A Grassmannian Latent Space for Goal-Conditioned Navigation

Method, protocol and results on TwoRoom-v1

Charles De Clercq

Abstract

We replace the predictor of a pretrained world model with a model whose latent space is the total space of the tautological bundle over a Grassmannian. A state is a pair (Z, v) with Z given by an orthonormal basis of a k-dimensional subspace of $\mathbb{R}^n$ (Stiefel variety) and v its fibre coordinates. Orthonormality is enforced at every forward pass by a Newton-Schulz iteration, which makes representation collapse structurally impossible without any regularisation term. On TwoRoom-v1 this reaches $98.3\% \pm 1.4\%$ success over six seeds with 26,760 trainable parameters, replacing a 7,039,920-parameter predictor (with visual encoder frozen). This document gives the construction, training and evaluation protocol, and the measured results.

Contents

1 Setup

LeWM [1] is a pixel-observation world model for goal-conditioned navigation. Its pipeline is

$$\text{image}_t \xrightarrow{\text{ViT-tiny}} c_t \in \mathbb{R}^{192} \xrightarrow{\text{MLP}} h_t \in \mathbb{R}^{256} \xrightarrow{\text{Transformer}} \hat{h}_{t+1},$$

with planning by the Cross-Entropy Method (CEM) using mean squared error in $\mathbb{R}^{256}$ as cost. It reports $\approx 84\%$ success on TwoRoom-v1 (6-seed average, from the paper).

We keep its ViT-tiny encoder and freeze it: LeWM's pretrained weights are loaded and never updated. We discard the MLP projector and the Transformer predictor, and train a bundle model in their place.

Component	LeWM	Ours	Status here
ViT-tiny encoder	5,501,376	5,501,376	frozen, LeWM's weights
MLP projector + Transformer	7,039,920	—	discarded
BundleProjector	—	26,248	trained from scratch
MorphismA predictor	—	512	trained from scratch
Trained parameters	12,541,296	26,760	
Inference parameters	12,541,296	5,528,136	

The meaningful comparison is between the parts that differ: a 7,039,920-parameter predictor is replaced by a 26,760-parameter one, a factor of 263. At inference the full system still runs the shared ViT, so the end-to-end cost is 5,528,136 against 12,541,296, a factor of 2.3.

2 The bundle token

2.1 Definition

A *bundle token* is a pair (Z_t, v_t) with

$$Z_t \in \mathrm{St}(k, n) = \{Z \in \mathbb{R}^{n \times k} \mid Z^\top Z = I_k\}, \qquad v_t \in \mathbb{R}^k.$$

The columns of Z_t span a k-dimensional subspace of $\mathbb{R}^n$, that is a point of the Grassmannian $\mathrm{Gr}(k, n)$; v_t holds the coordinates within that subspace. The ambient reconstruction is $\hat{x}_t = Z_t v_t \in \mathbb{R}^n$. Throughout, $n = 16$ and $k = 8$, so $\dim \mathrm{Gr}(8, 16) = k(n-k) = 64$.

The distance used between subspaces is the chordal distance,

$$d_{\text{chord}}(Z_1, Z_2)^2 \;=\; k - \|Z_1^\top Z_2\|_F^2 \;=\; \sum_{i=1}^{k} \sin^2 \theta_i,$$

where the θ_i are the principal angles between the two subspaces. It vanishes exactly when the two subspaces coincide, it is bounded by $\sqrt{k}$, and it costs one matrix product to evaluate.

2.2 Orthonormalisation

After any linear operation that breaks the Stiefel constraint, Z is projected back by the cubic Newton–Schulz iteration

$$Z \leftarrow \tfrac{3}{2}Z - \tfrac{1}{2}ZZ^\top Z,$$

run for three steps. Convergence is local and requires the input to be scaled (in practice $\|Z\|_F < \sqrt{10}$ suffices here). The iteration consists only of matrix products, so it is differentiable and goes through backpropagation for training. This is applied at every forward pass on every sample by construction.

3 Architecture

$$\underbrace{\text{ViT-tiny}}_{\text{frozen}} \longrightarrow c_t \in \mathbb{R}^{192} \longrightarrow \underbrace{\text{BundleProjector}}_{\text{trained}} \longrightarrow (Z_t, v_t) \longrightarrow \underbrace{\text{MorphismA}}_{\text{trained}} \longrightarrow (Z_{t+1}, v_{t+1})$$

3.1 BundleProjector

Two independent linear heads on the CLS token:

$$Z_t = \mathrm{NS}\big(W_Z c_t + b_Z\big).\mathrm{view}(n,k) \in \mathrm{St}(k,n), \qquad v_t = W_v c_t + b_v \in \mathbb{R}^k,$$

with $W_Z \in \mathbb{R}^{192\times nk}$ and $W_v \in \mathbb{R}^{192\times k}$. Parameter count: $192 \cdot 128 + 128 = 24{,}704$ for the Z head and $192 \cdot 8 + 8 = 1{,}544$ for the v head, hence 26,248.

3.2 MorphismA predictor

Given a state (Z_t, v_t) and an action $a_t = (a^{(1)}, a^{(2)}) \in [-1,1]^2$, the predictor builds a linear map affine in the action,

$$A(a_t) = I_n + a^{(1)}B_1 + a^{(2)}B_2, \qquad B_1, B_2 \in \mathbb{R}^{n\times n},$$

then updates the subspace and transports the fibre coordinates,

$$Z_{t+1} = \mathrm{NS}\big(A(a_t)Z_t\big), \qquad v_{t+1} = Z_{t+1}^\top A(a_t) Z_t \, v_t.$$

The $k \times k$ matrix $Z_{t+1}^\top A(a_t) Z_t$ is the transition matrix between the two frames, so the second equation is parallel transport of v from the old frame to the new one.

There is no nonlinearity anywhere in the predictor. Its parameters are the two matrices B_1, B_2, that is $2n^2 = 512$ values. Total trainable parameters for the whole model: **26,760**. Note that although not useful in this context, non-linearity may be introduced in the context : the meaningful operations are meant to be morphisms of vector bundles (see the conclusion).

4 Training

4.1 Data

Dataset `tworoom.h5`: 10,000 scripted and random episodes, 920,809 frames, 224×224 pixel observations, continuous actions in $[-1,1]^2$. Training uses 2,000 of the available episodes.

4.2 Encoder pass and centering

The frozen ViT is run once over all frames and the CLS tokens are cached ($\approx 297\,\mathrm{MB}$ in float16, about 17 minutes on an A100).

Raw CLS tokens have a mean pairwise cosine of ≈ 0.97: all features point in nearly the same direction. The dataset mean $\bar{c} = \frac{1}{N}\sum_i c_i$ is subtracted, which brings the mean pairwise cosine to ≈ 0.04. The vector $\bar{c}$ is stored in the checkpoint and subtracted identically at evaluation time.

4.3 Objective

Each sample is a sequence $(c_t, a_t, c_{t+1}, \ldots, c_{t+H})$ of horizon $H = 5$, drawn within episode boundaries. A target network, a copy of the BundleProjector updated as an exponential moving average

$$\theta_{\mathrm{EMA}} \leftarrow \tau\theta_{\mathrm{EMA}} + (1-\tau)\theta_{\mathrm{online}}$$

with $\tau = 0.99$, produces targets $(\bar{Z}_h, \bar{v}_h)$ from the true future frames. The online model is rolled out autoregressively from $(Z_0, v_0) = \mathrm{BundleProjector}(c_t)$.

The loss has two terms:

$$\mathcal{L} = \underbrace{\frac{1}{H}\sum_{h=1}^{H}\Big(1 - \cos\big(Z_h v_h,\ \bar{Z}_h \bar{v}_h\big)\Big)}_{\mathcal{L}_{\mathrm{ambient}}} + \lambda_Z \underbrace{\frac{1}{H}\sum_{h=1}^{H}\Big(k - \|Z_h^\top \bar{Z}_h\|_F^2\Big)}_{\mathcal{L}_{\mathrm{chord}}}, \qquad \lambda_Z = 0.1. \tag{1}$$

The first term is a JEPA-style cosine loss between the predicted and target ambient embeddings. The second is the squared chordal distance between the predicted and target subspaces. It supervises the geometry of Z directly, requiring the predicted subspace to land near the target on $\mathrm{Gr}(k, n)$ and not merely to produce the right ambient product.

There is no loss on v. The fibre coordinates are supervised only implicitly, through $\mathcal{L}_{\mathrm{ambient}}$ and through the transport structure of the predictor.

4.4 Hyperparameters

Optimiser	AdamW
Learning rate	3×10^{-4}, cosine annealing
Weight decay	10^{-4}
Gradient clipping	1.0, global norm
Batch size	512 sequences
Epochs	50
Training episodes	2,000
Horizon H	5
EMA τ	0.99
λ_Z	0.1
(n, k)	$(16, 8)$

Training takes about 17 minutes on an A100 once the CLS tokens are cached.

5 Planning

At evaluation time the goal frame is encoded by the online BundleProjector, giving (Z_g, v_g). CEM searches for an action sequence minimising the chordal distance between the rolled-out subspace

and the goal subspace,

$$c_{\text{chordal}}(Z_H, Z_g) = k - \|Z_H^\top Z_g\|_F^2.$$

The criterion is purely geometric and does not involve v. It measures how far the predicted subspace is from the goal subspace on $\mathrm{Gr}(8, 16)$, and vanishes exactly when they coincide.

Action samples per iteration	300
Elite set	30
Refinement iterations	30
Planning horizon	5
Action block	5 environment steps per planned action
Evaluation budget	50 steps
Goal offset	25 frames

6 Results

6.1 Main result

TwoRoom-v1, 50 episodes per seed, seeds 42 to 47.

Model	Trained	Inference	Success rate	Per-seed
$\mathrm{Gr}(8, 16)$, chordal loss	26,760	5,528,136	$\mathbf{98.3}\% \pm \mathbf{1.4}\%$	98, 98, 100, 96, 98, 100
$\mathrm{Gr}(8, 16)$, ambient only	26,760	5,528,136	$95.7\% \pm 3.1\%$	94, 92, 98, 92, 98, 100
Sub-JEPA [2]	—	—	$\approx 95\%$	from the paper
LeWM [1]	12,541,296	12,541,296	$\approx 84\%$	from the paper

The Sub-JEPA and LeWM figures are the ones reported by their authors and were not reproduced here. The result above was re-verified from the checkpoint `bundle_n16_k8_morphism_a_lc0.1.pt`.

6.2 Effect of the chordal term

Varying λ_Z and evaluating under four planning criteria (ambient distance, chordal distance, fibre distance, and combined cost):

λ_Z	ambient	chordal	fibre	combined
0.0	**94**%	34%	84%	42%
0.1	76%	**98**%	70%	**98**%
0.5	diverges, gradient explosion at epoch 1			

Without the chordal term the subspace is geometrically unconstrained and planning on it fails: 34% under the chordal criterion. With $\lambda_Z = 0.1$ the subspace tracks the trajectory and the chordal criterion becomes the best one available. At $\lambda_Z = 0.5$ the chordal gradient destabilises the Newton–Schulz iteration and training diverges; 0.1 is the only nonzero value that was found stable.

6.3 Geometry against parameter count

Model	Trained parameters	Success rate
Flat MLP, $h = 60$	27,384	30%
Flat MLP, $h = 117$	53,034	28%
$\mathrm{Gr}(4, 16)$ bundle	13,636	84%
$\mathrm{Gr}(8, 16)$ bundle	26,760	**98.3**%

A flat MLP at the same budget reaches 30%, and doubling its budget does not help. The gain is therefore not a compression effect. The choice of dimensions is interesting by itself: it can be seen as a measure of the problem/world modelling complexity and can be suited to a needed situation.

6.4 Anti-collapse

Adding SIGReg, the anti-collapse regulariser used by Sub-JEPA, on top of the bundle model changes nothing: all variants score between 95% and 99% with no significant winner. The Stiefel constraint enforced by Newton–Schulz is sufficient on its own. This is a structural rather than a statistical guarantee. Collapse means the encoder producing a constant representation. The compacity of the Grassmannian and Newton-Schulz at every forward pass ensure norms are bounded and directions distinct by construction, independently of the data distribution or the optimizer.

7 Conclusion

7.1 What was done

The latent space of a world model was replaced by the total space of the tautological bundle over a Grassmannian and the predictor is a bundle morphism given by a single linear map acting on subspaces and pointed vectors. Planning minimises the chordal distance between the predicted subspace and the goal subspace. Orthonormality is restored in a differentiable way at every forward pass by Newton-Schulz iterations.

7.2 What it achieves

On TwoRoom-v1, $98.3\% \pm 1.4\%$ over six seeds, against $\approx 84\%$ reported for LeWM and $\approx 95\%$ for Sub-JEPA. The predictor of LeWM has 7,039,920 parameters, while ours 26,760 (263 times less).

The gain does not come from the parameter count : a flat MLP of the same size reaches only 30%, and doubling its budget does not help. What the Grassmannian provides is an inductive bias that a affine latent space has no mechanism to express. Moreover, the dimension of the used Grassmannian may be thought of as a measure of the World Model complexity. As far as TwoRoom goes, the optimal dimension discovered is $\mathrm{Gr}(8, 16)$.

7.3 Collapse becomes impossible

The standard way to deal with the collapse of World Models is to add regularisers, such as SIGReg in the case of Sub-JEPA, at a real cost in training. Here the Stiefel constraint offers a garanteed safeguard versus collapse, and as the tests show, adding SIGReg on top of our geometric framework does not have any effect.

7.4 What the subspace learns

The model is never told where the agent of TwoRoom is, it only sees images and actions. We retraced after training what information the subspace effectively contains over 3720 frames (R^2 is the fraction of variance recovered; 1 is exact):

Representation	$R^2(x)$	$R^2(y)$	R^2(room)
Subspace Z	**0.989**	**0.992**	**0.979**
Ambient Zv	0.894	0.968	0.912
Fibre v	0.313	0.438	0.093

The subspace recovers by itself the agent's coordinates, and which room it is in, almost exactly. This is also why planning works: if the subspace is a faithful map of position, the chordal distance between two subspaces approximates the distance between the corresponding positions.

7.5 Scope and perspectives

TwoRoom-v1 is a two-dimensional navigation task, and the visual encoder is LeWM's: nothing here is learned from pixels. In this simple setting, the predictor was deliberately kept to be a simple single linear map. This approach is shown to be enough since the dynamics of TwoRoom are not too complicated. The bundle mechanism we introduced does not rely on linear maps, any morphism respecting the bundle structure can be used. A richer environment may need subtler morphisms, and the bundle structure remains perfectly suited, keeping the Stiefel constraint, the chordal metric and the planning criterion as they are.

What the experiment establishes is narrow and specific: on a task where a flat model of the same size reaches 30%, putting a Grassmannian structure on the latent space and planning by chordal distance reaches 98.3%, the learned subspace turns out to encode the agent's position almost exactly, and the anti-collapse machinery that the flat approach requires becomes unnecessary.

References

[1] Maes, L., Le Lidec, Q., Scieur, D., LeCun, Y., Balestriero, R. *LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels*. arxiv:2603.19312 (2026).

[2] Zhao, K., Nie, D., Lin1, Y., Luo, Z., Gu1, Y., Fan3, D-P. Dan Zeng1*Sub-JEPA: Subspace Gaussian Regularization for Stable End-to-End World Models*, arxiv:2605.09241 (2026).