V-JEPA 2 Video Transcoders Find Motion-Based Features

Susan Soraya Afshar Ghassemlou
School of Computer Science
McGill University, Mila
Montreal, Canada
susan.ghassemlou@mail.mcgill.ca
soraya.ghassemlou@mila.quebec

Sonia Joseph
School of Computer Science
McGill University, Mila
Montreal, Canada
sonia.joseph@mila.quebec

***Abstract*—As embodied systems are released with increasingly complex environments, it is necessary to ensure their physical safety. Previous work shows that SAE (Sparse Autoencoder) feature patterns can signal out-of-distribution operation and impending hallucination at inference time. We hypothesize that developing similar interpretability probes for embodied models could eventually help to monitor internal states and detect errors. As an initial step, we apply these techniques to a self-supervised video model, which is increasingly being incorporated into the physical AI stack. We introduce a V-JEPA 2 video transcoder, a sparse decomposition that combines the layer-to-layer prediction objective of sparse transcoders with the convex-hull decoder constraint of archetypal sparse autoencoders. The architecture targets a regime where off-the-shelf sparse trans decomposition has not been characterized: the features of self-supervised video models.**

We instantiate the architecture on V-JEPA 2 features, a frozen self-supervised video model whose representations are optimized for masked latent prediction, and report preliminary observations on prediction quality and on a per-feature label-distribution interpretability protocol over the *Something-Something v2* motion-defined action templates. We find resulting features that visibly track a common motion primitive across different objects, and a statistically enriched sub-population that fires selectively on physical hazards such as falls and collisions. We will release our video sparse transcoder weights to the wider mechanistic interpretability community for further research at: github.com/sghassemlou/V-JEPA-2-Transcoder.

I. Introduction

Sparse autoencoders are now a routine probe for the internal structure of deep models, decomposing dense activations into sparse linear combinations of dictionary atoms [1]–[4]. Past work [5] shows that SAE feature patterns can reliably signal out-of-distribution operation and impending hallucination at inference time by capturing how models traverse ungrounded conceptual landscapes. Translated to embodied systems, analogous probes could potentially help monitor an agent's sensory attention and latent goals, offering a way to detect internal errors in the physical world.

Sparse *transcoders* [6], [7] extend this idea by replacing the activation-reconstruction objective of an SAE with a layer-to-layer prediction objective, producing a sparse factorization of the model's own computation rather than only of its activations. Both methods are well studied for language and image substrates, but predictive video models occupy a different regime. Concurrent work [8] demonstrates that SAE's can destroy temporal coherence in video representations motivating the exploration of alternative structures. The self-supervised video model V-JEPA, which has been proposed as a foundation model in the embodied AI stack for the physical world, is trained to predict masked spatiotemporal regions in latent space, producing features whose defining property is dynamical consistency rather than reconstructive fidelity [9]–[11]. It is unclear whether off-the-shelf sparse decomposition preserves this dynamical structure, and unclear which inductive biases on the dictionary are appropriate when the underlying features are themselves predictions.

For video features, the underlying physical configurations form a low-dimensional manifold inside the much larger space of pixel sequences governed by conservation laws, continuity of motion, and contact dynamics. Because of these environmental constraints, we hypothesize that atoms inside this hull are more likely to correspond to physically realizable configurations than free atoms, as motivation for the architecture, not as a claim we test directly.

We also hypothesize that geometrically anchored dictionary atoms could provide a safety probe for monitoring these latent states. We train *archetypal sparse transcoders* on V-JEPA 2 features, combining the layer-to-layer prediction objective of [6] with the archetypal decoder constraint of [4]. The present results train a single archetypal transcoder on V-JEPA 2.1 ViT-B/16 layer-10 FFN activations from Something-Something v2 [12], whose motion-defined action labels (e.g. *pushing X off Y*) inform feature selection of motion-based features. Cross-substrate (FineVideo, Kinetics) and top-k-transcoder comparisons [3] are out of scope and pre-registered as the next phase (App. A-D). Our contributions are:

- **Sparse transcoders on a video model.** We train a sparse *transcoder*—which factorizes the layer-to-layer computation rather than reconstructing activations—on a self-supervised video model's internal features, complementing concurrent sparse-autoencoder work on video representations [8].
- **Label-distribution protocol for SSv2.** A non-VLM interpretability handle: per-feature histograms over the

174 SSv2 motion templates and over open-vocabulary placeholder tokens (with a per-entry token-presence rule that collapses surface forms like *cup/plastic cup*). Crossing the two concentrations yields candidate regimes (motion-, object-, joint, noise/scene), with sub-regimes (joint-impostor, camera-motion, scene-anchored, motion-narrow), surfaced by inspection.

- **Preliminary observations.** On the best-performing configuration of a Bayesian sweep, held-out $\mathrm{EV} = +0.479$ at $L_0 = 59$, controlled against a random-init ablation at the same architecture ($\mathrm{EV} = -1.4\times10^4$, $L_0 = 6{,}121$). We exhibit motion-aligned features whose filmstrips visibly track a common motion primitive across different objects.
- **A first physical-hazard safety probe.** Reusing only the held-out activations, we find an enriched sub-population of hazard-selective features: 17 of 408 analyzable features fire predominantly on uncontrolled falls, drops, and collisions (top-10 precision ≥ 0.5; four at ≥ 0.8, against a 0.111 chance rate), $5.7\times$ rarer in the random-init control ($p = 5.7\times10^{-3}$), and localized in space and time.

II. Background

A sparse autoencoder (SAE) decomposes activations A as $A \approx ZD$ subject to sparsity on the codes Z, with top-k [3] or L_1 [1], [2] sparsity. A sparse *transcoder* [6], [7] replaces the reconstruction objective with layer-to-layer prediction: given A^{in} and A^{out} at the same nonlinear sub-block, learn sparse codes Z from A^{in} that predict A^{out}. Archetypal SAEs [4], [13] constrain each decoder atom to lie inside the convex hull of a fixed bank of K-means centroids: $D = WC + \Lambda$, W row-stochastic, $\|\Lambda_i\|_2 \leq \delta$ (RA-SAE). While initially proposed to address SAE seed-instability [14], recent findings indicate that unstable features have weak individual functional impact [15].

Consequently, we utilize this parameter-side constraint primarily to explore the inductive bias of geometrically anchored dictionaries, rather than to mitigate instability. This constraint applies to any sparse decoder, including a transcoder's; in our setting, C lives in A^{out} space because the decoder writes there. V-JEPA 2 [9]–[11] is a self-supervised video model that predicts features of masked spatiotemporal regions in latent space; unlike pixel-level masked autoencoders, its objective rewards representations that support latent-space prediction, making its features a natural substrate for studying what video models internally treat as predictable structure.

III. Methods

A. Substrate, hook site, and feature cache

Activations are extracted from a frozen V-JEPA 2.1 ViT-B/16 encoder [10], [11] run at 384 px on $T = 16$-frame clips. We target block $\ell = 10$, which is the standard regime for transcoders on ViT-class encoders. The transcoder operates on $A^{\mathrm{in}} = \mathrm{LN}_2(x_\ell)$, the post-LayerNorm input to the block-10 FFN, and is trained to predict $A^{\mathrm{out}} = \mathrm{FFN}_\ell(A^{\mathrm{in}})$. To break within-clip token correlations that would otherwise inflate L_0 and pull features toward scene detectors [2], [16], tokens are read through a randomized shuffle stream, preserving the per-token provenance row $(\mathrm{src}, \mathrm{clip}, t, h, w)$ needed for downstream spatiotemporal evaluation.

B. Dataset motivation

The present results use Something-Something v2 [12] ($\sim$220k clips) exclusively, with a deterministic $90/10$ train/held-out split keyed on the seed. We use SSv2 to focus the learned dictionary on the kinds of features for which the manifold-structure hypothesis of Sec. I predicts archetypal anchoring should be useful. Because SSv2's action classes are defined primarily by motion dynamics rather than texture or colour, a transcoder trained on its activations should preferentially learn features that mediate the motion dynamics targeted by our hypothesis.

For example, the class "poking something lightly so that it doesn't move" has multiple object identities, so that the class becomes the motion type, and not confounded by the objects in the scene. Caches for FineVideo [17] and the Kinetics-400/600/700 family [18]–[20] are configured with the identical hook-site hash and are reserved for the cross-domain generalization experiment in Section A-D; they are not used here.

C. Architecture and Training Objective

The encoder is a single rectified linear layer $z(x) = \mathrm{ReLU}(W_e x + b_e)$ with $W_e \in \mathbb{R}^{H\times d}$ Kaiming-initialized and $b_e \in \mathbb{R}^H$ zero [2], [6]; sparsity is induced by an L_1 penalty on z rather than by a hard top-K activation. Following [4], we register a frozen pool $C \in \mathbb{R}^{N\times d}$ of N K-means centroids in A^{out} space (App. A-A). The decoder reconstruction $\hat{y}_b$ for an input x_b is:

$$\hat{y}_b = (\sigma_K(A)C + \Lambda)^\top z_b + b_d \tag{1}$$

where $b_d \in \mathbb{R}^d$ is zero-initialized, $A \in \mathbb{R}^{H\times N}$ are free *mixing logits*, σ_K is a row-wise top-K softmax on the K largest entries (and 0 elsewhere), making each row exactly K-sparse and row-stochastic. $\Lambda \in \mathbb{R}^{H\times d}$ is a slack term bounded by an l_2 ball $\|\Lambda_i\|_2 \leq \delta$ (RA-SAE). Setting $\delta = 0$ recovers pure archetypal anchoring; $\delta \to \infty$ recovers an unconstrained decoder.

For a batch B, the transcoder is optimized to predict targets $y_b = \mathrm{FFN}_l(x_b)$ (recomputed at train time) using the loss:

$$\mathcal{L}(B) = \tfrac{1}{|B|}\sum_b \left\|\widehat{y}_b - y_b\right\|_2^2 + \lambda(t)\cdot \tfrac{1}{|B|}\sum_b \|z_b\|_1, \tag{2}$$

We induce sparsity via an L_1 penalty on z rather than a hard top-K encoder activation because the top-K structure already appears inside the decoder via σ_K. The coefficient $\lambda(t)$ is linearly warmed from 0 to λ^* over T_w steps.

Optimization and Constraints: We initialize A near-one-hot ($A_{i,i \bmod N} = \log(100K)$ and zero elsewhere) so atom i starts as centroid $i \bmod N$ deterministically across sweeps. Because row-stochasticity is enforced inside the forward pass, A is unconstrained and optimized by Adam directly. Λ initializes at zero and is optimized using Adam with a global l_2 gradient clip of 1.0 (to prevent divergence from dead-feature gradients),

followed by a row-wise δ-ball projection. This bounded projection naturally provides the "encoder-shrink/decoder-grow" anti-hack invariant of unit-norm SAEs, as the convex hull of C is strictly bounded.

D. Configuration executed

We selected the reported configuration by Bayesian search over three sweeps (focused, pool-32k, and a Prisma-aligned wide-L_1 sweep); see App. A-C for the search spaces. The best-performing trial uses V-JEPA 2.1 ViT-B/16 at 384 px, block-10 FFN on SSv2 [12] with a 90/10 shard split: $d = 768$, $H = 12{,}288$, $N = 32{,}000$, $K = 512$, $\delta = 0.5$, A=I and $\Lambda = 0$ at init, $\lambda^\star = 1.11 \times 10^{-9}$, $T_{\mathrm{w}} = 10{,}000$, Adam $\eta = 4.17 \times 10^{-4}$, grad clip 1.0, batch 4,096 tokens, 50,000 steps, trained on $1\times$ NVIDIA L40S in $\sim$3 h 27 min. We report the post-loop `step_50000` checkpoint; held-out metrics are given in Sec. IV-A. Dead-feature resampling (App. A-B) was scheduled every 25,000 steps with window $W = 12{,}500$, but did not modify the model in this run: the step-25,000 call resampled zero features (within the rolling window almost every feature had fired at least once at that point), and the step-50,000 tick is never reached because the training loop terminates first.

IV. Preliminary Observations

We report initial results for the single archetypal transcoder of Sec. III-D; the unconstrained top-K baseline and the multi-source / Large-encoder extensions of Sec. A-D are reserved for follow-up work.

A. Prediction quality of the present configuration

On 2×10^6 tokens from the held-out 10% shard split (clip-disjoint from train), the transcoder of Sec. III-D achieves EV $= +0.479$, recon. cosine 0.727, recon. MSE 1.324, L_0 mean 59.07 (p25/p50/p75 = 55/59/64), alive fraction 4.49% (552 of 12,288 features fired), and mean decoder column ℓ_2 norm 41.997.

For comparison, a DINO-B SAE on image features attains EV ≈ 0.99 at $L_0 \approx 500$–1100 under a $64\times$ expansion [16]; reconstruction quality on video features thus remains well below the image-substrate state of the art, our primary quantitative target for future work.

a) Random-initialization control: To isolate what the training objective contributes beyond the architectural prior, we also evaluate a *random-initialization* baseline at the same $d_{\mathrm{hidden}} = 12{,}288$ archetypal-transcoder shape with the *same* K-means pool and the same pool-projection-through-FFN, but with W_e, b_e, A permutation, and Λ at their default random init (Sec. III-C). Held out on the same 2 M tokens, this baseline scores EV $= -1.4 \times 10^4$, $L_0 = 6{,}121$, alive fraction $= 100\%$: the architectural prior alone (frozen pool, bounded Λ, K-sparse simplex on A) does not by itself produce sparse codes or coherent features.

Appendix D compares the top five sweep winners: EV is consistent across the top four ($[0.465, 0.479]$), so we report the best configuration as the headline rather than a lucky single trial.

B. Feature concentration on SSv2 action labels

For each alive feature with at least five labelled top-K hits we compute two histograms over its top-activating clips:

- **Template histogram** counts over the 174 SSv2 motion templates (e.g. *Pulling* [something] *from right to left*). Concentration $c_{\mathrm{template}} = \max \text{ template count} \,/\, n_{\mathrm{labelled}} \in [0, 1]$.
- **Placeholder histogram** counts over the open-vocabulary content tokens drawn from each clip's filled-in label. To avoid surface-form fragmentation (binning *cup*, *plastic cup*, *the cup* into separate bins), we use a per-entry token-presence rule: for each top-K entry we union the non-stop tokens across all of the clip's placeholders, then increment each unique token's count by one for that entry. Concentration $c_{\mathrm{placeholder}} = \max \text{ token count} \,/\, n_{\mathrm{labelled}}$. By construction the count of any token is bounded by n_{labelled}, so multi-placeholder clips do not inflate the denominator.

The two metrics partition each feature along orthogonal axes (what motion is happening / what object is involved), and crossing them at a threshold τ yields the four candidate regimes described next.

We extract per-feature top-10 activating tokens from the held-out evaluation pass (Sec. IV-A). Of $H = 12{,}288$ features, 552 fired at least once and 408 collected ≥ 5 labelled hits, qualifying for histogram analysis. Across those 408, $\bar{c}_{\mathrm{template}} = 0.43$ (median 0.40, max 1.00) and $\bar{c}_{\mathrm{placeholder}} = 0.43$ (median 0.40, max 1.00), roughly $7\times$ the single-bin chance baseline at $K = 10$ over the 174-template support, so the dictionary is concentrating non-trivially on the SSv2 taxonomy on average. The $(c_{\mathrm{template}}, c_{\mathrm{placeholder}})$ scatter (Fig. 3) and its per-axis marginals (Fig. 4) are given in App. B.

a) Candidate quadrant breakdown: Crossing the two concentrations at a heuristic threshold of $\tau = 0.5$ partitions the 408 features into four candidate regimes: **motion-aligned** (high c_t, low c_p): 17; **object-aligned** (low c_t, high c_p): 22; **joint** (both high): 136; **noise / scene** (both low): 233. Counts at $\tau \in \{0.4, 0.6, 0.7\}$ are reported in App. C. The labels are candidate readings rather than confirmed classifications: final calls require inspection of top-clip filmstrips, and the joint column in particular includes features that visual inspection re-classifies as same-clip impostors, camera-motion, or motion-narrow.

C. Example features and tubelets

One example feature illustrates the candidate motion-aligned regime (Fig. 1); four further examples (motion-narrow, object, camera-motion, scene-anchored) are included in Appendix E. Each filmstrip stacks the top-3 unique clips of the feature (one per row); the orange boxes are the (h, w) cells of every top-K hit on that clip, drawn at panel t. Captions are descriptive readings of what the histogram and filmstrip together suggest, not assertions that the feature is the named concept.

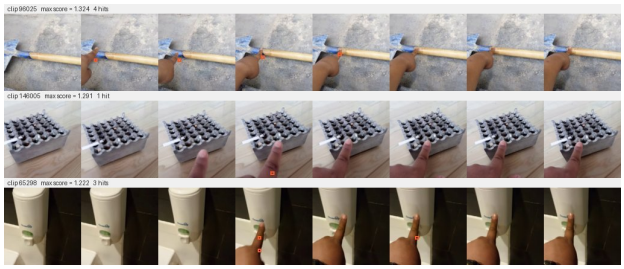

Fig. 1: **Feature 4949, candidate motion-aligned.** $c_{\text{template}} = 1.00$, $c_{\text{placeholder}} = 0.40$, top template = *Poking something so lightly that it doesn't or almost doesn't move*, top placeholder token = *shovel*, 5 unique clips in top-10. All ten top-activating tokens share the poking template while the placeholders span shovel, hand soap container, ashtray, air conditioner, and sock; the orange marks track the poking finger approaching each object, consistent with a motion primitive that generalizes across object identity, though a co-varying scene attribute cannot be ruled out.

Fig. 2: **Feature 4560, a falling-projectile hazard detector** (precision@10 = 0.90). Rows are held-out clips' 8-panel tubelet filmstrips; orange cells are the feature's top-K activations. One feature fires on a thrown-and-falling medicine bottle, knife, and comb—different objects, a single hazard—localized to the airborne object during the throw. It has zero activations on the near-identical *"... without letting it drop"* minimal-pair twin.

D. A first physical-hazard safety probe

Section I motivates archetypal-transcoder features as internal probes for physical-AI safety, by analogy to sparse features that flag out-of-distribution operation in language models [5]. We give a first self-supervised test of that claim on SSv2, reusing only the held-out top-activating clips of Sec. IV-B—no extra training.

a) Method: We label each of the 174 SSv2 templates as a physical *hazard*—an uncontrolled fall, drop, collision, or spill (25 templates, e.g. *Something colliding...*, *... so it falls down*)—or not, and score each feature by its top-10 hazard precision: the fraction of its top-10 activating clips whose template is a hazard. The unconditional hazard rate over all $2.2{\times}10^5$ labelled clips is 0.111, the chance precision of a random detector. We contrast the trained transcoder with the random-initialization control of Sec. IV-A (same architecture and K-means pool, untrained encoder), and exploit SSv2 *minimal pairs* (*... until it falls down* vs. *... without it falling down*) as a selectivity check: a genuine fall detector should fire on the former and not on its near-identical no-fall twin.

b) Result: Of the 408 histogram-eligible features (Sec. IV-B), 17 are hazard-selective (precision ≥ 0.5) and 4 are near-pure (≥ 0.8)—7–9$\times$ the 0.111 base rate. Near-pure hazard features are $5.7\times$ rarer in the random-init control (Poisson $p = 5.7{\times}10^{-3}$; $6.0\times$, $p = 7{\times}10^{-5}$ under a broader 30-template hazard set), so the selectivity is a product of *training*, not of the architecture or of random projections. The four near-pure detectors are a collision feature (feat. 10866, precision 1.00; watch, mouse), a falling-projectile feature (4560, 0.90; knife, medicine bottle, comb, ring), an unsupported-fall feature (1795, 0.90; cup, clothes iron), and a lift-then-drop feature (7978, 0.80; pencil, glass). Their top clips span very different objects (Fig. 2). Qualitative inspection suggests the hazard dynamics, rather than object identity, influence the activation, and none of the four fire on the matched no-fall twin (safe-twin count 0), consistent with detecting the fall dynamics rather than the scene. Because each activation carries its (t, h, w) provenance, the signal is localized in space and time (Fig. 2 marks the airborne object during the throw).

This is a first concrete instance of the safety-monitoring hypothesis of Sec. I: an embodied agent already running V-JEPA exposes a low-dimensional, interpretable, localized feature that fires on collisions and falls, a potential early indicator that could inform runtime monitoring. We are careful about scope: top-10 precision characterizes detector precision on the model's own top activations, not a full ROC over held-out negatives, and the minimal-pair check uses top-K membership; the complete false-alarm characterization—feature activation on matched safe clips—is the immediate next step.

V. Discussion - Open Questions and Limitations

This work establishes the feasibility of training archetypal transcoders on V-JEPA 2, but the ultimate test is utility for physical AI safety. Three measurements are needed: (i) whether these activations predict downstream control failures, (ii) whether atoms inside the convex hull decode to configurations an agent can safely act upon, and (iii) whether the transcoder flags out-of-distribution sensory inputs over long-horizon tasks. Two limitations bound the present claims: the dictionary suffers severe feature collapse (4.49% alive), and the safety probe is observational—showing that safety-relevant behaviour is *causal* requires intervention.

VI. Conclusion

Motivated by the need for interpretable tools to monitor how AI systems represent physical environments, we combine the layer-to-layer prediction objective of transcoders with the convex-hull decoder constraint of archetypal SAEs. On SSv2 we obtain held-out $\text{EV} = +0.479$ at $L_0 = 59$, with candidate motion-aligned features whose filmstrips track a common motion primitive across different objects, and—among them—a statistically enriched sub-population firing selectively on physical hazards (falls, drops, collisions) at up to $9\times$ the base

rate and localized in space and time. This gives a first concrete handle on the safety signal such probes might expose, and groundwork for runtime verification and anomaly detection, though establishing that these features are causally predictive of failure remains open.

VII. Acknowledgements

Special thanks to the Laude Institute for compute.

References

[1] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," *arXiv preprint arXiv:2309.08600*, 2023.

[2] T. Bricken, A. Templeton, J. Batson *et al.*, "Towards monosemanticity: Decomposing language models with dictionary learning," *Transformer Circuits Thread*, 2023.

[3] L. Gao, T. D. la Tour, H. Tillman, G. Goh, R. Troll, A. Radford, I. Sutskever, J. Leike, and J. Wu, "Scaling and evaluating sparse autoencoders," in *International Conference on Learning Representations*, 2025.

[4] T. Fel, E. S. Lubana, J. S. Prince, M. Kowal, V. Boutin, I. Papadimitriou, B. Wang, M. Wattenberg, D. Ba, and T. Konkle, "Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models," in *International Conference on Machine Learning*, 2025.

[5] P. Suresh, J. Stanley, S. Joseph, L. Scimeca, and D. Bzdok, "From noise to narrative: Tracing the origins of hallucinations in transformers," *Advances in Neural Information Processing Systems*, vol. 38, pp. 22 465–22 502, 2026.

[6] J. Dunefsky, P. Chlenski, and N. Nanda, "Transcoders find interpretable LLM feature circuits," *Advances in Neural Information Processing Systems*, 2024.

[7] J. Lindsey, A. Templeton, J. Marcus, T. Conerly, J. Batson, and C. Olah, "Sparse crosscoders for cross-layer features and model diffing," *Transformer Circuits Thread*, 2024.

[8] A. Dokme and S. Vishwanath, "Interpreting video representations with spatio-temporal sparse autoencoders," 2026. [Online]. Available: https://arxiv.org/abs/2604.03919

[9] A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas, "V-JEPA: Latent video prediction for visual representation learning," *arXiv preprint*, 2024.

[10] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Komeili, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, S. Arnaud, A. Gejji, A. Martin, F. Robert Hogan, D. Dugas, P. Bojanowski, V. Khalidov, P. Labatut, F. Massa, M. Szafraniec, K. Krishnakumar, Y. Li, X. Ma, S. Chandar, F. Meier, Y. LeCun, M. Rabbat, and N. Ballas, "V-JEPA 2: Self-supervised video models enable understanding, prediction and planning," *arXiv preprint arXiv:2506.09985*, 2025.

[11] L. Mur-Labadia, M. Muckley, A. Bar, M. Assran, K. Sinha, M. Rabbat, Y. LeCun, N. Ballas, and A. Bardes, "V-JEPA 2.1: Unlocking dense features in video self-supervised learning," *arXiv preprint arXiv:2603.14482*, 2026.

[12] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5842–5850.

[13] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, 1994.

[14] G. Paulo and N. Belrose, "Sparse autoencoders trained on the same data learn different features," *arXiv preprint*, 2025.

[15] G. Gerasimov, T. Rusalev, N. Balagansky, D. Laptev, V. Kurochkin, and D. Gavrilov, "Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders," 2026. [Online]. Available: https://arxiv.org/abs/2606.12138

[16] S. Joseph, P. Suresh, L. Hufe, E. Stevinson, R. Graham, Y. Vadi, D. Bzdok, S. Lapuschkin, L. Sharkey, and B. A. Richards, "Prisma: An open source toolkit for mechanistic interpretability in vision and video," 2025. [Online]. Available: https://arxiv.org/abs/2504.19475

[17] M. Farré, A. Marafioti, L. Tunstall, L. von Werra, and T. Wolf, "FineVideo," https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024.

[18] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The Kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[19] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about Kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

[20] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the Kinetics-700 human action dataset," *arXiv preprint arXiv:1907.06987*, 2019.

Appendix A
Implementation details

A. Archetype-pool construction

Eq. 1 requires a fixed pool C of N centroids that spans the geometry of the FFN's *output* distribution A^{out}, since the decoder writes into that space. We construct C in three steps:

1) Drain $n_{\text{tr}} = 5 \times 10^6$ tokens of A^{in} from the same shuffle stream that training will consume.
2) Project them to A^{out} space via the frozen block-10 FFN in chunks of 65,536, retaining float16 on CPU.
3) Cluster the projected tokens with sklearn's `MiniBatchKMeans` with N clusters, batch size 4,096, ≤ 100 iterations, `n_init=1`, fixed random state.

Step (2) is essential: clustering A^{in} would build a pool in the wrong space, so atoms expressed as convex combinations of those centroids would not span the target distribution. The pool is content-hashed by $(N, n_{\text{tr}}, d, \text{seed}, \text{POOL_SOURCE}=\text{"mlp_out"})$ and cached on disk so sweep trials reuse it rather than repeating the K-means fit, which at $N = 32{,}000$ centroids costs on the order of hours of CPU time.

B. Dead-feature resampling

Following [2] we resample features that go dead during a rolling window. A `DeadFeatureTracker` maintains a per-feature firing counter over a rolling window of $W = 12{,}500$ steps; every $R = 25{,}000$ steps any feature whose counter is zero is resampled. The resampler maintains a $|P| = 4{,}096$ rolling pool of the highest-loss tokens seen since the last resample. For each dead feature i, with q a uniformly sampled high-loss input token:

1) $W_e[i] \leftarrow s \cdot q \,/\, \|q\|_2$, where s is the median row norm of currently-alive encoder rows.
2) $b_e[i] \leftarrow 0$.
3) Project q to A^{out} space via the frozen block-10 FFN, giving $\tilde{q} = \text{FFN}_\ell(q)$, and find the pool centroid $j^\star = \arg\max_j \cos(\tilde{q}, C_j)$. Set $A_{i,j^\star} \leftarrow 5.0$ and $A_{i,j} \leftarrow 0$ for $j \neq j^\star$, so after σ_K the resampled feature is again $\approx$one-hot at the pool atom whose direction best matches the encoder's new selection.
4) $\Lambda_i \leftarrow 0$.
5) Zero the Adam first/second moments for all four parameter rows touched above.

Step (3) is the key archetypal-aware adjustment: the cosine search must be performed in A^{out} space because C lives there. Running the search in A^{in} space cross-compares mismatched

distributions and assigns near-arbitrary pool atoms to dead features; we observed exactly this failure mode at training step ∼50k in an early sweep, where a mistuned resample produced a slow cascade of bad atoms.

C. Sweep protocol

We use Bayesian optimization in W&B with no early termination, so every trial trains to its full step budget of 50,000 steps. Three sweeps were run in sequence:

1) **Focused sweep** (`baby_archetypal_focused`): Bayesian search over $K \in \{32, 64, 128, 256\}$, $\delta \in \{0, 0.25, 0.5\}$, $H = 6{,}144$, modest $\lambda^\star$ range. Used to fix $\delta = 0.5$ and validate that larger K generally wins. Best trial: `feasible-sweep-1`, EV $= +0.39$ at `step_40000` pre-resample.
2) **Pool-32k sweep** (`baby_archetypal_pool32k`): $\lambda^\star \in [10^{-5}, 5 \times 10^{-4}]$, $\eta \in [3 \times 10^{-4}, 3 \times 10^{-3}]$, $H \in \{4{,}608, 6{,}144, 8{,}192\}$, $T_{\mathrm{w}} \in \{2k, 5k, 10k\}$, $K = 512$ fixed. Produced `solar-sweep-1`, `ethereal-sweep-6`, `deft-sweep-2` (EV $\in [0.465, 0.472]$).
3) **Prisma-aligned wide-L_1 / large-H sweep** (`baby_archetypal_pool32k_prisma`): $\lambda^\star \in [10^{-9}, 10^{-5}]$, same η and T_{w} grids, $H \in \{6{,}144, 12{,}288, 24{,}576\}$ matching the upper end of [16]'s DINO-B SAE sweep. The headline configuration `decent-sweep-5` ($H = 12{,}288$, $\lambda^\star = 1.11 \times 10^{-9}$) comes from this sweep.

The all-runs comparison is maintained in `eval_outputs/all_evals_table.md` in the companion code release.

D. Planned extensions reserved for future work

A ViT-Large encoder, four-source training (FineVideo + Kinetics-400/600/700 + SSv2), and a side-by-side comparison against an unconstrained top-K transcoder baseline [3], [6] are configured but not executed in the present submission. The vanilla MLP transcoder (encoder identical to Sec. III-C but with a free decoder constrained only by per-column unit-ℓ_2-norm projection after each step) is implemented and tested under the same training pipeline. The Large encoder requires a one-time re-cache and reuses the entire post-cache pipeline unchanged. The cross-domain study uses identical caches per source and a single re-trained transcoder per source. The intrinsic-dimensionality and code-predictability protocols below operate on either substrate.

E. Evaluation axes

a) Intrinsic dimensionality of V-JEPA 2 features: MLE estimator on held-out token activations at multiple sample sizes, compared against ambient $d = 768$. Provides a quantitative anchor for the manifold-structure motivation. *Not yet run for the present submission.*

b) Code predictability across time: A linear probe $P : \mathbb{R}^k \to \mathbb{R}^k$ to predict z_{t+1} from z_t on held-out clips, where t indexes adjacent tubelet timesteps at the same spatial location. Compares against the same probe applied to raw A^{in} features. Separates the predictive structure of V-JEPA's substrate from the predictive structure preserved by the transcoder's sparse codes.

c) Motion-feature interpretability: For each atom we extract the top-K activating clips on held-out SSv2 [12], join on the SSv2 action-template registry and per-clip placeholder list, build the per-feature template and placeholder histograms of Sec. IV-B, and report concentration / entropy across atoms. Concentration above the dataset prior is direct evidence that the atom selects on a motion primitive or object identity that coincides with a human-annotated category, without invoking a VLM. The body of Sec. IV reports this for the headline configuration; the remaining axes feed into the follow-up.

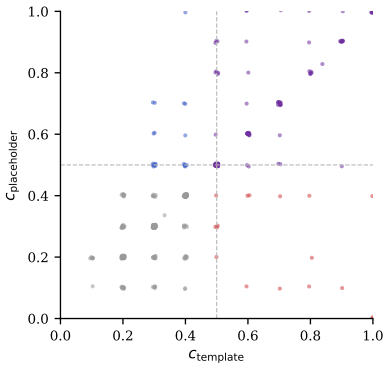

Fig. 3: Per-feature concentration scatter on the $(c_{\mathrm{template}}, c_{\mathrm{placeholder}})$ plane, jittered, across the 408 alive features with ≥ 5 labelled hits. Dots are coloured by candidate quadrant at threshold $\tau = 0.5$: red = candidate motion-aligned (high c_t, low c_p), blue = candidate object-aligned, purple = candidate joint, grey = likely noise / scene. Threshold lines are dashed.

Appendix B
Marginal concentration distributions

Fig. 4 shows the per-axis marginal histograms behind the concentration scatter of Sec. IV-B (Fig. 3). Right-tail mass directly indicates the population of motion-aligned (c_{template}) and object-aligned ($c_{\mathrm{placeholder}}$) candidates respectively.

Appendix C
Candidate quadrant counts at multiple thresholds

Table I reports how the candidate quadrant populations of Sec. IV-B shift with the threshold τ: the motion- and object-aligned counts are stable, while the joint column drains into noise/scene as τ rises.

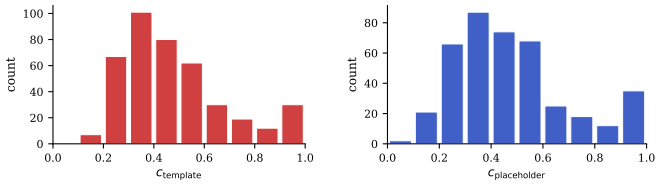

Fig. 4: Marginal distributions of per-feature c_{template} (left) and $c_{\text{placeholder}}$ (right) across the 408 alive features with ≥ 5 labelled hits. c_{template} is the max-bin share of each feature's 174-template histogram across its labelled top-K hits; $c_{\text{placeholder}}$ is the analogous max-bin share over the open-vocabulary content-token support after the per-entry token-presence correction (Sec. IV-B).

TABLE I: Candidate quadrant counts at four thresholds τ, over 408 alive features with ≥ 5 labelled hits. Each cell counts the features whose $(c_{\text{template}}, c_{\text{placeholder}})$ satisfies the quadrant condition. Labels are candidate readings rather than confirmed classifications.

τ	motion	object	joint	noise / scene
0.4	27	26	206	149
0.5	**17**	**22**	**136**	**233**
0.6	17	16	74	301
0.7	13	17	48	330

Appendix D
All locally validated checkpoints

Table II collects held-out metrics for the top five validated checkpoints across the three sweeps of Sec. A-C, plus the random-initialization control and the single late-training instability that motivated the ℓ_2-norm-1 gradient clip of Sec. III-C.

Appendix E
Additional qualitative examples

The following four example features illustrate the *motion-narrow*, *object-aligned*, *camera-motion*, and *scene-anchored* sub-regimes. As in Sec. IV-C, each filmstrip stacks the top-3 unique clips of the feature (one per row) with the (h, w) cells of every top-K hit drawn at panel t.

TABLE II: All locally validated archetypal-transcoder checkpoints, sorted by held-out explained variance (2 M tokens of held-out 10% shard split). Configurations are V-JEPA 2.1 ViT-B/16 layer-10 FFN on SSv2. `feasible-sweep-1 step 50k`† is the post-loop state of `feasible-sweep-1` after a late-training instability (recon MSE drifted from 1.55 at step 40,000 to 3.69×10^6 at step 50,000, EV collapsed to $-3.8{\times}10^4$); the `step_40000` pre-resample checkpoint is the clean comparison. This instability motivated the gradient-clip-at-1.0 rule that all subsequent runs use. The random-initialization row is the same architecture as `decent-sweep-5` with W_e, b_e, A permutation, and Λ at default random init and the *same* K-means pool; the only thing varying between it and the headline is the training objective.

checkpoint	EV ↑	L_0 mean	alive frac
`decent-sweep-5` step 50k	$+\mathbf{0.479}$	59.07	4.49%
`solar-sweep-1` step 50k	$+0.472$	53.63	5.96%
`ethereal-sweep-6` step 50k	$+0.468$	48.50	5.32%
`deft-sweep-2` step 50k	$+0.465$	63.95	6.87%
`feasible-sweep-1` step 40k*	$+0.393$	33.40	4.51%
`random-init` (control)	-1.4×10^4	6,121	100%
`feasible-sweep-1` step 50k†	-3.8×10^4	279.0	7.23%

*pre-resample checkpoint, the clean state. †post-loop state after late-training instability; shown for completeness, not used.

Fig. 5: **Feature 4804, candidate motion-narrow.** $c_{\text{template}} = 1.00$, $c_{\text{placeholder}} = 1.00$, top template = *Letting something roll along a flat surface*, 2 unique clips in top-10. Two visually different objects, a spray paint can and a paint tin, both rolling along a flat surface. $c_{\text{placeholder}}$ saturates because *paint* appears in both clips' labels, but only 2 clips support the feature: the visual reading is rolling-cylindrical-object and the *paint* co-occurrence is more plausibly a dataset-wording artifact than a real object selectivity. By the same-clip-impostor rule, this feature is not eligible for the joint regime; a top-200 evaluation should help resolve which signal drives the firing.

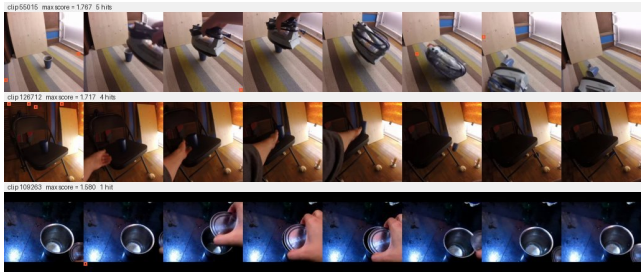

Fig. 6: **Feature 1795, candidate object-aligned.** $c_{\text{template}} = 0.50$, $c_{\text{placeholder}} = 0.90$, top placeholder token = *cup*, top template = *Putting something onto something else that cannot support it so it falls down*, 3 unique clips in top-10. After the per-entry token-presence correction, 9 of 10 hits' clips contain *cup* (binning *cup* and *plastic cup* together). Templates spread across falling-down / pushing-off / pretending-to-close; the cup-ness appears to be doing the selecting, but a confounding scene attribute (kitchen / surface / handheld scale) cannot be ruled out from the histograms alone.

Fig. 7: **Feature 8542, candidate camera-motion.** $c_{\text{template}} = 0.90$, $c_{\text{placeholder}} = 0.50$, top template = *Approaching something with your camera*, top placeholder token = *pendrive*, 4 unique clips in top-10. Joint by the histogram, but the dominant template is a camera-framing class rather than an object-motion class. Marks cluster near $h = 0$ (top of frame) where the approached object first becomes visible. The reading is camera-motion rather than object-motion, illustrating why the joint quadrant cannot be pooled with object-in-motion features without inspecting the dominant template class.

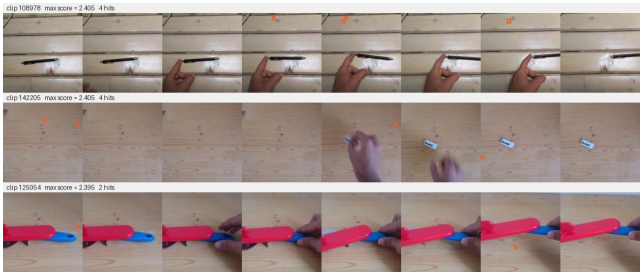

Fig. 8: **Feature 2069, likely scene-anchored noise.** $c_{\text{template}} = c_{\text{placeholder}} = 0.40$. By the histograms this feature lands in the noise / scene quadrant, but the top clips visibly share a wooden-surface background with small handheld objects, a coherent visual primitive that the SSv2 taxonomy does not annotate. We treat it as a scene-anchored candidate rather than confirmed noise; visual inspection is one of the few ways to distinguish these from polysemantic noise at this activation depth.