Disentanglement by Prediction: A Twin Autoencoder with a Factored Latent Traveler

Audun Myers AUDUN.MYERS@PNNL.GOV
Timothy Marrinan TIMOTHY.MARRINAN@PNNL.GOV
Sarah Scullen SARAH.SCULLEN@PNNL.GOV
Tegan Emerson TEGAN.EMERSON@PNNL.GOV
Pacific Northwest National Laboratory

Abstract

We recast disentangled representation learning as latent-space prediction with a factored predictor. Existing approaches—β-VAE (Higgins et al., 2017), FactorVAE (Kim and Mnih, 2018), β-TCVAE (Chen et al., 2018)—cast disentanglement as a statistical property of a marginal latent distribution: they identify *what* factors describe an image but cannot predict *how* the latent should move when a factor changes. Drawing on the joint-embedding predictive architecture (JEPA) framework (LeCun, 2022; Assran et al., 2023), in which predicting in latent space yields structured representations, we ask: *what is the right structure for that predictor when the world is governed by independent generative factors?*

We propose the *Factored-Traversal Twin Autoencoder* (FT-TAE): a JEPA-inspired twin-encoder model in which a discrete factor predictor (Gumbel-Softmax (Jang et al., 2017)) identifies what changed between two views and a learnable state-embedding table $\mathbf{E} \in \mathbb{R}^{K \times S \times d}$ composes the latent transition as a sum of per-factor displacements. Each displacement is the difference between two table lookups, so unchanged factors contribute exactly zero by construction and the additive form is order-independent without any explicit regularizer. The system is trained end-to-end with a single reconstruction loss—disentanglement emerges as a byproduct of learning accurate factored transitions, not from a marginal-statistics objective. We evaluate on 3D Shapes (Burgess and Kim, 2018) and MPI3D (Gondal et al., 2019), measuring factor alignment, traversal accuracy, and compositionality.

Keywords: disentangled representation learning, joint-embedding predictive architecture, factored world models, autoencoders

Note on AI use. Portions of this proposal were drafted and edited using Claude (version 4.7 Opus, Anthropic) to assist with writing, editing, and content generation. Initial drafts and all technical content, ideas, and decisions were developed by the authors; all final content was reviewed for accuracy and approved by the authors.

1. Introduction

Recovering the independent factors of variation that generate observed data is a central goal of unsupervised representation learning (Higgins et al., 2017; Locatello et al., 2019). A disentangled representation maps each generative factor (e.g. object color, size, or orientation) to a distinct, independently controllable dimension, supporting systematic generalization, interpretability, and controllable generation. Prior work approaches this goal from two angles

(Section 1.1): shape the marginal statistics of a latent posterior via a modified variational autoencoder (VAE) objective, or predict in latent space with a generic predictor. Neither builds a per-factor structure into the predictor itself. This paper asks: *what is the right inductive bias for the predictor when the signal is governed by a small number of independent generative factors?*

Our answer is that the predictor itself should be factored — one sub-network per factor, composed additively — so disentanglement is enforced by the prediction task rather than by a marginal-statistics regulariser. We propose the *Factored-Traversal Twin Autoencoder* (FT-TAE), an end-to-end architecture that jointly learns a latent representation (shared-weight twin autoencoder), a discrete factor decomposition (Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017)), and a factored latent transition model. We define disentanglement operationally: the model must predict x_2 from x_1 by decomposing the transformation into independent per-factor displacements; if the factors miss the true generative structure, the traveler cannot compose accurate predictions and the reconstruction loss penalises the system. The formulation contributes three inductive biases not present in prior work:

- **Additive, commutative transitions via embedding-table lookup.** The predicted target latent is $\hat{z}_2 = z_1 + \sum_k (\hat{f}_{2,k}^\top \mathbf{E}[k] - \hat{f}_{1,k}^\top \mathbf{E}[k])$, with $\mathbf{E} \in \mathbb{R}^{K \times S \times d}$ a learnable state-embedding table. Each per-factor displacement is a difference of two lookups, so contributions sum and the result is order-independent.

- **Implicit hard-zero gating.** When a factor is unchanged ($\hat{f}_{1,k} = \hat{f}_{2,k}$), the two retrieved vectors are identical and the difference is exactly zero — no gate needed.

- **End-to-end gradient flow.** All four sub-networks are optimised jointly under a single image-space reconstruction loss: the encoder learns a latent space that is reconstructable *and* easy to traverse; the factor predictor learns factors useful for traversal, not just for latent reconstruction.

1.1. Related work

The dominant approach to disentanglement modifies the VAE evidence lower bound (ELBO) (Kingma and Welling, 2014) to encourage a factorised latent posterior: β-VAE (Higgins et al., 2017) upweights the Kullback–Leibler (KL) divergence term, capacity-controlled β-VAE (Burgess et al., 2018) anneals a channel-capacity target, FactorVAE (Kim and Mnih, 2018) penalises total correlation directly, and β-TCVAE (Chen et al., 2018) isolates the same term through an ELBO decomposition. All four constrain the marginal statistics of $q(z)$ — they describe what an image contains but not how the latent should move when a factor changes — and Locatello et al. (2019) show that a purely unsupervised version of this goal is underdetermined without a model or data bias.

A complementary line predicts in latent space rather than shaping its marginals. JEPA (LeCun, 2022; Assran et al., 2023; Bardes et al., 2024) predicts the embedding of a target from the embedding of a context using a generic Transformer predictor. Latent-dynamics world models pursue a related idea for control: World Models (Ha and Schmidhuber, 2018) and DreamerV3 (Hafner et al., 2023) pair a generative encoder with a learned transition, and Contrastive Structured World Models (C-SWM) (Kipf et al., 2020) factor the transition

by object. FT-TAE shares JEPA's commitment to a latent predictor and C-SWM's commitment to a factored transition, but factors by *generative factor of variation* rather than by object, uses a state-embedding-table lookup rather than a per-factor MLP, and trains through image-space reconstruction rather than a latent-prediction or contrastive loss.

A third line argues that if the true generative factors form a group, the representation should transform equivariantly under its action (Cohen and Welling, 2016; Quessard et al., 2020). Our additive displacements are a discrete, non-parametric instance of the same idea: each slice $\mathbf{E}[k] \in \mathbb{R}^{S\times d}$ implements an action of the cyclic group $\mathbb{Z}/S\mathbb{Z}$ on the latent by translation, and the sum-of-displacements form makes the joint action of all K slots Abelian — a discrete torus $(\mathbb{Z}/S\mathbb{Z})^K$ acting freely on $\mathbb{R}^d$. Three empirical properties we report in Section 4 — exact order-independence, exact zero-displacement on unchanged factors, and per-factor direction consistency across source images — fall out of this structure by construction rather than from an equivariance regulariser.

1.2. Preliminaries and notation

We observe images $x \in \mathcal{X}$ generated from a small number of independent *ground-truth factors* $y = (y_1, \ldots, y_{K^{\mathrm{gt}}})$, each taking one of a finite set of discrete states (for 3D Shapes, K^{gt}=6: floor / wall / object hue, scale, shape, orientation; for MPI3D, K^{gt}=7). The learner is given *pairs* (x_1, x_2); some factors may agree between x_1 and x_2 and some may differ, and the learner is not told which. This pair-based setting is what lets the model be trained by prediction — the pair carries the "what changed" signal that a single image cannot.

Throughout the paper, three learnable maps act on any image: an *encoder* $E : \mathcal{X} \to \mathbb{R}^d$ producing a continuous latent code $z = E(x)$, a *decoder* $D : \mathbb{R}^d \to \mathcal{X}$ that maps latents back to images, and a *factor predictor* $F : \mathbb{R}^d \to \{0, 1\}^{K\times S}$ that produces a discrete $K \times S$ factor template $\hat{f} = F(z)$ with one active entry per row. The factor template is the discrete summary of what factors are present in an image: each of its K rows encodes one factor slot as a one-hot vector over S possible states.

We use "*slot*" (indexed by $k \in \{1, \ldots, K\}$) for the model's K predicted factor dimensions and "*ground-truth factor*" (indexed by $j \in \{1, \ldots, K^{\mathrm{gt}}\}$) for the true underlying axes of variation. The number of slots K is a hyperparameter of the model; the number of ground-truth factors K^{gt} is a property of the dataset. There is no built-in one-to-one correspondence between the two indexings — learning that correspondence is exactly what we mean by disentanglement, and factor-slot alignment (how tightly slot k tracks a single ground-truth factor j) is what Figure 3 measures.

Finally, the acronym FT-TAE stands for *Factored-Traversal Twin Autoencoder*. "Twin" refers to the shared-weight application of E to both members of a pair, so $z_1 = E(x_1)$ and $z_2 = E(x_2)$ live in the same continuous latent space. A *factored traversal* is the operation of composing a latent transition $z_1 \mapsto \hat{z}_2$ as an unordered sum of per-factor displacements (defined in Section 3); the module that implements this operation is called the *factored traveler* and is denoted T. Informally, the traveler is the noun (a network module, on par with E, F, D) and a traversal is the verb (the operation the traveler performs). The paper is named after the operation because factored traversal is the abstract property we care about; the traveler is the concrete implementation we chose.

2. Sequential Baseline Pipeline

To motivate the end-to-end design of Section 3, we first sketch a three-stage *sequential* pipeline that trains the encoder, factor labeler, and traveler in isolation. The failure modes of this pipeline motivate the unified model.

Stage 1: encoder–decoder. A convolutional autoencoder (or β-VAE (Higgins et al., 2017)) with the 3D Shapes backbone (four strided conv blocks, channels 32/32/64/64; stride 2; kernel 4) maps images to $z \in \mathbb{R}^d$ (d=10); the decoder mirrors the encoder. The same family is reused — widened to channels 64/64/128/128 and d=128 — in FT-TAE (Section 3); VAE baselines (Section 4) are held to $d \leq 10$.

Stage 2: factor labeler. A factor autoencoder (FAE) over $z \in \mathbb{R}^d$ discretises the latent into a $K \times S$ template. Its encoder is a shared multi-layer perceptron (MLP) $\mathbb{R}^d \to 128 \to KS$, reshaped to $K \times S$ and discretised row-wise via straight-through Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017); the decoder maps the flattened one-hot template back to $\hat{z} \in \mathbb{R}^d$. Training minimises latent-reconstruction mean-squared error (MSE) plus an entropy regulariser. This shared-MLP head is replaced in FT-TAE by K *independent* per-factor heads to avoid gradient cross-contamination.

Stage 3: sequential traveler. Given templates f_s, f_t for a pair, a per-factor MLP predicts $\Delta z_k = g_k(z, \Delta f_k)$, applied along an ordering π of $\{1, \dots, K\}$:

$$z^{(i)} = z^{(i-1)} + \Delta z_{\pi_i}\big(z^{(i-1)}, \Delta f_{\pi_i}\big), \qquad i = 1, \dots, K.$$

Because each update conditions on the running latent, the traversal is *path-dependent*: applying the same changes in a different order yields a different decoded image. The three stages are also trained in isolation, so the encoder is unaware of factor structure and the labeler is unaware of how its templates will be consumed.

The unified architecture of Section 3 fixes both issues: an additive state-embedding-table lookup replaces the sequential MLPs (making traversal order-independent), and joint training couples all four sub-networks under a single image-space objective.

3. The Factored-Traversal Twin Autoencoder

FT-TAE is trained end-to-end on image pairs (x_1, x_2) and comprises four components (Figure 1). A shared-weight convolutional encoder E maps each image to a latent code $z_i = E(x_i) \in \mathbb{R}^d$ with d=128. A factor predictor F maps a latent code to a discrete $K \times S$ factor template $\hat{f}_i \in \{0, 1\}^{K \times S}$ via K independent per-factor heads (two-layer MLPs, hidden 128), discretised row-wise with straight-through Gumbel-Softmax (Jang et al., 2017) during training and argmax at inference; we use K=6 slots and S=15 states. The factored traveler T is a learnable state-embedding table $\mathbf{E} \in \mathbb{R}^{K \times S \times d}$ that composes the target-latent prediction as an additive sum of per-factor displacements:

$$\hat{z}_2 = z_1 + \sum_{k=1}^{K} \big(\hat{f}_{2,k}^{\top} \mathbf{E}[k] - \hat{f}_{1,k}^{\top} \mathbf{E}[k]\big). \tag{1}$$

Each term is the displacement for factor slot k; when a factor is unchanged the two lookups are identical and contribute exactly zero, and the sum is order-invariant by construction. A transposed-conv decoder D (mirroring E; final sigmoid) produces $\hat{x}_2 = D(\hat{z}_2)$.

How one pair (x_1, x_2) flows through the model. As a concrete example, take a 3D Shapes pair that differs only in object hue. The shared encoder gives $z_1{=}E(x_1)$ and $z_2{=}E(x_2)$; the factor predictor is applied to each latent independently, producing one-hot templates $\hat{f}_1{=}F(z_1)$ and $\hat{f}_2{=}F(z_2)$ that describe what factors are present in each image (there is no direct comparison between x_1 and x_2 inside F). The traveler then adds the six per-factor displacements to z_1: five of them are exactly zero because $\hat{f}_{1,k}{=}\hat{f}_{2,k}$ for the unchanged factors, and only the object-hue term contributes. The decoder produces $\hat{x}_2$, and the pixel-space reconstruction loss $\|\hat{x}_2{-}x_2\|^2$ backpropagates through all four sub-networks jointly.

Objective and training. The training objective is pixel-space reconstruction of the target with an optional cross-entropy (CE) auxiliary that trains F against ground-truth factor labels when available:

$$\mathcal{L} \;=\; \|\hat{x}_2 - x_2\|^2 \;+\; \lambda_{\text{ce}}\,\text{CE}\big(F(z),\, y\big). \tag{2}$$

The unsupervised setting uses $\lambda_{\text{ce}} = 0$; the semi-supervised setting uses $\lambda_{\text{ce}} = 1$. Because the traveler has the fixed additive form of Eq. (1), reconstruction alone is enough to shape the encoder, factor predictor, and state-embedding table into a consistent factored representation — the only way for F to keep the loss low is to produce codes whose corresponding rows of $\mathbf{E}$ sum to z_2. This is why the model needs no separate latent-consistency, augmentation-consistency, or slot-entropy regulariser: the compositional structure they usually enforce is provided by the shape of Eq. (1) rather than by a loss. Training has two phases: encoder–decoder pretraining for T_{pre} epochs, then joint training of all four sub-networks with Eq. (2) on random pairs. Weights are optimised with Adam; the Gumbel temperature anneals as $\tau_t = \max(0.1,\ 0.5 \cdot 0.99^t)$. Full hyperparameters are in Appendix A.

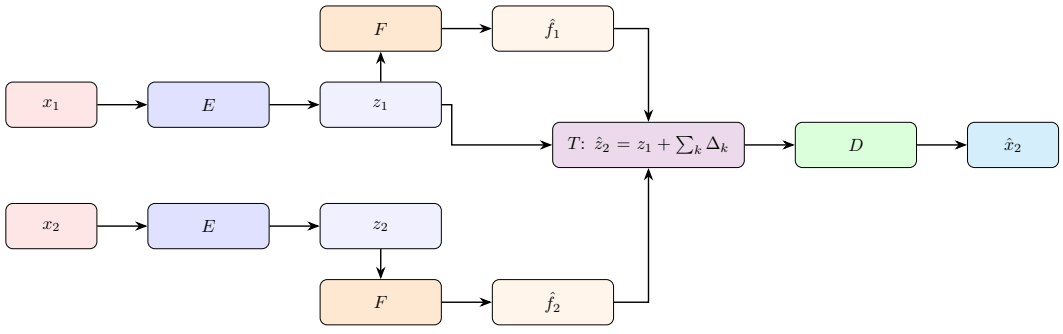

Figure 1: FT-TAE architecture. Shared-weight E produces z_1, z_2; F produces discrete templates $\hat{f}_1, \hat{f}_2$; T forms $\hat{z}_2$ by adding per-factor embedding differences (unchanged factors contribute zero); D produces $\hat{x}_2$. The loss $\|\hat{x}_2 - x_2\|^2$ backpropagates through all four sub-networks.

4. Results

We evaluate two FT-TAE variants (unsupervised and semi-supervised with λ_{ce}=1) against four VAE baselines — β-VAE (Higgins et al., 2017), capacity-controlled β-VAE (Burgess et al., 2018), FactorVAE (Kim and Mnih, 2018), β-TCVAE (Chen et al., 2018), all at latent dimension $d \leq 10$ — on 3D Shapes (Burgess and Kim, 2018) (K^{gt}=6: floor / wall / object hue, scale, shape, orientation) and MPI3D-realistic (Gondal et al., 2019) (K^{gt}=7). All models use the same convolutional backbone. For each dataset, 10,000 images are held out for evaluation and the reported metrics are computed there. We report reconstruction MSE ($\|D(E(x))-x\|^2$), linear-probe accuracy of each ground-truth factor from the encoder output (mean 5-fold cross-validation, CV), single-factor traversal MSE (SF-Trav; the pixel error of decoding a latent in which only one factor has been swapped, aligned to the ground truth via Hungarian matching for the VAE baselines), and Mutual Information Gap (MIG) (Chen et al., 2018).

4.1. 3D Shapes

Table 1 summarises the principal metrics. The semi-supervised FT-TAE (1,000 ground-truth-labelled pairs, equivalent to $\sim$45 labelled images since $\binom{n}{2}$ pairs come from n images) attains SF-Trav MSE $\mathbf{5.53 \times 10^{-3}}$ per pixel, more than 5$\times$ better than the best VAE baseline (30.6×10^{-3}), and the unsupervised variant attains the best reconstruction fidelity of all six models (0.71×10^{-3}). Both FT-TAE variants reach 93–97% probe accuracy vs. 58–69% for the baselines. MIG goes the other way — but MIG is biased downward by high-dimensional latents, and FT-TAE keeps a d=128 continuous latent on top of the discrete slots vs. $d \leq 7$ for the baselines. Qualitatively, Figure 2 shows that individual displacements Δ_k decoded from a source x_1 produce clean isolated edits under the semi-supervised model, and Figure 3 confirms one-to-one slot / ground-truth alignment: entry (k, j) is the row-normalised mutual information $I(\hat{y}_k; y_j^{\text{gt}})$ between the slot's argmax code $\hat{y}_k$ and the ground-truth factor value y_j^{gt}, estimated by histogram binning over the 10,000-image held-out set; a permutation-of-the-identity pattern indicates clean disentanglement.

Table 1: Quantitative comparison on 3D Shapes (10,000 held-out images). Recon. MSE and SF-Trav. MSE in per-pixel $\ell_2 \times 10^{-3}$ (lower better); probe accuracy is mean 5-fold CV across 6 factors; MIG higher better.

Method	Recon. MSE ↓	Probe acc. ↑	MIG ↑	SF-Trav. MSE ↓
β-VAE (β=8)	3.94	69.3	0.149	30.61
Cap. β-VAE (β=5, C=30)	1.54	69.1	**0.161**	30.73
FactorVAE (γ=10)	6.54	58.0	0.159	36.52
β-TCVAE (β=2)	3.34	57.6	0.089	30.81
FT-TAE (unsup)	**0.71**	**97.1**	0.049	49.19
FT-TAE (semi-sup, 1k)	1.02	93.3	0.059	**5.53**

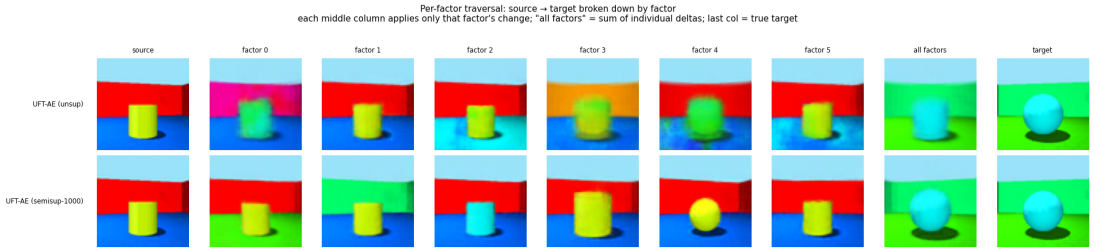

Figure 2: Per-factor traversal on 3D Shapes. Each row is a FT-TAE variant; the leftmost column is the source x_1, each middle column applies only the k-th displacement, the penultimate column is the full traversal (sum of all displacements), the rightmost column is the true target x_2.

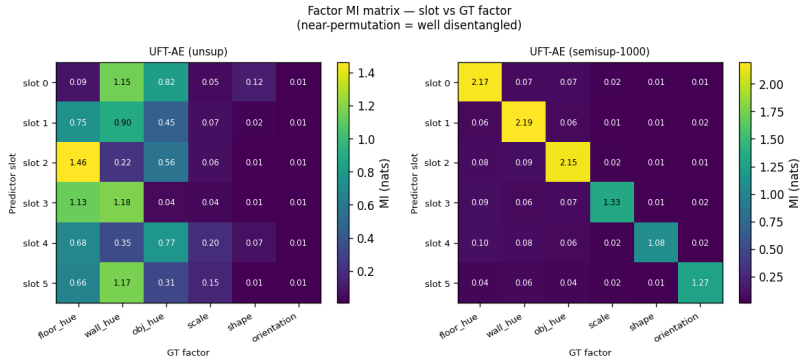

Figure 3: Normalised mutual information $I(\hat{y}_k; y_j^{\mathrm{gt}})$ on 3D Shapes between each of the K=6 predicted factor slots (rows) and the six ground-truth factors (columns), estimated by histogram binning on 10,000 held-out images and row-normalised. Perfect disentanglement is a permutation of the identity up to slot reordering.

4.2. MPI3D

MPI3D-realistic is substantially harder: images are photographic renderings of a robot arm, and the seven ground-truth factors have very different cardinalities — object size (2), camera height and background colour (3), object colour and shape (6), and two dense 40-level axes (horizontal / vertical) that our fixed S=15 per slot under-approximates by design (Figure 4).

Table 2 reports the results. The unsupervised FT-TAE takes the best reconstruction fidelity (5.26×10^{-3}, $\sim 2\times$ better than the strongest baseline) and the semi-supervised variant takes the lowest SF-Trav MSE (23.59×10^{-3}, $\sim$30% below FactorVAE). Both FT-TAE variants beat every baseline on probe accuracy (57–59% vs. 36–41%), with the gain concentrated in low-cardinality factors: object colour (81–86% vs. 20–48%) and object size (65–70% vs. 54–59%). Both 40-level axes remain hard for every model (4–20%), the failure mode discussed in Section 5. MIG is uniformly low across all methods and is dominated by latent-dimension bias, as before. Finally, two properties of the additive traveler that hold by construction on this dataset: pixel MSE between AB and BA factor-change orderings is exactly zero (ratio 0.000 against a random-traversal MSE of 103–111) and the mean pairwise

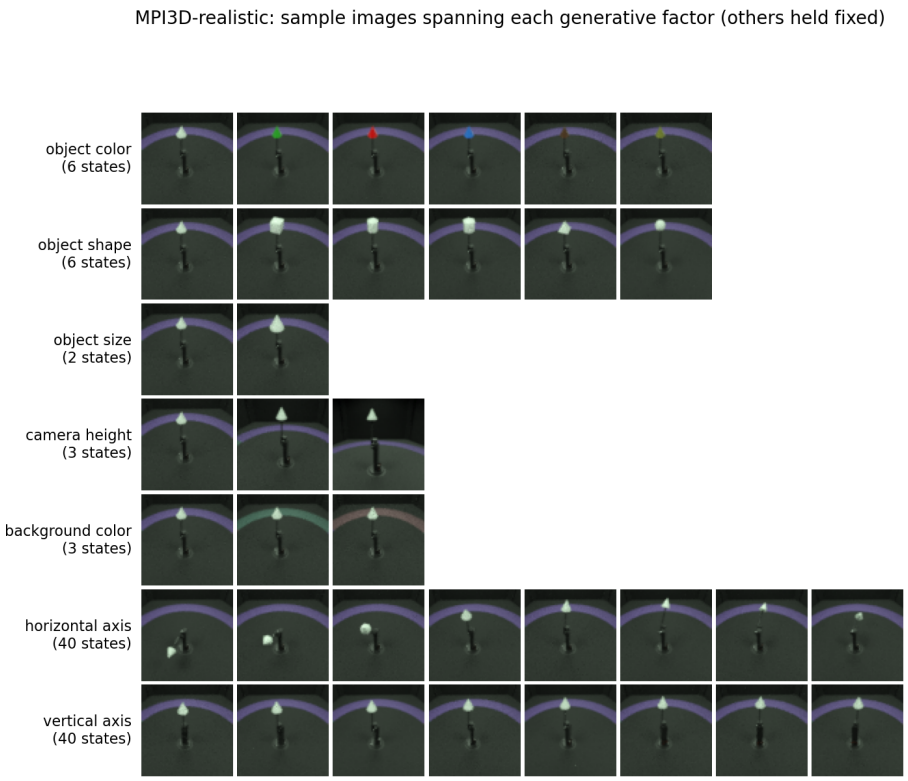

Figure 4: MPI3D-realistic sample images spanning each of the seven generative factors (others held fixed). Rows are at true factor cardinality (2–40 states); no padding.

cosine similarity of Δ_k across source contexts is 1.00 for all seven slots. Every VAE baseline has a monolithic transition and fails both properties.

Table 2: Quantitative comparison on MPI3D-realistic (10,000 held-out images). Columns and units match Table 1; probe accuracy is mean 5-fold CV across 7 factors.

Method	Recon. MSE ↓	Probe acc. ↑	MIG ↑	SF-Trav. MSE ↓
β-VAE (β=8)	31.02	36.4	0.005	32.36
Cap. β-VAE (β=5, C=30)	10.89	41.1	0.034	35.90
FactorVAE (γ=10)	20.34	41.0	**0.078**	31.30
β-TCVAE (β=2)	19.27	40.8	0.066	32.90
FT-TAE (unsup)	**5.26**	**58.5**	0.014	45.17
FT-TAE (semi-sup, 1k)	8.36	57.1	0.018	**23.59**

5. Discussion and Conclusion

We asked in the introduction: what is the right inductive bias for a latent-space predictor when the signal is governed by a small number of independent generative factors? Our exper-

iments give a compact answer for this regime: an additive per-factor state-embedding-table lookup. On 3D Shapes it yields SF-Trav MSE 5.53×10^{-3} for the semi-supervised FT-TAE ($\sim$5.5$\times$ below the best VAE baseline) and 93–97% probe accuracy vs. 58–69%; on MPI3D it takes the best reconstruction fidelity (5.26×10^{-3} vs. 10.9×10^{-3}), the lowest SF-Trav MSE among FT-TAE variants (23.6×10^{-3}, semi-supervised), and exact-by-construction order-independence and per-factor direction consistency. The bias helps most where the factor is close to discrete: on the two 40-level MPI3D axes our fixed S=15 under-approximates the range by design, and every model — ours and the baselines — struggles (4–20% probe accuracy). The formulation fixes K and S as hyperparameters (misspecification yields under-factorisation or dead factors), assumes "no change in factor k" is a meaningful event (exact for discrete factors, approximate for continuous ones), and is evaluated only on synthetic benchmarks. Natural next steps are a learned-cardinality factor predictor, composing the factored bias with image- or video-JEPA (Assran et al., 2023; Bardes et al., 2024), and using the factored traveler as an action model in control settings (Ha and Schmidhuber, 2018; Hafner et al., 2023; Kipf et al., 2020).

References

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

Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *arXiv preprint arXiv:2404.08471*, 2024.

Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, pages 2990–2999, 2016.

Muhammad Waleed Gondal, Manuel Wuthrich, Đorđe Miladinović, Francesco Locatello, Martin Breber, Bernhard Schölkopf, and Stefan Bauer. On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

David Ha and Jürgen Schmidhuber. World models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations (ICLR)*, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning (ICML)*, pages 2649–2658, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2014.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations (ICLR)*, 2020.

Yann LeCun. A path towards autonomous machine intelligence. *Open Review*, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, pages 4114–4124, 2019.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations (ICLR)*, 2017.

Robin Quessard, Thomas D. Barrett, and William R. Clements. Learning disentangled representations and group structure of dynamical environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Appendix A. Reproducibility

This appendix collects the implementation details needed to reproduce the results in Section 4. Values apply to both datasets unless otherwise noted.

A.1. Datasets and preprocessing

3D Shapes (Burgess and Kim, 2018). The full dataset has 480,000 images. From each run we uniformly subsample 10,000 images without replacement (seed 42) as the training pool and hold out the remaining 470,000 as an evaluation set that is not seen during training; all reported metrics are computed on a 10,000-image subset of this held-out pool. Images are RGB, $64 \times 64 \times 3$, and are converted to $[0, 1]$ by dividing by 255. Ground-truth factor labels are discretised per-column by mapping each unique floating-point value to a state index; the resulting cardinalities are $(10, 10, 10, 8, 4, 15)$ for (floor hue, wall hue, object hue, scale, shape, orientation).

MPI3D-realistic (Gondal et al., 2019). The full dataset has 1,036,800 images. We use the same 10,000-image train / held-out-evaluation split protocol as for 3D Shapes (seed 42). The seven factor cardinalities are $(6, 6, 2, 3, 3, 40, 40)$ (object colour, shape, size; camera height; background colour; horizontal axis; vertical axis), as visualised in Figure 4.

Pair sampling. FT-TAE trains on pairs (x_1, x_2). Pairs are formed by sampling x_1 uniformly from the training pool and x_2 independently and uniformly at random from the same pool for each mini-batch. No factor-overlap curriculum is used in the reported runs.

No image augmentation. The reported FT-TAE runs use no image-space augmentation (no brightness/contrast jitter, no cropping, no flipping); the training signal comes entirely from cross-pair reconstruction and, in the semi-supervised setting, the CE auxiliary.

A.2. FT-TAE architecture and hyperparameters

Table 3: FT-TAE architecture and training hyperparameters used for the reported runs on both datasets. Values match those in the released training notebook and *are the values used to produce Tables 1 and 2*. K is fixed to the ground-truth factor count of each dataset ($K^{\text{gt}} = 6$ for 3D Shapes and 7 for MPI3D); $S = 15$ matches the maximum factor cardinality of 3D Shapes but under-approximates the two 40-level MPI3D axes (horizontal / vertical robot-arm axis), which is a source of the harder MPI3D SF-Trav numbers.

Category	Hyperparameter	Value
Architecture	Latent dimension d	128
	Encoder channels	[64, 64, 128, 128]
	Encoder fully-connected hidden	128
	Factor-predictor hidden h	128
	Number of slots K	6 (Shapes3D), 7 (MPI3D)
	States per slot S	15 (both datasets)
Optimisation	Optimiser	Adam (default β_1=0.9, β_2=0.999)
	Learning rate	10^{-3}
	Weight decay	0
	Gradient-norm clip	10.0
	Batch size	512
Schedule	Phase 1 (autoencoder pretrain) epochs T_{pre}	10
	Phase 2 (end-to-end) epochs	100
	Gumbel-softmax τ_t	$\max(0.1,\ 0.5 \cdot 0.99^t)$
	Number of targets per source N_{tgt}	2
Loss	λ_{ce} (unsupervised)	0
	λ_{ce} (semi-supervised)	1
Randomness	PyTorch / NumPy seed	42

Semi-supervised training. In semi-supervised mode we use $\lambda_{\text{ce}} = 1$ and expose the model to ground-truth factor labels for 1,000 image pairs sampled once at the start of training; the traversal loss is applied to all pairs while the CE loss is applied to the labelled subset. All other hyperparameters match the unsupervised run.

A.3. VAE baselines

Table 4: Hyperparameters used for the four VAE baselines reported in Tables 1–2. The chosen β, C, and γ values follow the values reported in the corresponding original papers on the 3D Shapes / dSprites benchmark family; we did not perform an additional grid search over these hyperparameters. All baselines share the same backbone as FT-TAE except for the latent dimension, which is held to $d \leq 10$ following prior work.

Method	Hyperparameters	Reference
β-VAE	$\beta = 8$	Higgins et al. (2017)
Capacity-controlled β-VAE	$\beta = 5,\ C = 30$	Burgess et al. (2018)
FactorVAE	$\gamma = 10$	Kim and Mnih (2018)
β-TCVAE	$\beta = 2$	Chen et al. (2018)
All baselines	Latent dim $d = 10$ Optimiser: Adam, lr $= 10^{-3}$ Batch size $= 128$ Epochs $= 200$	

Each baseline is trained on the same 10,000-image training subset used for FT-TAE, with the same seed and the same held-out evaluation set.

A.4. Number of runs

The results in Tables 1 and 2 are from a single training run per (method, dataset) cell with the seed listed in Table 3. Running multiple seeds and reporting means with standard deviations is a natural next step; we did not do so for this submission because of compute constraints, and note that the exact-zero order-independence (AB/BA MSE ratio = 0.000; direction consistency = 1.00) reported on MPI3D is by construction — not a property that would drift across seeds.