WORLD MODELING THROUGH SPECTRAL ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

What information should a latent world model preserve? We study this question through the lens of spectral representation learning. We introduce SpecWM, a JEPA framework that explicitly specifies desired relationships between observations through a teacher similarity kernel and trains the encoder to reproduce those relationships in its embedding geometry. Different kernels provide different supervision, ranging from physical state information to a fully self-supervised temporal kernel requiring only the ordering of observations within trajectories. We characterize the global optima of the objective, establish conditions for linear recoverability and invariance to nuisance information, and connect temporal-kernel similarity to the behavioral successor measure under an idealized setting. Empirically, the choice of kernel strongly controls which physical quantities are recoverable from the representation. More importantly, temporal spectral alignment improves CEM planning success over LeWorldModel from 52% to 68% on OGBench Cube, 49% to 77% on OGBench Scene, and 29% to 51% on CALVIN. These results suggest that explicitly specifying representation geometry provides a simple way to steer what latent world models preserve for control.

1 INTRODUCTION

The goal of a world model is to encode sufficient information to predict the consequences of an agent's actions (Ha & Schmidhuber, 2018). At the same time, complex real-world observations contain large amounts of task-irrelevant information (Zhang et al., 2021). Learning a compact representation therefore hinges on deciding which aspects of an observation to retain. Joint Embedding Predictive Architectures (JEPAs; LeCun, 2022) predict outcomes in a latent space, rather than reconstructing high-dimensional observations in pixel space, allowing learned representations to abstract away details of the input. Due to the lack of explicit supervision or grounding in raw observation space, training JEPAs is generally non-trivial. The training objective must implicitly guide which distinctions the model preserves without requiring it to reproduce every detail of an observation.

Recent work has made substantial progress toward stable JEPA training through regularizing the embedding distribution (Balestriero & LeCun, 2025; Maes et al., 2026). However, avoiding collapse alone does not ensure that a representation preserves information needed for control, and the degree to which this happens is dictated implicitly by self-prediction (Tang et al., 2023). For instance, an encoder may distinguish robot configurations while ignoring the objects the robot must manipulate. We seek a direct method to express these relationships and understand their consequences.

We approach this problem through the lens of spectral representation learning (Belkin & Niyogi, 2003; Bengio et al., 2004), which historically focuses on extracting representations from pairwise relationships between samples. Prior work has shown that several contrastive and non-contrastive objectives generally admit spectral interpretations (HaoChen et al., 2021; Balestriero & LeCun, 2022). Building on this perspective, we make the pairwise notion of similarity between two observations the center of a new training objective for latent dynamics modeling.

We introduce a **spectral alignment** framework for training world models to preserve a chosen similarity structure. Leveraging a notion of distances between observations (the *teacher*), we define a target similarity kernel. We then train an encoder (the *student*) to learn embeddings whose pairwise similarities match the target and an action-conditioned predictor to output future latents. This framework can be instantiated through different teachers requiring varying degrees of supervision: when

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.

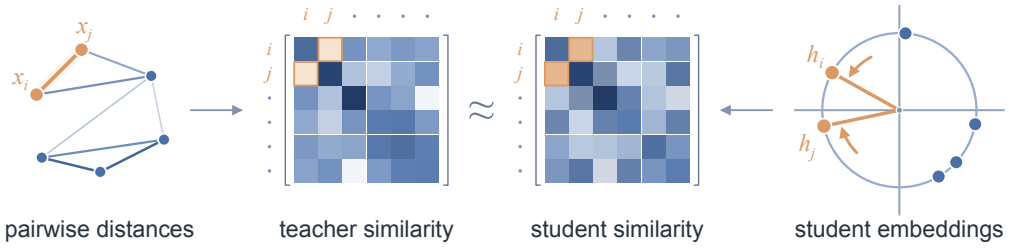

Figure 1: Spectral alignment framework. Pairwise distances (teacher) between samples define a kernel that an encoder (student) is trained to reconstruct through embedding similarities.

some information about the underlying system is known, encoders can be trained based on physical distances, but the natural ordering of states within a trajectory is also sufficient to provide a training signal. We name the resulting world model SpecWM.

Theoretically, we first study how the choice of teacher shapes the learned representation. When the teacher is specified for all pairs of states, we characterize the global optima as the kernel's spectral embeddings. This connection allows us to identify which quantities can be recovered from the representation via a linear probe. When teacher supervision is only available within a trajectory, we establish conditions under which matching the teacher removes observation noise from the embeddings. Finally, for the specific choice of temporal kernels, we connect the expected similarity target to the behavioral successor measure, which describes how likely the data-collection policy is to visit a given state in the future.

Practically, we instantiate this framework across visual manipulation environments spanning OGBench Scene and Cube (Park et al., 2025) as well as CALVIN (Mees et al., 2022). Empirically, we confirm that kernel choice substantially impacts which physical quantities are linearly recoverable. Further, when leveraging a self-supervised temporal teacher, whose only source of supervision is the order of observations in a trajectory, the resulting world models outperform existing JEPA world models (Maes et al., 2026) in planning performance.

Our contributions can be summarized as follows:

1. We introduce a novel world modeling objective based on spectral alignment.
2. We formally study this framework through the lens of spectral representation learning, with a particular focus on temporal kernels.
3. We present a broad empirical evaluation on visual control tasks and measure the ability of different kernels to recover physical quantities and drive planning performance.

2 Related Work

World models learn how environments respond to actions, enabling agents to plan and learn from predicted outcomes (Ha & Schmidhuber, 2018). Generative approaches predict future observations, often by modeling dynamics in a latent space and decoding predictions into images. PlaNet uses learned dynamics to evaluate candidate action sequences, while the Dreamer series trains policies on imagined trajectories (Hafner et al., 2019; 2025). More broadly, generative models support interactive environments and spatial world generation (Parker-Holder & Fruchter, 2025; Team HY-World, 2026; World Labs Team, 2026), with world-action models such as Cosmos Policy, DreamZero, and LingBotVA extending these capabilities to action generation for robot control (Kim et al., 2026; Ye et al., 2026; Li et al., 2026). In contrast, reconstruction-free world models focus prediction on representations without requiring them to reproduce observations. DINO-WM predicts features from a pretrained visual encoder (Zhou et al., 2025) while JEPAs such as LeWorldModel learn the representations alongside the dynamics (LeCun, 2022; Maes et al., 2026).

JEPA as proposed in LeCun (2022) features the central question of what anti-collapse mechanism is used to guide which information is retained and which is abstracted away. In the cases of I-

JEPA (Assran et al., 2023) and V-JEPA 2.1 (Mur-Labadia et al., 2026), the target encoder is an exponential moving average of the context encoder with a stop-grad, making collapse difficult to attain but not actually a non-optimal solution. LeJEPA (Balestriero & LeCun, 2025) solves anti-collapse by enforcing the distribution of latents to be an isotropic Gaussian, which prevents trivial collapse but does not require any particular information to be retained.

Spectral representation learning is intimately tied to self-supervised learning. HaoChen et al. (2021) interpret contrastive learning as spectral decomposition of an augmentation graph. Balestriero & LeCun (2022) extend this perspective, showing that several objectives including InfoNCE (van den Oord et al., 2019), VICReg (Bardes et al., 2022), and Barlow Twins (Zbontar et al., 2021) recover classical spectral embedding methods where the graph is defined implicitly using the choice of augmentations. Our spectral alignment objective extends this work by making previously implicit relations between samples the central object of study.

We cover additional families of related work in Appendix A.

3 PRELIMINARIES

Let $\{x_i\}_{i=1}^n$ be a collection of inputs and let $f_\theta : \mathcal{X} \to \mathbb{R}^d$ be an encoder parameterized by θ. We denote its representations by $h_i = f_\theta(x_i)$, which we stack as rows of $H \in \mathbb{R}^{n\times d}$. A symmetric matrix $K \in \mathbb{R}^{n\times n}$ can be used to specify the pairwise relationships between inputs. For instance, in an augmentation-based method, K_{ij} may be designated as 1 for two views produced from the same observation, and as 0 otherwise. More generally, K can be interpreted as the result of a symmetric similarity function — i.e., a *kernel* — $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which defines relationships through $K_{ij} = k(x_i, x_j)$.

Spectral embeddings. Given a teacher kernel K specifying the desired inner product between every pair of inputs in a collection, we can construct a coordinate vector for each input that realizes these pairwise relationships. Such a construction is called a *spectral embedding*. Suppose that K is symmetric and positive semidefinite, meaning that a valid spectral embedding exists. K then has eigendecomposition

$$K = \sum_{\ell=1}^{r} \lambda_\ell u_\ell u_\ell^\top = U_r \Lambda_r U_r^\top, \tag{1}$$

where $r = \operatorname{rank}(K)$, $\lambda_\ell > 0$ is the ℓ-th positive eigenvalue of K, $u_\ell \in \mathbb{R}^n$ is its corresponding unit eigenvector, $U_r = [u_1, \ldots, u_r]$, and $\Lambda_r = \operatorname{diag}(\lambda_1, \ldots, \lambda_r)$. A spectral embedding of K is given by $H_{\text{spec}} = U_r \Lambda_r^{1/2}$ where the i-th row h_i^{spec} provides the coordinates of sample i. Exactly satisfying all the inner products of K requires an embedding dimension $d \geq r$. When $d < r$, we get $H_d = U_d \Lambda_d^{1/2}$, for which $H_d H_d^\top = U_d \Lambda_d U_d^\top$ is the best rank-d approximation of K in Frobenius norm. Each eigenvalue determines the relative contribution of its corresponding eigenvector to the kernel, as seen in Equation 1, making eigenvalue rescaling a natural reweighting operation. To construct a particular spectral embedding, one selects d eigenvectors and assigns each a nonnegative scale s_ℓ yielding

$$H_{\text{spec}} = U_d \operatorname{diag}(s_1, \ldots, s_d), \qquad h_i^{\text{spec}} = \big(s_1 u_1(i), \ldots, s_d u_d(i)\big). \tag{2}$$

The canonical embedding of K is recovered through $s_\ell = \sqrt{\lambda_\ell}$; other choices reweight the spectrum.

Linear probing. To assess which physical quantities are linearly accessible from a representation, we fit a linear map from fixed embeddings to target labels. Given targets $Y \in \mathbb{R}^{n\times p}$, such as object positions, the optimal unregularized probing error is $\min_{W\in\mathbb{R}^{d\times p}} \|Y - HW\|_F^2$. This error is zero precisely when every column of Y lies in the column space of H, denoted $\operatorname{col}(H)$.

4 METHOD

We will state the general framework for arbitrary kernels, and later specialize it to practical kernels.

Building a target. We start from a distance d between pairs of observations (which could be derived from ground-truth measurements, if available, or simply by counting the number of observations that separate two observations in the data). Considering a sequence of observations $\{x_i\}_{i=1}^n$, we

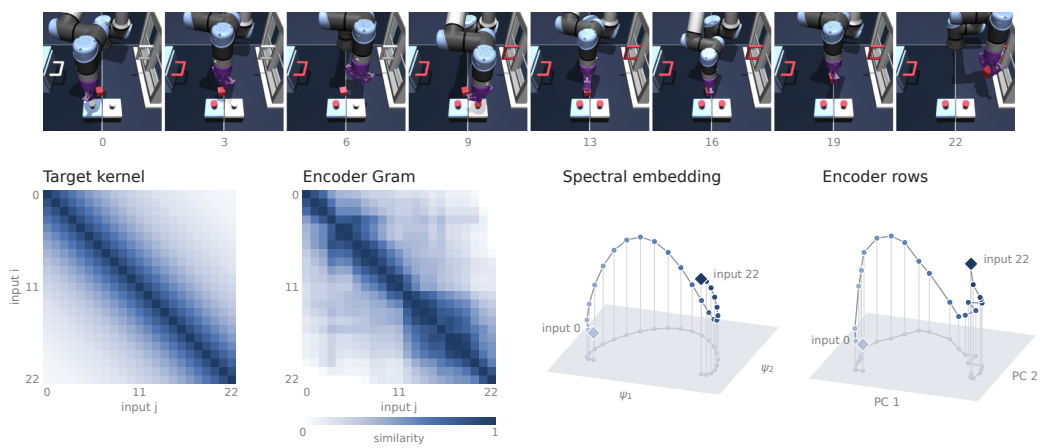

Figure 2: Spectral alignment on a single trajectory. Top: sampled observations from OGBench Scene (Park et al., 2025). Bottom: the temporal target kernel and encoder Gram matrix (before centering), the spectral embedding of K (top three eigenvectors), and the encoder rows h_i (top three principal components), colored by frame index. Matching pairwise similarities encourages the encoder to recover the target's geometry.

leverage the distance d to build a target *kernel* K_{ij}. In practice, we adopt a Laplacian kernel to convert distances into similarities:

$$K_{ij} = \exp\big(-d(x_i, x_j)/\sigma\big), \tag{3}$$

with values in $(0, 1]$. Its slope at $d = 0$ is $-1/\sigma$, so that observations that are close according to d remain distinguishable, while pairs more than a few σ apart have similarity near 0. The bandwidth σ therefore sets the range of distances the target resolves; we set it to the median teacher distance over the pairs the kernel is defined on. We then double center K to form $\tilde{K}$:

$$\tilde{K}_{ij} = K_{ij} - \frac{1}{n}\sum_k K_{ik} - \frac{1}{n}\sum_k K_{kj} + \frac{1}{n^2}\sum_{k,l} K_{kl}. \tag{4}$$

Without centering, the kernel would be dominated by a positive baseline, since kernel values concentrate above zero, and regressing this target would favor matching that shared baseline rather than the variation around it.

Still, if a few eigenvalues dominate the kernel, the encoder can achieve a small squared error by matching those directions while largely ignoring the others. We therefore *temper* the kernel by reshaping its spectrum before it is used as a target. We write the eigendecomposition of the centered kernel as $\tilde{K} = U\Lambda U^T$ with eigenvalues in decreasing order, and define Λ^α by raising the leading m positive eigenvalues to the power $\alpha \in [0, 1]$ and setting all remaining entries to zero. The target is then

$$K^\alpha = n\frac{U\Lambda^\alpha U^T}{\mathrm{tr}\Lambda^\alpha}. \tag{5}$$

At $\alpha = 1$, each eigenvector occupies the target's variance according to its eigenvalue fraction. At $\alpha = 0$, each eigenvector's share is uniform. In practice, we will use $\alpha = 1/2$, thus using the square root of the kernel matrix as target, and truncate the spectrum at $m = 128$.

Training the encoder. We implement the encoder f_θ as a deep neural network that maps each observation x_i to an embedding $h_i \in \mathbb{R}^d$. It will act as a *student*, and regress the kernel designed from the teacher distance. We first rescale each embedding to the length the target prescribes

$$\bar{h}_i = \sqrt{K^\alpha_{ii}}\frac{h_i}{\|h_i\|}, \tag{6}$$

and then compute the student's Gram matrix — the symmetric matrix of inner products of its rows — $S = \bar{H}\bar{H}^T$. Similarly to the kernel, we double center the Gram matrix as $\tilde{S}$, and regress the

target through a spectral loss:

$$\mathcal{L}_{\text{spec}} = \left\| \tilde{S} - K^{\alpha} \right\|_F^2 . \tag{7}$$

Prediction anchor. While $\mathcal{L}_{\text{spec}}$ is sufficient for training an encoder, planning in latent space requires learning a predictor, i.e., a neural network $g_\phi : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ parameterized by ϕ and trained to predict the latent outcome of applying some action from an action space $\mathcal{A}$ after receiving a given observation. For stability, we apply a parameter-free RMS normalization to the encoder output, so that every embedding has norm $\sqrt{d}$. While the spectral loss is invariant to this step, it fixes the scale of the embeddings seen by the predictor and the planner. We train a predictor by a simple regression objective in latent space, which is also backpropagated through the encoder, such that the total loss is

$$\mathcal{L} = \mathcal{L}_{\text{spec}} - \lambda \sum_i \big(\cos(G, H') \big)_i, \tag{8}$$

where G stacks predictions $g_\phi(f_\theta(x_i), a_i)$, H' stacks representations of next observations $f_\theta(x'_i)$ for each training transition (x_i, a_i, x'_i), and $\cos(\cdot)$ extracts the vector of row-wise cosine similarities from the two matrices.

Planning. Once a spectral world model is trained through the objective in Equation 8, it can be directly deployed in the standard planning pipeline for JEPA world models (Maes et al., 2026). In practice, we use the Cross-Entropy Method (CEM) (Rubinstein, 1999) to optimize action sequences from the current observation x_0: candidate action sequences are sampled from a Gaussian distribution, and the world model predicts the representation of observations T steps ahead for each. Candidates are then scored according to similarity to the embedding of a goal observation g:

$$\cos\left(g_\phi^T(f_\theta(x_0), a_0^{T-1}), f_\theta(g) \right), \tag{9}$$

where g_ϕ^T denotes recursive application of the predictor over T steps, and a_0^{T-1} is a sequence of actions. The action distribution is refit on the best candidates for a fixed number of iterations. We then execute a prefix of the selected action sequence before replanning from the new observation. We detail hyperparameters in Appendix C.

Kernel design. Often, robotics data is labeled by informative metadata such as proprioception or depth masks. In such cases, that metadata can be used to construct a *supervised* kernel. We evaluate three such kernels: state, depth, and proprioception. For each, we use the Euclidean distance between the corresponding feature vectors as the teacher distance. The state kernel uses the simulator state s_i underlying each observation x_i as its feature vector. Although this information is generally unavailable in real-world settings, it lets us evaluate a teacher with access to the full physical state. The proprioception and depth kernels use the robot's proprioceptive state and the flattened depth mask as feature vectors, respectively.

When metadata is not available, the ordering of observations in trajectories itself can form an *unsupervised* kernel. We use a temporal kernel, which relies on absolute difference between indices for observations ordered in a trajectory: $d(x_i, x_j) = |i - j|$. We also evaluate a quasimetric kernel as a close relative to the temporal kernel. This kernel relies on a distance d_q learned through quasimetric reinforcement learning (Wang et al., 2023; Wang & Isola, 2022), which estimates temporal distances between observations according to the optimal goal-reaching policy. While this potentially introduces estimation errors, it labels state pairs by the minimum number of steps between states, instead of expressing the temporal separations observed in the data, and can thus compensate for poor data quality. Since the student's Gram matrix is symmetric, we symmetrize the learned quasimetric $d(x_i, x_j) = d_q(x_i \to x_j) + d_q(x_j \to x_i)$.

While supervised kernels can describe relationships between all pairs of inputs, i.e., are *fully specified*, the unsupervised kernels we use only express relationships for pairs of inputs in the same trajectory. In this *underspecified* setting, cross-trajectory similarities are excluded from the training objective. We also use a different normalization of the spectral loss: we omit tempering and the target-diagonal rescaling in Equation 6. Each clip's squared error is normalized by the mean squared magnitude of its centered target before averaging across clips. While representations are fixed up to an orthogonal transformation in the fully specified case, underspecification allows the model to form its own abstractions (HaoChen & Ma, 2023).

5 THEORETICAL ANALYSIS

In this section, we build upon previous analyses (Bengio et al., 2004; Balestriero & LeCun, 2022; Blier et al., 2021) and provide formal statements regarding the spectral alignment objective and some intuition on their practical relevance. Full proofs of each theorem are provided in Appendix D.

5.1 THE SPECTRAL LOSS HAS A KNOWN OPTIMUM

A foundational property of our objective is that in the fully specified setting, an embedding table that minimizes $\mathcal{L}_{\text{spec}}$ recovers the row-normalized spectral embeddings of K^α (Bengio et al., 2004).

Theorem 1 (Spectral recovery)**.** *Suppose $m \leq d$ and $K^\alpha_{ii} > 0$ for all i. Then $\min_H \mathcal{L}_{spec}(H) = 0$, which is attained exactly at*

$$H^\star = \operatorname{diag}(c)\, H_{spec}\, Q, \tag{10}$$

with $c \in \mathbb{R}^n_{>0}$, $Q \in \mathbb{R}^{d\times d}$ an orthogonal matrix, and H_{spec} the spectral embedding of K^α with square-root eigenvalue scaling.

This result is useful as the spectral embeddings of the kernel can be calculated through eigendecomposition as described in Section 3. We can thus evaluate the probing qualities of spectral embeddings derived from arbitrary kernels without ever having to train a model.

5.2 DOWNSTREAM PROBING GUARANTEES

The ability to probe a given quantity from spectral embeddings depends on its relationship with the chosen kernel. Following prior work (Balestriero & LeCun, 2022), we show that a kernel constructed from label relationships can make the labels themselves linearly recoverable, without directly regressing encoder outputs onto them. Intuitively, this suggests that the choice of a certain kernel encourages embeddings to extract the information used to construct it; this motivates the supervised kernels evaluated in Section 6.3.

Theorem 2 (Linear recoverability)**.** *Let $Y \in \mathbb{R}^{n\times p}$ satisfy $\mathbf{1}^\top Y = 0$, row norms $\|y_i\| = 1$, and $\operatorname{rank}(Y) \leq d$. For $K = YY^\top$, $\alpha = 1$, and all positive spectral directions retained, every global minimizer $H^\star$ of $\mathcal{L}_{spec}$ subject to $\|h_i\| = \sqrt{d}$ satisfies*

$$\min_{W\in\mathbb{R}^{d\times p}} \|Y - H^\star W\|_F^2 = 0. \tag{11}$$

5.3 TASK-IRRELEVANT INFORMATION

A central goal of a JEPA is selective invariance— dropping irrelevant parts of an observation (LeCun, 2022). We define task-irrelevant information as any aspect of an observation the teacher kernel is invariant to. For example, a cost-to-go kernel for robotic manipulation may ignore the exact patterns of leaves rustling in the background. We ask whether matching such a teacher also makes the encoder ignore these details. Formally, consider a sequence of observations

$$\{x_i\}_{i=1}^n = \{(s_i, n_i)\}_{i=1}^n,$$

where s_i is the task-relevant component and n_i is the task-irrelevant component. Let $\nu(\cdot \mid s_i)$ denote the distribution of task-irrelevant values at state s_i, and let Z collect the task-relevant trajectory. We establish invariance under a coverage assumption: nuisance values that are individually possible at two states must also be possible together in a trajectory visiting both states. This excludes a background color that varies across clips but is always identical between frames of a clip.

Coverage. (A1) Fix a task-relevant trajectory Z and two distinct observation indices i and j. Draw $n \sim \nu(\cdot \mid s_i)$ and $n' \sim \nu(\cdot \mid s_j)$ independently. For almost every Z, all $i \neq j$, and every measurable set E of nuisance pairs, assume

$$\Pr\big((n, n') \in E \mid s_i, s_j\big) > 0 \quad \Longrightarrow \quad \Pr\big((n_i, n_j) \in E \mid Z\big) > 0.$$

Conditioning on the trajectory may change which nuisance pairs are likely, but cannot make an otherwise possible set of pairs impossible. This assumption covers independently sampled sensor noise and allows correlated disturbances, such as stochastic foliage motion or rainfall, when all otherwise possible combinations remain possible. It excludes persistent trajectory-level attributes, such as a background identity or lighting that varies across trajectories but is fixed within each one.

Theorem 3 (Abstraction of task-irrelevant information)**.** *Under the coverage assumption, fixed-norm encoder outputs, and the target-diagonal rescaling in Equation 6, if each trajectory's target matrix K^α has strictly positive diagonal entries almost surely and $\mathbb{E}[\mathcal{L}_{\mathrm{spec}}] = 0$, then for almost every clip and every frame i,*

$$f_\theta((s_i, n)) = f_\theta((s_i, n')) \tag{12}$$

for almost every pair (n, n') drawn independently from $\nu(\cdot \mid s_i)$.

At zero expected loss, varying the task-irrelevant component while holding the task-relevant component fixed leaves the representation unchanged.

5.4 Temporal kernels and the successor measure

Having analyzed what information spectral alignment keeps or discards for arbitrary kernels, we now focus on the temporal kernel, and show that the representations it retrieves can be desirable for planning with a world model. This analysis replaces the free embedding tables above with a state-conditioned encoder $h : \mathcal{S} \to \{z \in \mathbb{R}^d : \|z\| = 1\}$. We will show that $\mathcal{L}_{\mathrm{spec}}$ corresponds to a regression problem, whose target can be related to the (behavioral) successor measure (Blier et al., 2021): the discounted cumulative probability of the data generation process to visit a given state in the future. This connects the embedding similarity used in planning to how readily states lead to one another under the behavior represented in the data.

We assume that each trajectory in the dataset is generated by a stationary Markov policy π_β in a Markov decision process with a finite state space $\mathcal{S}$, and that the induced Markov chain P is irreducible and aperiodic, admitting a stationary distribution μ, from which the initial state x_0 is sampled. We can introduce a discount factor derived from the Laplacian kernel and its temperature: $\gamma := e^{-1/\sigma} \in (0, 1)$, and then define the behavioral successor measure as

$$M^\beta(x, x') \;=\; \sum_{t=0}^{\infty} \gamma^t \, \Pr(x_t = x' \mid x_0 = x, \pi_\beta)\,. \tag{13}$$

The training data is generated by drawing trajectories $(x_0, \ldots, x_{T-1})$ from the induced Markov chain, and indices (i, j) uniformly from $\{0, \ldots, T-1\}^2$. For simplicity, we finally assume that the student and the teacher are not centered and that the latter does not undergo tempering: $\alpha = 1$, and no truncation is applied. Let $s(x, x') = h(x)^\top h(x')$ denote embedding similarity. Under these assumptions, we consider the population-level objective:

$$\mathcal{L}^{\mathrm{pop}}_{\mathrm{spec}}(h) \;=\; \mathbb{E}_{\tau,i,j}\Big[\big(s(x_i, x_j) - \gamma^{|i-j|}\big)^2\Big], \tag{14}$$

with $\tau := (x_0, \ldots, x_{T-1}) \sim \mathcal{D}$ and $(i, j) \sim \mathcal{U}(\{0, \ldots, T-1\}^2)$. We also define the expected teacher similarity of the state pair as

$$R_T(x, x') = \mathbb{E}_{\tau,i,j}[\gamma^{|i-j|} | x_i = x, x_j = x']. \tag{15}$$

In Appendix D, we demonstrate that $\mathcal{L}^{\mathrm{pop}}_{\mathrm{spec}}(h)$ is equivalent to minimizing $\mathbb{E}_{\tau,i,j}\big[(s(x_i, x_j) - R_T(x_i, x_j))^2\big]$, with the embedding similarity $s(x_i, x_j) = h(x_i)^\top h(x_j)$. We can then study the target R_T in the limit of long trajectories, and connect it to the successor measure M^β of the behavioral policy π_β.

Theorem 4 (Population target)**.** *Under the setup above, with a finite state space, an irreducible, aperiodic Markov chain P induced by π_β with stationary distribution μ and $x_0 \sim \mu$, for all $x, x' \in \mathcal{S}$, as $T \to \infty$,*

$$T\, R_T(x, x') \;\longrightarrow\; \frac{M^\beta(x, x')}{\mu(x')} + \frac{M^\beta(x', x)}{\mu(x)} - \frac{\mathbf{1}[x = x']}{\mu(x)}. \tag{16}$$

Moreover, if the induced Markov chain is reversible, i.e., $\mu(x)P(x'|x) = \mu(x')P(x|x')$,

$$T\, R_T(x, x') \;\longrightarrow\; \frac{2M^\beta(x, x')}{\mu(x')} - \frac{\mathbf{1}[x = x']}{\mu(x')}. \tag{17}$$

Embedding similarity is thus encouraged to regress a target that tracks a symmetrized likelihood ratio, holding the behavioral successor measure as its numerator. This relates the planning score in Equation 9 to discounted visitation between predicted states and the goal, in both temporal directions and relative to their stationary frequencies.

6 EXPERIMENTS

Our empirical evaluation revolves around three visual manipulation environments: OGBench Cube and Scene (Park et al., 2025) and CALVIN (Mees et al., 2022). Scene and CALVIN extend the evaluation beyond the simpler manipulation settings considered in LeWorldModel (Maes et al., 2026), introducing multiple objects and articulated fixtures whose states must be represented for control. These environments directly test whether the learned representation preserves information about the surrounding scene as well as the robot itself. We primarily compare SpecWM to LeWM (Maes et al., 2026), and we instantiate SpecWM for each of the kernels introduced in Section 4. A generally observed phenomenon in world modeling is that predictors tend to generalize poorly to random actions when trained on only expert data (Fu et al., 2021; Kidambi et al., 2020; Yu et al., 2020). Motivated by this, we generate failure data for each environment and append it to the canonical datasets; this decision is ablated in Appendix B.3.

Our main points of comparison are success rate with CEM planning (Section 6.1) and linear recoverability of physical quantities from embeddings (Section 6.2). Further, we provide an ablation of various kernels in both fully specified and underspecified settings in Section 6.3. We extend these results with additional ablations in Appendix B. For CALVIN, we augment training clips with temporally distant observations and use a multiscale kernel to capture block movements that short clips often miss. Appendix C details this construction, failure data generation, and evaluation settings.

6.1 PLANNING

With temporal and quasimetric kernels, SpecWM improves upon LeWM by an average of 21 percentage points in planning success rate (Figure 3). Intuitively, the planner ranks predicted outcomes by their embedding similarity to the goal, which spectral alignment shapes to reflect temporal or quasimetric proximity. Overall, we observe that a learned quasimetric kernel is generally on par with a readily available temporal kernel, but either may be preferable depending on the environment.

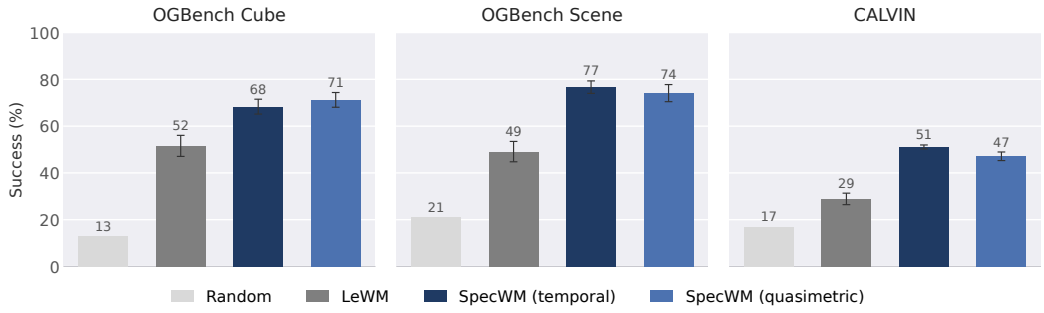

Figure 3: Average CEM planning success rate on 100 fixed tasks per environment. Bars report the mean over the 20k, 24k and 28k checkpoints and 3 seeds, with whiskers showing one cross-seed standard deviation. SpecWM consistently outperforms LeWM.

6.2 PROBING

To isolate representation quality from prediction accuracy, we now evaluate linear probing performance for the position of objects across environments. We consider a held-out set of observations that the encoder was never trained on for each dataset, and define 5 train-evaluation splits. For each split, we fit a ridge regressor for each state variable, choose regularization by cross-validation, and report linear recoverability on the evaluation set. We report R^2 between predicted and ground-truth values, averaged over variables and splits.

As reported in Figure 4, we find that SpecWM does not always beat LeWM on probing, and probe quality does not directly correlate with planning performance.

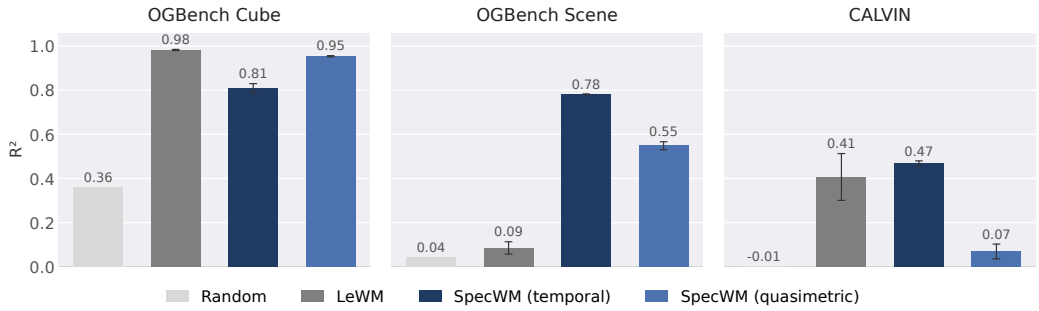

Figure 4: Linear probe R^2 of the manipulated object's position from each model's final checkpoint embedding. Bars report the mean over 3 seeds, with whiskers showing one cross-seed standard deviation. Random is an untrained encoder of the same architecture.

6.3 Kernel Ablations

While the previous evaluations focus on the temporal and quasimetric targets, we now extend the investigation to the full set of kernels we introduced in Section 4, including supervised, fully specified variants (i.e., state, depth, and proprioception), and unsupervised, underspecified variants (i.e., temporal, quasimetric). The latter will help in quantifying the information lost due to lack of ground-truth supervision. Table 1 reports linear probes on all of the fixtures in the Scene environment.

The fully specified state kernel achieves the highest average R^2 (0.93). Under underspecified supervision, the temporal kernel nearly matches the state kernel and outperforms the quasimetric kernel. Proprioceptive kernels preserve the robot's configuration but largely omit object states, supporting Theorem 3. For the fully specified kernels, we note that kernel eigenvectors are indeed strongly recoverable from the learned embeddings (Appendix B.1).

Table 1: Linear probe R^2 on held-out OGBench Scene episodes. Bold marks the best result in each row and results within 0.01 of it.

	Underspecified					Fully specified		
	Temporal	Quasimetric	State	Proprio	Depth	State	Proprio	Depth
Scene (avg)	0.84	0.78	0.85	0.16	0.15	**0.93**	0.16	0.50
`cube`	0.79	0.51	0.57	0.10	0.09	**0.96**	0.08	0.09
`arm`	0.67	0.66	0.69	0.69	0.60	**0.73**	0.67	0.59
`drawer`	0.91	0.90	**0.98**	0.01	0.01	**0.99**	0.01	0.87
`window`	0.87	0.88	**0.99**	0.00	0.01	**0.99**	0.00	0.91
`buttons`	0.96	0.94	**1.00**	0.02	0.01	**0.99**	0.02	0.04

7 Conclusion

We present world modeling through spectral alignment as a framework for designing the spectral methods that have so far been implicit in self-supervised learning. The core idea is to directly specify the pairwise relationships that representations should preserve through a teacher kernel. We explore theoretical connections to fundamental results in spectral manifold learning and, for a specific choice of kernel, to the likelihood of reaching goals expressed in the data distribution. Further, we validate the method empirically and analyze its abstraction properties. We find that readily available kernels result in strong planning performance, which is not entirely correlated with probing accuracy. Pursuing a better understanding of the mechanisms enabling effective planning would be valuable. We view the study of which kernels are most conducive to large-scale self-supervised training as the next primary direction of work. We further view kernels involving language for semantic distances as an interesting future direction.

AI USE STATEMENT

In this work, we used generative AI tools to implement research methods and develop and revise code. We also used these tools to discuss research methodology and theoretical results, identify relevant literature, assist in proving theoretical statements, and draft and edit portions of the manuscript. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure effective reproducibility, all code necessary for this paper is available at (`https://anonymous.4open.science/r/SpecWM/`). Full proofs of presented theorems can be found in Appendix D. The dataset generation process along with evaluation protocols and training settings are described in Appendix C as well as available in the code.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, June 2023.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=SzXDuBN8M1`.

Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In *Advances in Neural Information Processing Systems*, volume 35 of *NIPS '22*, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

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

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

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. doi: 10.1162/089976603321780317.

Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, 16(10):2197–2219, 2004. doi: 10.1162/0899766041732396.

Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint, 2021. URL `https://arxiv.org/abs/2101.07123`.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz Kandola. On kernel-target alignment. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL `https://proceedings.neurips.cc/paper_files/paper/2001/file/1f71e393b3809197ed66df836fe833e5-Paper.pdf`.

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, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=vGQiU5sqUe3.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2021. URL https://arxiv.org/abs/2004.07219.

Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11321–11339. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ghosh23a.html.

David Ha and Jürgen Schmidhuber. World models, 2018. URL https://zenodo.org/record/1207631.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models, 2025. URL https://arxiv.org/abs/2509.24527.

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

Jeff Z. HaoChen and Tengyu Ma. A theoretical study of inductive biases in contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=AuEgNlEAmed.

Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

Michael Kampffmeyer, Sigurd Løkse, Filippo M. Bianchi, Robert Jenssen, and Lorenzo Livi. The deep kernelized autoencoder. *Applied Soft Computing*, 71:816–825, 2018. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2018.07.029. URL https://www.sciencedirect.com/science/article/pii/S1568494618304174.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21810–21823. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f7efa4f864ae9b88d43527f4b14f750f-Paper.pdf.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

Misha Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/091d584fced301b442654dd8c23b3fc9-Paper-round2.pdf.

Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FkeURAdA0h.

Yann LeCun. A path towards autonomous machine intelligence. Open Review, June 2022. URL https://openreview.net/forum?id=BZ5a1r-kVsf. Version 0.9.2.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal World Modeling for Robot Control. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.016.

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

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, and Nicolas Ballas. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. In *Computer Vision – ECCV 2026*, volume 17005 of *Lecture Notes in Computer Science*, pp. 671–689, Cham, 2026. Springer Nature Switzerland. doi: 10.1007/978-3-032-37369-4_36. URL https://doi.org/10.1007/978-3-032-37369-4_36.

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

Jack Parker-Holder and Shlomi Fruchter. Genie 3: A new frontier for world models. Google DeepMind Blog, 2025. URL https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/.

Reuven Y. 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.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-03051-4. URL http://dx.doi.org/10.1038/s41586-020-03051-4.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Remi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, András György, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, and Michal Valko. Understanding self-predictive learning for reinforcement learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett

(eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33632–33656. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/tang23d.html.

Team HY-World. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026. doi: 10.48550/arXiv.2604.14268. URL https://arxiv.org/abs/2604.14268.

Stephen Tian, Suraj Nair, Frederik Ebert, Sudeep Dasari, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Model-based visual planning with self-supervised functional distances. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=UcoXdfrORC.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=2a96Bf7Qdrg.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Tongzhou Wang and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. URL https://openreview.net/forum?id=KRiST_rzkGl.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *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.

World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, 2026. https://www.worldlabs.ai/blog/atlas.

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

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14129–14142. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a322852ce0df73e204b7e67cbbef0d0a-Paper.pdf.

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

Amy Zhang, Rowan Thomas 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.

Chongyi Zheng, Royina Karegoudra Jayanth, and Benjamin Eysenbach. Can we really learn one representation to optimize all rewards? In *Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning*, 2026. URL https://openreview.net/forum?id=ClWj7AfhcY.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

A Extended Related Work

Kernel alignment, the normalized Frobenius inner product between a candidate kernel and a target similarity matrix, was introduced by Cristianini et al. (2001) as a criterion for kernel selection. Kampffmeyer et al. (2018) later introduced alignment of an autoencoder's embedding Gram matrix to a provided kernel. We extend this paradigm by removing the reconstruction term and use action conditioned prediction as an additional shaping force for the latent. Their unsupervised kernel derives similarity from multiscale clustering of input features, which need not reflect the dynamical relationships relevant to control. We instead study both supervised and unsupervised kernels with a more direct relevance for control and world modeling. Furthermore, our trajectory-local formulation leaves cross-trajectory relationships unspecified, allowing flexibility in the learned geometry.

Self-Supervised Representations in RL have been explored both as auxiliary objectives in model-free reinforcement learning (Schwarzer et al., 2021), and as part of latent model-based approaches (Schrittwieser et al., 2020; Hansen et al., 2024). A further line of work has focused on representation learning as a central component of practical methods, for instance enabling generalization in behavior cloning (Lawson et al., 2026), or robustness in value-based learning (Zhang et al., 2021). Self-supervised representations are also a core mechanism in unsupervised (Laskin et al., 2021) and zero-shot reinforcement learning (Touati et al., 2023). In particular, contrastive RL (Eysenbach et al., 2022), forward-backward representation (Touati & Ollivier, 2021; Zheng et al., 2026), ICVF (Ghosh et al., 2023) and TD-JEPA (Bagatella et al., 2026) recover representations which can be connected to the successor measure; we highlight a similar connection for spectral alignment with a temporal kernel. Finally, this class of kernels shares the motivation of early works on temporal distance regression (Tian et al., 2021), which however do not extract representations explicitly.

B Additional Experiments

B.1 Spectral recovery

Theorem 1 characterizes optimal representations as row-normalized spectral embeddings of the teacher kernel, up to an orthogonal transformation. To examine this connection empirically, we measure how well linear probes recover individual kernel eigenvectors from the learned representations (Figure 5). We observe that recovery is strongest for the leading eigenvectors and decreases for those associated with smaller eigenvalues.

The lower correlations observed in the state kernel are largely the result of a less even distribution of eigenvalue size, concentrating most of the energy in the lower eigenvector indices. In terms of mean R^2 weighted by each eigenvector's normalized eigenvalue, state gets a 0.965, proprio a 0.987, and depth a 0.992.

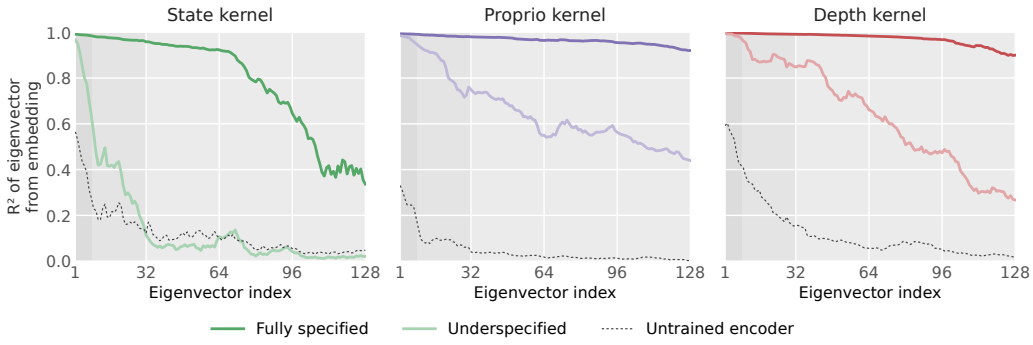

Figure 5: Linear probe R^2 between eigenvectors and embeddings from encoders trained on fully specified state, proprioception, and depth. Underspecified variants and an untrained ViT are included for comparison.

B.2 PREDICTION GRADIENTS AND DOWNSTREAM PERFORMANCE

In Figure 6, we additionally evaluate a variant of SpecWM which does not backpropagate the predictor regression objective through the target (i.e., the next state representations):

$$\mathcal{L} = \mathcal{L}_{\text{spec}} - \lambda \sum_i \big(\cos(G, \text{sg}[H'])\big)_i \tag{18}$$

The prediction loss still updates the encoder through the current observation, but no longer through the next observation used as its target. We observe that planning success generally decreases, while probing results do not follow an obvious trend.

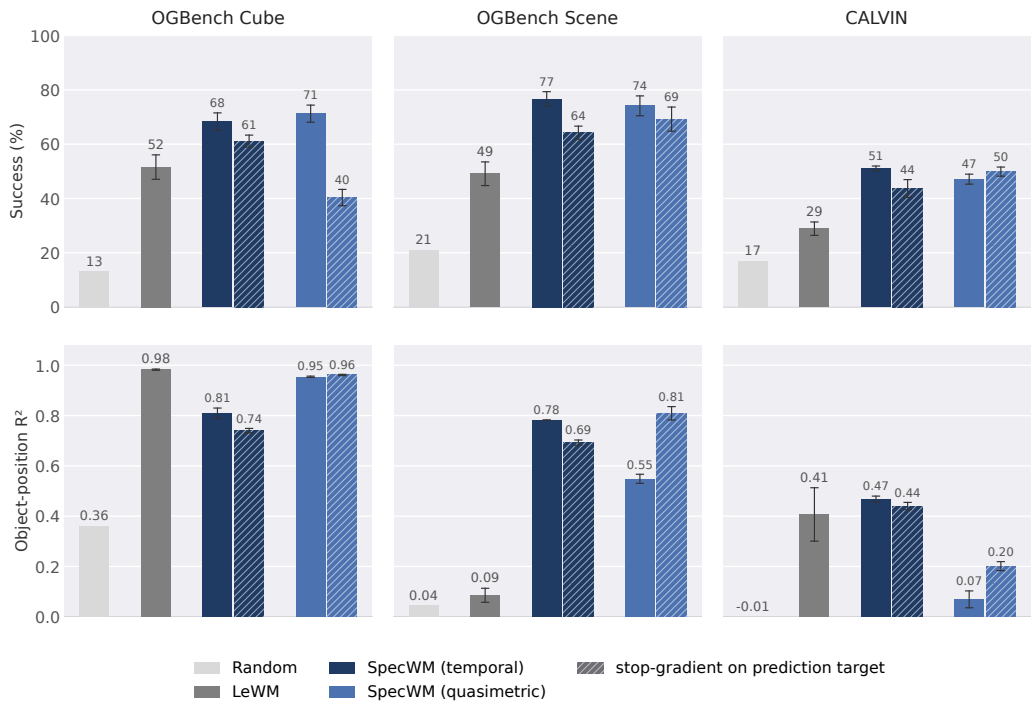

Figure 6: Effect of a stop-gradient on the prediction target. Bars report the mean over 3 seeds, with whiskers showing one cross-seed standard deviation. The stop-gradient generally lowers planning success and raises object recoverability in some cases while diminishing recoverability in others.

B.3 DATA ABLATIONS

It is known that predictors tend to generalize poorly to random actions when trained on expert-only data (Fu et al., 2021; Kidambi et al., 2020; Yu et al., 2020). Here we ablate the failure data (Appendix C.5) by training LeWM and SpecWM with the temporal kernel on the play data alone, at matched updates, frames per update and schedule. As Figure 7 shows, the failure data raises planning success for both models in every environment, but does not explain the difference between them.

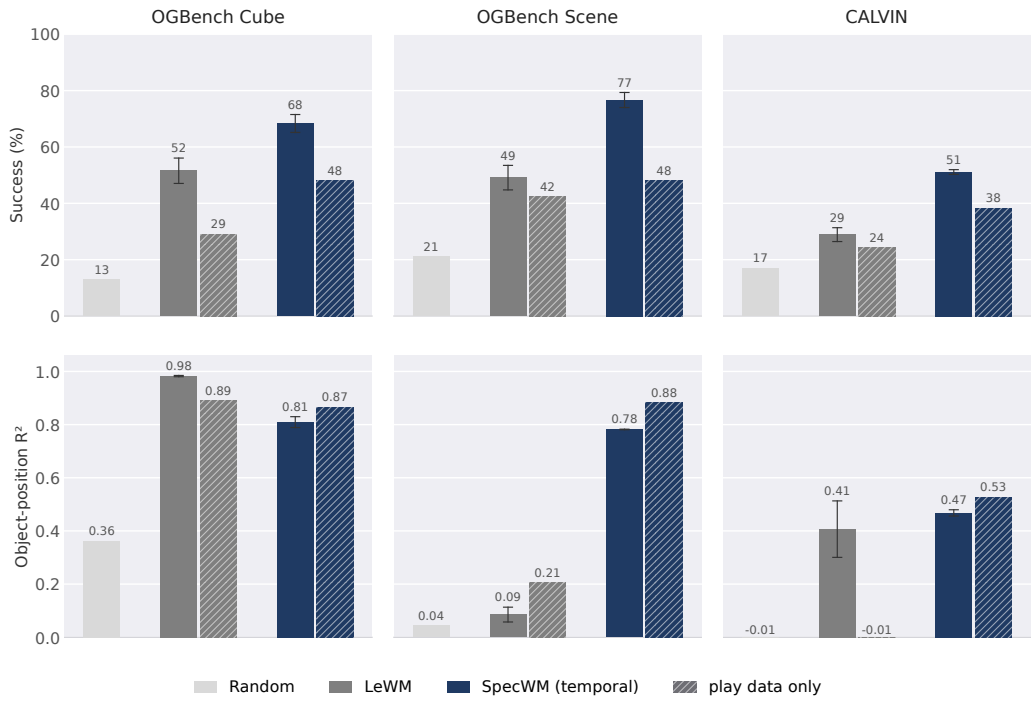

Figure 7: Effect of the failure data. CEM success (top) and object position probe R^2 (bottom) for LeWM and SpecWM (temporal), trained with the failure rollouts (solid, 3 seeds) or on the play data alone (hatched, 1 seed), at matched updates and frames per update. Failure data raises planning success for both models; SpecWM outperforms LeWM in both cases.

B.4 CEM COST ABLATION

The two compared world models have different inherent distance metrics between samples in their latent distribution. For the purposes of the main paper, we use the one that fits each latent most naturally (L2 for LeWM and cosine similarity for SpecWM). Figure 8 demonstrates that the choice of measure is not the cause of the planning gains.

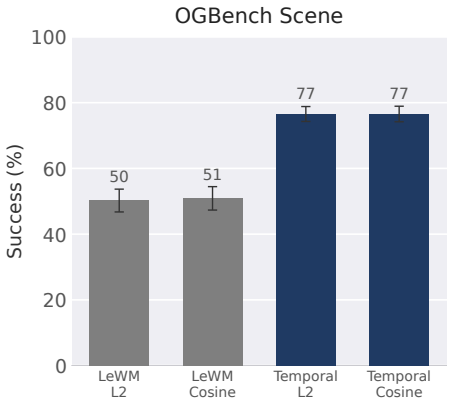

Figure 8: Effect of the choice of distance measure on CEM planning results. Cosine similarity and L2 distance between current embedding and target embedding are evaluated on both LeWM and SpecWM. The choice of distance measure appears to not be a major contributor to planning ability. Three training seeds are evaluated at their respective final three checkpoints for each setting.

B.5 NORMALIZATION LAYER ABLATION

To be faithful to the original LeWorldModel (Maes et al., 2026) implementation, we use BatchNorm (Ioffe & Szegedy, 2015) in the MLP producing the embedding from the ViT (Dosovitskiy et al., 2021) encoder's CLS token. As SpecWM does not require BatchNorm to effectively optimize its objective, we use a LayerNorm for simplicity. As demonstrated in Figure 9, the choice of norm does not have a substantial impact on planning performance.

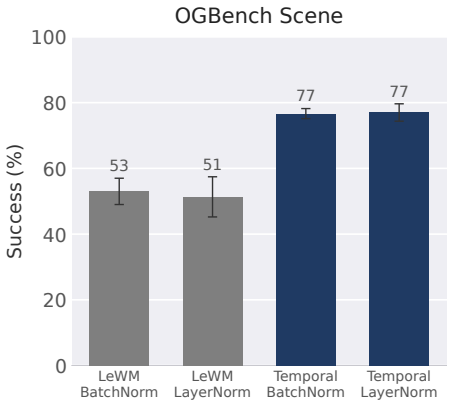

Figure 9: Effect of choice of normalization layer in the MLP converting the CLS token into an embedding on downstream CEM planning. One seed is used per run and CEM is reported as the average of the final three checkpoints of each run.

C REPRODUCIBILITY

This appendix contains the necessary details to reproduce the results of the paper. For code, see: `https://anonymous.4open.science/r/SpecWM/`.

C.1 PROBING

To measure to what extent the embeddings contain control-relevant objects, we perform linear probing of position coordinates. We sample 5,000 frames from the validation episodes, which the encoder never trained on, and split full episodes into training and testing sets of 80% and 20% respectively, repeated over five splits. Linear readouts use ridge regression, with all preprocessing and regularization selection performed using training episodes only. Hyperparameters are selected using episode-grouped subsets of the training split. Ridge coefficients are selected from a logarithmic grid of 13 values between 10^{-2} and 10^{4}. We then report held-out R^2 of the best ridge coefficient, using the full 80% to fit the linear probe, averaged over the five splits.

C.2 LONG-RANGE PAIRS ON CALVIN

In CALVIN, blocks move only when manipulated, so their positions often remain constant within a short clip. Within-clip temporal supervision therefore provides little signal to distinguish block positions. We augment each clip with 8 observations from the same episode, sampled 300 to 3,000 raw frames away (roughly 20 seconds to 3 minutes), to capture changes over longer periods. These observations contribute only to the spectral loss, using their original time indices, and require no associated actions.

To capture both short and long separations, we average two centered, trace-normalized Laplacian kernels with bandwidths $\sigma_1 = 7$ and $\sigma_2 = 45$, measured in sampled observations. We use the same construction for the quasimetric kernel:

$$K_{ij} = \tfrac{1}{2}\Big(e^{-|i-j|/\sigma_1} + e^{-|i-j|/\sigma_2}\Big). \tag{19}$$

Scene and Cube clips already capture the relevant changes, so we retain a single-scale kernel.

C.3 PLANNING VIA CEM

CEM hyperparameters are detailed in Table 2. The base action sampling distribution is a Gaussian with starting mean and variance set according to expert data statistics. SpecWM cost-to-go is evaluated using cosine similarity and LeWM uses L2 distance.

Table 2: CEM planning configuration.

Parameter	**Scene**	**Cube**	**CALVIN**
Candidate sequences	300	300	300
CEM iterations	30	30	30
Elite sequences	30	30	30
Initial variance scale	1.0	1.0	1.0
Planning horizon (blocks)	5	5	5
Environment steps per block	5	5	10
Steps executed before replanning	15	15	30
Number of evaluation tasks	100	100	100
Goal offset (steps)	25	25	50
Evaluation budget (steps)	60	60	100

C.4 TRAINING

Architecture. The JEPA architecture follows that of LeWorldModel and is consistent across objectives aside from normalization scheme. LeWorldModel uses BatchNorm in its projection head in line with Maes et al. (2026), whereas SpecWM uses LayerNorm. Their remaining shared architectural settings are summarized in Table 3.

Table 3: Shared visual-encoder and action-conditioned dynamics architecture.

Component	Configuration
Visual encoder	Randomly initialized ViT-T/14 with output dimension 192
Projection head	MLP $192 \to 2048 \to 192$
Action encoder	1×1 Conv1D followed by an MLP with hidden dimension 768 and output dimension 192
Dynamics predictor	6-layer action-conditioned transformer
Attention	16 heads with head dimension 64
Transformer MLP dimension	2048
Dropout	0.1
Observation-history length	3 sampled observations

Table 4: Environment-dependent action representation. Each action block contains all low-level actions between two consecutive sampled observations.

Environment	Frame skip	Raw action dim.	Action-block dim.
OGBench Scene	5	5	25
OGBench Cube	5	5	25
CALVIN	10	7	70

Optimization and training. Optimization hyperparameters are detailed in Table 5. The unusual learning-rate values result from square-root batch-size scaling of base rates 5×10^{-5} for AdamW and 10^{-3} for Muon. For CALVIN, SpecWM additionally encodes 8 distant observations per clip for the spectral loss, giving $24 \times (23 + 8) = 744$ total images per update. LeWorldModel uses 552 images per update in all environments.

Quasimetric training. We train the quasimetric teachers using a separate ViT-T/14 followed by a projection and an IQE head. The IQE head's 384-dimensional output is divided into 24 IQE groups of dimension 16, producing an asymmetric directed distance. The training objective expands distances between distinct observations while constraining consecutive observations (Wang et al., 2023):

$$\mathcal{L}_{\text{push}} = -\mathbb{E}_{i \neq j}[d(z_i, z_j)]\,,$$
$$\mathcal{L}_{\text{step}} = \mathbb{E}_t\Big[\max(0, d(z_t, z_{t+1}) - 1)^2\Big]\,,$$
$$\mathcal{L}_{\text{QRL}} = \mathcal{L}_{\text{push}} + \lambda\left(\mathcal{L}_{\text{step}} - \epsilon^2\right).$$

The dual variable is initialized to 1 and updated with learning rate 0.05 toward a constraint value of $\epsilon = 0.25$, while being clipped to $[0, 50]$. Teachers are trained for 9,500 updates using 552 frames per update (24 clips of 23 frames).

C.5 Data

For all environments, we supplement the official datasets with failure data to aid in predictor training using noised actions. For each episode or segment i of length L_i, we sample an onset

$$t_i \sim \text{Uniform}\left(\{\lfloor 0.05L_i \rfloor, \ldots, \lfloor 0.90L_i \rfloor\}\right)$$

and alternate the noise scale between $\sigma_i = 0.3$ and $\sigma_i = 0.6$. The replayed action is

$$\tilde{a}_t = \begin{cases} a_t, & t < t_i, \\ \text{clip}(a_t + \sigma_i \epsilon_t, -1, 1), & t \geq t_i, \end{cases} \qquad \epsilon_t \sim \mathcal{N}(0, I).$$

Each replay is initialized from its recorded simulator state and follows the recorded open-loop actions exactly before the noise onset. For Scene and Cube, noise is applied to all five action dimensions. For CALVIN, noise is applied only to the six continuous motion dimensions, while the binary gripper command is retained from the recorded trajectory. The resulting Scene and Cube datasets each contain 1,000,000 frames from the official dataset and 1,000,000 frames from our failure rollouts. The CALVIN dataset contains 1,795,045 official frames and 1,793,322 failure frames.

Table 5: Optimization and student-training settings.

Setting	Configuration
Peak learning rates	1.461×10^{-3} for matrices, 2.922×10^{-4} for convolutions, and 7.30×10^{-5} for biases and normalization parameters
MuonAdamW	$(\beta_1,\beta_2)=(0.9,0.999)$, $\epsilon=10^{-8}$, Muon momentum 0.95, Nesterov updates, and RMS matching
Regularization	Weight decay 10^{-3} on weights; no decay on biases or normalization parameters; gradient clipping at 1.0
SpecWM	33,375 updates; 552 base clip frames per update (24×23); $\lambda_{\text{spec}}=1$ for Cube and Scene and 2 for CALVIN; cosine prediction-anchor weight 2.
LeWorldModel	33,375 updates; 552 frames per update (138×4); prediction-MSE weight 1 and SIGReg weight 0.09

D Full Proofs

The proofs below follow the order of the stated theorems in Section 5.

D.1 Proof of Theorem 1

Let $H\in\mathbb{R}^{n\times d}$ be a free embedding table with nonzero rows $h_1,\dots,h_n$, and let K^α be the centered, tempered target. The loss rescales each row to the length the target prescribes, forms the Gram matrix, and compares it to the target after double centering:

$$\bar h_i=\sqrt{K^\alpha_{ii}}\,\frac{h_i}{\|h_i\|},\qquad S=\bar H\bar H^\top,\qquad \mathcal{L}_{\text{spec}}(H)=\|CSC-K^\alpha\|_F^2\,, \tag{20}$$

with $C=I-\frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Note that the rescaling pins the diagonal of S to that of the target,

$$S_{ii}=\left\|\bar h_i\right\|^2=K^\alpha_{ii}, \tag{21}$$

for every H.

Let $\tilde K=CKC$ have eigenvalues $\lambda_1\ge\cdots\ge\lambda_n$ with unit eigenvectors $u_1,\dots,u_n$, and let $m\ge1$ be the number of directions the tempering keeps (the minimum of the count of positive λ_ℓ and the truncation count). Since the tempered spectrum vanishes outside these m directions, the target becomes

$$K^\alpha=\sum_{\ell=1}^{m}s_\ell^2\,u_\ell u_\ell^\top,\qquad s_\ell=\left(\frac{n\lambda_\ell^\alpha}{\sum_{j\le m}\lambda_j^\alpha}\right)^{1/2}. \tag{22}$$

Three properties of the target are used below:

$$K^\alpha\succeq0,\qquad \operatorname{rank}K^\alpha=m,\qquad K^\alpha\mathbf{1}=0. \tag{23}$$

The first two come from equation 22, whose m terms have orthogonal directions u_ℓ and positive weights s_ℓ^2. For the third, $C\mathbf{1}=0$ gives $\tilde K\mathbf{1}=0$, and every u_ℓ in equation 22 has $\lambda_\ell>0$, so each is orthogonal to $\mathbf{1}$. As assumed in Theorem 1, the target has strictly positive diagonal entries.

Lemma 1.1. *$\mathcal{L}_{spec}(H)=0$ if and only if $S=K^\alpha$.*

Proof. If $S=K^\alpha$ the loss is $\|CK^\alpha C-K^\alpha\|_F^2$. Since K^α is symmetric and $K^\alpha\mathbf{1}=0$, we have $\mathbf{1}^\top K^\alpha=0$. Thus, $CK^\alpha=K^\alpha-\frac{1}{n}\mathbf{1}(K^\alpha\mathbf{1})^\top=K^\alpha$ and likewise $K^\alpha C=K^\alpha-\frac{1}{n}(K^\alpha\mathbf{1})\mathbf{1}^\top=K^\alpha$. Hence $CK^\alpha C=K^\alpha$ and the loss is 0.

Conversely, suppose $CSC = K^\alpha$. Centering is not injective, so this alone does not give $S = K^\alpha$. Accordingly, we use the diagonal constraint equation 21. Since C is symmetric and idempotent,

$$\begin{aligned}\operatorname{tr} K^\alpha &= \operatorname{tr}(CSC)\\ &= \operatorname{tr}(SC^2)\\ &= \operatorname{tr}(SC)\\ &= \operatorname{tr} S - \tfrac{1}{n}\,\mathbf{1}^\top S\mathbf{1}.\end{aligned}$$

Since $\operatorname{tr} S = \operatorname{tr} K^\alpha$ by equation 21, we have that $\mathbf{1}^\top S\mathbf{1} = 0$. Expanding, we see that $\mathbf{1}^\top S\mathbf{1} = \mathbf{1}^\top \bar{H}\bar{H}^\top\mathbf{1} = \left\|\bar{H}^\top\mathbf{1}\right\|^2 = 0$, which forces $\bar{H}^\top\mathbf{1} = 0$. Thus, we have that $\bar{H}\bar{H}^\top\mathbf{1} = S\mathbf{1} = 0$. Since S is symmetric, $S\mathbf{1} = 0$ also gives $\mathbf{1}^\top S = 0$. Hence $CS = SC = S$, and therefore $S = CSC = K^\alpha$. □

Lemma 1.2. *Assume $m \le d$. Let $U_m = [u_1, \dots, u_m]$ contain the retained orthonormal eigenvectors of the centered kernel $\widetilde{K} = CKC$. Define $H_{\text{spec}} \in \mathbb{R}^{n\times d}$ by assigning sample i the row*

$$(H_{\text{spec}})_{i,:} = \big(s_1u_1(i), \dots, s_mu_m(i), \underbrace{0, \dots, 0}_{d-m}\big),$$

where $u_\ell(i)$ is the i-th entry of eigenvector u_ℓ and the scales s_ℓ are given by equation 22. Then $S = K^\alpha$ if and only if $H = \operatorname{diag}(c)H_{\text{spec}}Q$ for some $c \in \mathbb{R}^n_{>0}$ and orthogonal $Q \in \mathbb{R}^{d\times d}$.

Proof. First, we will establish that $H_{\text{spec}}H_{\text{spec}}^\top = K^\alpha$. By equation 2 we have that $H_{\text{spec}} = U_d \operatorname{diag}(s_1, \dots, s_d)$. Thus,

$$\begin{aligned}H_{\text{spec}}H_{\text{spec}}^\top &= U_m \operatorname{diag}(s_1^2, \dots, s_m^2)\, U_m^\top\\ &= \sum_{\ell=1}^{m} s_\ell^2\, u_\ell u_\ell^\top\\ &= K^\alpha,\end{aligned}$$

where the last equality is equation 22. The diagonal of a Gram matrix lists the squared row norms, so $\|(H_{\text{spec}})_i\|^2 = K^\alpha_{ii}$. Second, we use a standard fact about Gram matrices: if $BB^\top = B'B'^\top$ for $B, B' \in \mathbb{R}^{n\times d}$, then $B = B'Q$ for some orthogonal $Q \in \mathbb{R}^{d\times d}$.

Suppose $S = K^\alpha$. Then $\bar{H}\bar{H}^\top = H_{\text{spec}}H_{\text{spec}}^\top$, so $\bar{H} = H_{\text{spec}}Q$ for some orthogonal Q. By equation 20, $h_i = c_i\bar{h}_i$ with $c_i = \|h_i\| / \sqrt{K^\alpha_{ii}} > 0$, so $H = \operatorname{diag}(c)\,\bar{H} = \operatorname{diag}(c)\,H_{\text{spec}}Q$.

Suppose instead $H = \operatorname{diag}(c)\,H_{\text{spec}}Q$. Since Q preserves lengths, row i of $H_{\text{spec}}Q$ has the same length as row i of H_{spec}, namely $\sqrt{K^\alpha_{ii}} > 0$. The rows of H are therefore nonzero, and rescaling them to length $\sqrt{K^\alpha_{ii}}$ removes the factor c_i, so $\bar{H} = H_{\text{spec}}Q$. Hence $S = \bar{H}\bar{H}^\top = H_{\text{spec}}QQ^\top H_{\text{spec}}^\top = H_{\text{spec}}H_{\text{spec}}^\top = K^\alpha$. □

Theorem 1 (Spectral recovery). *Suppose $m \le d$. Then $\min_H \mathcal{L}_{spec}(H) = 0$, attained exactly at*

$$H^\star = \operatorname{diag}(c)\, H_{spec}\, Q, \tag{24}$$

with $c \in \mathbb{R}^n_{>0}$, $Q \in \mathbb{R}^{d\times d}$ orthogonal, and H_{spec} the truncated spectral embedding of equation 2.

Proof of Theorem 1. By Lemma 1.1, $\mathcal{L}_{\text{spec}}(H) = 0$ if and only if $S = K^\alpha$, and by Lemma 1.2 this holds if and only if H has the form equation 10. The set of such H is nonempty, since it contains H_{spec} itself ($c = \mathbf{1}$, $Q = I$), so the minimum is attained and equals 0. □

D.2 Proof of Theorem 2

We next connect this embedding geometry to linear probing performance. In particular, a kernel constructed from label relationships can make the labels themselves linearly recoverable from the embeddings, without directly regressing encoder outputs onto them. We formalize this below for row-normalized and centered vector-valued labels Y.

Theorem 2 (Linear recoverability). *Let $Y \in \mathbb{R}^{n\times p}$ satisfy $\mathbf{1}^\top Y = 0$, row norms $\|y_i\| = 1$, and $\operatorname{rank}(Y) \le d$. For $K = YY^\top$, $\alpha = 1$, and all positive spectral directions retained, every global minimizer $H^\star$ of $\mathcal{L}_{spec}$ subject to $\|h_i\| = \sqrt{d}$ satisfies*

$$\min_{W\in\mathbb{R}^{d\times p}} \|Y - H^\star W\|_F^2 = 0. \tag{25}$$

Proof of Theorem 2. We will construct W that satisfies 25. Since Y is centered and its rows have unit norm, we have that $\tilde{K} = CYY^TC = YY^T$ with $\operatorname{tr}(\tilde{K}) = n$. With $\alpha = 1$ and all positive directions retained, the target is therefore $K^\alpha = YY^T$, with $K^\alpha_{ii} = 1$. Write the compact SVD of Y and YY^T as

$$Y = U_m\Sigma_m V^T, \qquad YY^T = U_m\Sigma_m^2 U_m^T.$$

To pad the m-dimensional spectral embeddings to the encoder's d dimensions, let $E_m = [I_m \;\; 0] \in \mathbb{R}^{m\times d}$. The spectral embedding is

$$H_{\text{spec}} = U_m\Sigma_m E_m.$$

The rows of H_{spec} have unit norm because $H_{\text{spec}}H_{\text{spec}}^\top = K^\alpha$ and $K^\alpha_{ii} = 1$. Multiplying by $\sqrt{d}$ makes each row satisfy the required norm $\sqrt{d}$, while preserving zero loss by Theorem 1. Since the loss is nonnegative, every global minimizer under this row-norm constraint must therefore also achieve zero loss. Further by Theorem 1, each such minimizer has the form

$$H^\star = \operatorname{diag}(c)\, H_{\text{spec}}Q.$$

Since Q preserves lengths, the RMS constraint forces $c_i = \sqrt{d}$ for every i. Hence, we have that

$$H^\star = \sqrt{d}U_m\Sigma_m E_m Q.$$

We can now define $W^\star$:

$$W^\star = \frac{1}{\sqrt{d}} Q^T E_m^T V^T. \tag{26}$$

Using $QQ^T = I_d$ and $E_mE_m^T = I_m$, we have that

$$\begin{aligned} H^\star W^\star &= U_m\Sigma_m E_m QQ^T E_m^T V^T \\ &= U_m\Sigma_m V^T \\ &= Y. \end{aligned}$$

Therefore $\min_W \|Y - H^\star W\|_F^2 = 0$. □

D.3 Proof of Theorem 3

Theorem 3 (Abstraction of task-irrelevant information). *Under the coverage assumption, fixed-norm encoder outputs, and the target-diagonal rescaling in Equation 6, if each trajectory's target matrix K^α has strictly positive diagonal entries almost surely and $\mathbb{E}[\mathcal{L}_{\text{spec}}] = 0$, then for almost every clip and every frame i,*

$$f_\theta((s_i, n)) = f_\theta((s_i, n'))$$

for almost every pair (n, n') drawn independently from $\nu(\cdot \mid s_i)$.

Proof of Theorem 3. We will show that coverage allows independent nuisance draws while preserving the target inner products. Centering then forces their rescaled embeddings to sum to zero, so changing one nuisance draw cannot change its embedding.

Since the loss is nonnegative, $\mathbb{E}\,\mathcal{L}_{\text{spec}} = 0$ implies zero loss almost surely. Lemma 1.1 therefore gives $S = K^\alpha$ almost surely. Fix a task-relevant trajectory and teacher metadata Z for which this equality holds with conditional probability one and the coverage assumption 5.3 holds. Let q be the number of frames. For this fixed Z, define

$$b_i(n) = \sqrt{\frac{K^\alpha_{ii}}{d}}\, f_\theta((s_i, n)).$$

These are the rescaled embeddings used in the spectral loss. Since the encoder outputs have norm $\sqrt{d}$, we have $\|b_i(n)\|^2 = K^\alpha_{ii}$. Now draw auxiliary nuisance values $\widehat{n}_1, \ldots, \widehat{n}_q$ independently, with $\widehat{n}_i \sim \nu(\cdot \mid s_i)$. For each $i \neq j$, coverage implies

$$b_i(\widehat{n}_i)^\top b_j(\widehat{n}_j) = K^\alpha_{ij} \qquad \text{almost surely.}$$

If the set of pairs violating this equality had positive probability under the independent draws, assumption (A1) would assign it positive conditional probability in the trajectory, causing non-zero loss. For diagonal entries, we also have $b_i(\widehat{n}_i)^\top b_i(\widehat{n}_i) = K^\alpha_{ii}$ by the definition of b_i. Together with the off-diagonal equalities above, this gives

$$b_i(\widehat{n}_i)^\top b_j(\widehat{n}_j) = K^\alpha_{ij} \qquad \text{for all } i, j \quad \text{almost surely.}$$

Using this equality together with the fact that K^α is centered,

$$\left\| \sum_{j=1}^{q} b_j(\widehat{n}_j) \right\|^2 = \sum_{j,k=1}^{q} b_j(\widehat{n}_j)^\top b_k(\widehat{n}_k) = \sum_{j,k=1}^{q} K^\alpha_{jk} = \mathbf{1}^\top K^\alpha \mathbf{1} = 0 \qquad \text{almost surely.}$$

Fix a frame i and replace only $\widehat{n}_i$ by an independent draw $\widehat{n}'_i \sim \nu(\cdot \mid s_i)$. The new tuple has the same distribution, so both sums below vanish almost surely:

$$b_i(\widehat{n}_i) + \sum_{j \neq i} b_j(\widehat{n}_j) = 0, \qquad b_i(\widehat{n}'_i) + \sum_{j \neq i} b_j(\widehat{n}_j) = 0.$$

Subtracting gives $b_i(\widehat{n}_i) = b_i(\widehat{n}'_i)$ almost surely. Since $K^\alpha_{ii} > 0$, the rescaling can be cancelled, yielding

$$f_\theta((s_i, \widehat{n}_i)) = f_\theta((s_i, \widehat{n}'_i)) \qquad \text{almost surely.}$$

The two nuisance draws are independent with distribution $\nu(\cdot \mid s_i)$, which is the claimed invariance. The argument applies to every frame and almost every Z. □

D.4 Proof of Theorem 4

We resume from the assumptions stated in Section 5.4, from the population-level variant of the spectral alignment objective

$$\mathcal{L}^{\text{pop}}_{\text{spec}}(h) \;=\; \mathbb{E}_{\tau,i,j}\Big[\big(s(x_i, x_j) - \gamma^{|i-j|}\big)^2\Big], \tag{27}$$

with $\tau := (x_0, \ldots, x_{T-1}) \sim \mathcal{D}$ and $(i, j) \sim \mathcal{U}(\{0, \ldots, T-1\}^2)$, and from the expected teacher similarity of the state pair

$$R_T(x, x') = \mathbb{E}_{\tau,i,j}[\gamma^{|i-j|} | x_i = x, x_j = x']. \tag{28}$$

We can then apply a bias-variance decomposition and rewrite the loss as:

$$\begin{aligned}
\mathcal{L}^{\text{pop}}_{\text{spec}}(h) &= \mathbb{E}_{\tau,i,j}\Big[(s(x_i, x_j) - \gamma^{|i-j|})^2\Big] && (29)\\
&= \mathbb{E}_{\tau,i,j}\Big[\big((s(x_i, x_j) - R_T(x_i, x_j)\big) + \big(R_T(x_i, x_j) - \gamma^{|i-j|}\big)\big)^2\Big] && (30)\\
&= \mathbb{E}_{x,x'}\Bigg[\mathbb{E}_{(\tau,i,j)\sim p(\cdot|x_i=x,\, x_j=x')}\Big[\big(s(x_i, x_j) - R_T(x_i, x_j)\big)^2\Big] && (31)\\
&\quad + \mathbb{E}_{(\tau,i,j)\sim p(\cdot|x_i=x,\, x_j=x')}\Big[\big(R_T(x_i, x_j) - \gamma^{|i-j|}\big)^2\Big] && (32)\\
&\quad + 2\mathbb{E}_{(\tau,i,j)\sim p(\cdot|x_i=x,\, x_j=x')}\Big[\big(s(x_i, x_j) - R_T(x_i, x_j)\big)\big(R_T(x_i, x_j) - \gamma^{|i-j|}\big)\Big]\Bigg], && (33)
\end{aligned}$$

where $\mathbb{E}_{x,x'}$ is over pairs of observations sampled from the dataset. The second term is not dependent on h and the third term fades as $(s(x_i, x_j) - R_T(x_i, x_j))$ is constant upon conditioning on $(x_i = x, x_j = x')$, and $\mathbb{E}_{(\tau,i,j)\sim p(\cdot|x_i=x,\, x_j=x')}[R_T(x_i, x_j) - \gamma^{|i-j|}] = 0$; thus minimizing $\mathcal{L}^{\text{pop}}_{\text{spec}}(h)$ is equivalent to minimizing $\mathbb{E}_{\tau,i,j}\big[(s(x_i, x_j) - R_T(x_i, x_j))^2\big]$. We will now study the target R_T in the limit of long trajectories, and connect it to the successor measure M^β of the behavioral policy π_β.

Theorem 4 (Population target). *Under the setup above, with a finite state space, an irreducible, aperiodic Markov chain P induced by π_β with stationary distribution μ and $x_0 \sim \mu$, for all $x, x' \in \mathcal{S}$, as $T \to \infty$,*

$$T\, R_T(x, x') \;\longrightarrow\; \frac{M^\beta(x, x')}{\mu(x')} + \frac{M^\beta(x', x)}{\mu(x)} - \frac{\mathbf{1}[x = x']}{\mu(x)}.$$

Moreover, if the induced Markov chain is reversible, i.e., $\mu(x)P(x'|x) = \mu(x')P(x|x')$,

$$T\, R_T(x, x') \;\longrightarrow\; \frac{2M^\beta(x, x')}{\mu(x')} - \frac{\mathbf{1}[x = x']}{\mu(x')}.$$

Proof of Theorem 4. By definition, $TR_T(x, x') = \frac{N(x,x')}{D(x,x')}$ with

$$N(x, x') := T\, \mathbb{E}_{\tau,i,j}\big[\gamma^{|i-j|}\, \mathbf{1}[x_i = x, x_j = x']\big], \quad D(x, x') := E_{\tau,i,j}\big[\mathbf{1}[x_i = x, x_j = x']\big]. \tag{34}$$

We will study the numerator and the denominator separately.

Numerator. Let us consider the index difference $k := j - i$, which is distributed as $p(k) = (T - |k|)/T^2$ for $|k| \le T - 1$, and denote the probability that $x_i = x$ and $x_j = x'$ as

$$q_k(x, x') := \Pr(x_i = x, x_j = x' \mid j - i = k) = \begin{cases} \mu(x)P^k(x, x'), & k \ge 0, \\ \mu(x')P^{|k|}(x', x), & k < 0. \end{cases}$$

We then have

$$\begin{aligned}
N(x, x') &= T\, \mathbb{E}_{\tau,i,j}\big[\gamma^{|i-j|}\, \mathbf{1}[x_i = x, x_j = x']\big] && (35)\\
&= T\, \mathbb{E}_{i,j}\big[\gamma^{|i-j|} \Pr(x_i = x, x_j = x'|i, j)\big] && (36)\\
&= T\, \mathbb{E}_{i,j}\big[\gamma^{|i-j|} q_{j-i}(x, x')\big] && (37)\\
&\overset{k=j-i}{=} T\, \mathbb{E}_k\big[\gamma^{|k|} q_k(x, x')\big] && (38)\\
&= T \sum_{k=-T+1}^{T-1} p(k)\gamma^{|k|} q_k(x, x') && (39)\\
&= T \sum_{k=-T+1}^{-1} p(k)\gamma^{|k|} q_k(x, x') + T \sum_{k=0}^{T-1} p(k)\gamma^{|k|} q_k(x, x') && (40)\\
&= T \sum_{k=-T+1}^{-1} \frac{T+k}{T^2}\gamma^{|k|}\mu(x')P^{|k|}(x', x) + T \sum_{k=0}^{T-1} \frac{T-k}{T^2}\gamma^k \mu(x)P^k(x, x') && (41)\\
&= \sum_{k=-T+1}^{-1} \frac{T+k}{T}\gamma^{|k|}\mu(x')P^{|k|}(x', x) + \sum_{k=0}^{T-1} \frac{T-k}{T}\gamma^k \mu(x)P^k(x, x') && (42)\\
&\overset{k'=-k}{=} \sum_{k'=1}^{T-1} \frac{T-k'}{T}\gamma^{k'}\mu(x')P^{k'}(x', x) + \sum_{k=0}^{T-1} \frac{T-k}{T}\gamma^k \mu(x)P^k(x, x') && (43)\\
&= \mu(x')f_T(x', x) + \mu(x)f_T(x, x') - \mu(x)\mathbf{1}[x = x'], && (44)
\end{aligned}$$

where $f_T(x, x') = \sum_{k=0}^{T-1} \frac{T-k}{T}\gamma^k P^k(x, x')$. Each summand is bounded by γ^k and $\sum_{k=0}^{\infty} \gamma^k < \infty$, so by dominated convergence

$$f_T(x, x') \to \sum_{k=0}^{\infty} \gamma^k P^k(x, x') = M^\beta(x, x'). \tag{45}$$

Thus, we conclude that

$$N(x, x') \to \mu(x')M^\beta(x', x) + \mu(x)M^\beta(x, x') - \mu(x)\mathbf{1}[x = x']. \tag{46}$$

Denominator. We can deal with the denominator in a similar fashion to the numerator.

$$D(x,x') = \mathbb{E}_{\tau,i,j}\big[\mathbf{1}[x_i = x, x_j = x']\big] \tag{47}$$
$$= \mathbb{E}_{i,j} q_{j-i}(x,x') \tag{48}$$
$$\overset{k=j-i}{=} \mathbb{E}_k q_k(x,x') \tag{49}$$
$$= \sum_{k=-T+1}^{-1} p(k) q_k(x,x') + \sum_{k=0}^{T-1} p(k) q_k(x,x') \tag{50}$$
$$= \sum_{k=-T+1}^{-1} \frac{T+k}{T^2}\mu(x')P^{|k|}(x',x) + \sum_{k=0}^{T-1} \frac{T-k}{T^2}\mu(x)P^k(x,x') \tag{51}$$
$$\overset{k'=-k}{=} \sum_{k'=1}^{T-1} \frac{T-k'}{T^2}\mu(x')P^{k'}(x',x) + \sum_{k=0}^{T-1} \frac{T-k}{T^2}\mu(x)P^k(x,x') \tag{52}$$
$$= \mu(x')g_T(x',x) + \mu(x)g_T(x,x') - \frac{1}{T}\mu(x)\mathbf{1}[x=x'], \tag{53}$$

with $g_T(x,x') := \sum_{k=0}^{T-1} \frac{T-k}{T^2} P^k(x,x')$. We have that $0 \le \frac{T-k}{T^2} \le \frac{1}{T}$ summing to $\frac{T+1}{2T} \to \frac{1}{2}$; moreover, due to irreducibility and aperiodicity, $P^k(x,x') \to \mu(x')$. Hence, by the Toeplitz lemma, $g_T(x,x') \to \frac{\mu(x')}{2}$, and finally $D(x,x') \to \mu(x)\mu(x')$.

Dividing the limits of $N(x,x')$ and $D(x,x')$ concludes the proof.

This concludes our analysis: embedding similarity is encouraged to regress a target that contains a symmetrized likelihood ratio, holding the behavioral successor measure as its numerator.

D.5 Additional result: kernel dominance

We have established that a label kernel can make a particular target linearly recoverable from embeddings. We now ask when one kernel produces a representation that is at least as useful as another for every linear probe task. This provides a notion of superiority of kernels.

Recall that K^α is the centered, tempered target. At the optimum, the *rescaled* embeddings satisfy $\bar{H}\bar{H}^\top = K^\alpha$. However, a downstream probe reads the encoder output $H^\star$, whose rows have fixed norm $\sqrt{d}$. We therefore first express the geometry of $H^\star$ directly in terms of K^α. Since K^α is the Gram matrix of $\bar{H}$, its diagonal entry K^α_{ii} is the squared length of row $\bar{h}_i$. Dividing each inner product by the lengths of the two rows gives their cosine similarity. Define

$$K^{\cos}_{ik} = \frac{K^\alpha_{ik}}{\sqrt{K^\alpha_{ii}}\sqrt{K^\alpha_{kk}}}.$$

Thus $K^{\cos}$ records the pairwise similarities after removing the row lengths prescribed by K^α. In particular, $K^{\cos}_{ii} = 1$. Recall also that the loss rescales each encoder output according to $\bar{h}_i = \sqrt{K^\alpha_{ii}}\frac{h^\star_i}{\sqrt{d}}$. At a global minimizer, it follows that

$$K^\alpha_{ik} = \langle \bar{h}_i, \bar{h}_k \rangle = \frac{\sqrt{K^\alpha_{ii}}\sqrt{K^\alpha_{kk}}}{d}\langle h^\star_i, h^\star_k \rangle.$$

Consequently, $\frac{1}{d}H^\star(H^\star)^\top = K^{\cos}$. This identifies $K^{\cos}$ as the cosine Gram matrix of the encoder output.

A scalar linear probe predicts y as $H^\star w$, a linear combination of the columns of $H^\star$. Thus the exactly recoverable targets form $\mathrm{col}(H^\star)$. Using the identity $\mathrm{col}(HH^\top) = \mathrm{col}(H)$ and the relation $H^\star(H^\star)^\top = dK^{\cos}$, we obtain

$$\mathrm{col}(H^\star) = \mathrm{col}(K^{\cos}).$$

For $j \in \{1,2\}$, let K^α_j be the centered, tempered target constructed from K_j, with positive diagonal entries and rank at most d. Let $H^\star_j$ be any corresponding global minimizer subject to $\|h_i\| = \sqrt{d}$. We say that K_1 **dominates** K_2 for linear probing if

$$\min_W \|Y - H^\star_1 W\|_F^2 \le \min_W \|Y - H^\star_2 W\|_F^2 \tag{54}$$

for every vector-valued target Y. Dominance is strict if the inequality is strict for at least one target.

Theorem 5 (Kernel dominance)**.** *Under the assumptions above, K_1 dominates K_2 if and only if*

$$\mathrm{col}(K_2^{cos}) \subseteq \mathrm{col}(K_1^{cos}). \tag{55}$$

Dominance is strict if and only if this inclusion is strict.

Proof. Suppose first that $\mathrm{col}(K_2^{\cos}) \subseteq \mathrm{col}(K_1^{\cos})$. Since $\mathrm{col}(H_j^\star) = \mathrm{col}(K_j^{\cos})$, every column of $H_2^\star$ is a linear combination of the columns of $H_1^\star$. Hence there exists $T \in \mathbb{R}^{d\times d}$ such that $H_2^\star = H_1^\star T$. In particular, any prediction made from $H_2^\star$ can also be made from $H_1^\star$. To see this explicitly, let $W_2^\star$ minimize the probe error for a target Y. We have that

$$\begin{aligned}\min_W \|Y - H_1^\star W\|_F^2 &\le \|Y - H_1^\star T W_2^\star\|_F^2 \\ &= \|Y - H_2^\star W_2^\star\|_F^2 \\ &= \min_W \|Y - H_2^\star W\|_F^2 .\end{aligned}$$

Thus K_1 dominates K_2.

Conversely, suppose K_1 dominates K_2. Take any scalar target $y \in \mathrm{col}(K_2^{\cos})$. This target is exactly recoverable from $H_2^\star$, so dominance implies

$$0 \le \min_w \|y - H_1^\star w\|^2 \le \min_w \|y - H_2^\star w\|^2 = 0.$$

It is therefore also exactly recoverable from $H_1^\star$, giving $y \in \mathrm{col}(H_1^\star) = \mathrm{col}(K_1^{\cos})$. Since this holds for every $y \in \mathrm{col}(K_2^{\cos})$, the required inclusion follows. If the inclusion is strict, choose a target $y \in \mathrm{col}(K_1^{\cos}) \setminus \mathrm{col}(K_2^{\cos})$. Its optimal probe error is zero under K_1 and positive under K_2, establishing strict dominance. If the column spaces coincide, their optimal probe errors agree for every target. Thus strict dominance requires strict inclusion. □