, \qquad D = A - I_d.$$

For fixed D, Proposition D.11 gives

$$\sigma_k^2(B^\star) = (c - \lambda_k^D)_+.$$

Thus stronger drift lowers the preferred action sensitivity along the corresponding latent direction. For fixed λ_k^D, overly large sensitivities receive stronger downward pressure, although differences in λ_k^D prevent any general ordering of how much the largest and smallest singular values change. The spectral spread can therefore shrink mainly through suppression of strong action sensitivities, while the weakest sensitivity changes little.

This distinction can affect planners differently. For quadratic planning, optimal fixed-step gradient descent depends on $\kappa_{\text{eff}}(G_H) = L/m$, whereas the slowest active-direction population MPPI contraction is

$$q_M = \frac{1}{1 + (\sigma_p^2/\tau)m}.$$

Hence a change dominated by reducing L with little increase in m can improve gradient-based conditioning more clearly than MPPI contraction. CEM may show a similarly limited gain because its elite update depends on the overall active spectral shape and the evolving proposal covariance, so reducing excessive curvature alone need not make elite selection substantially more balanced. This is consistent with the Reacher results, where straightening methods improve more significant under GD but modest on sampling-based planners.