THE INDEPENDENCE PRIOR OF SAES FRAGMENTS VISUAL CONCEPTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Autoencoders (SAEs) decompose model activations into sparse combinations of interpretable dictionary atoms. Although SAEs are grounded in the Linear Representation Hypothesis (LRH), their objective smuggles in an additional prior: concepts across patches are treated as independent, an assumption clearly violated by natural images and by the activations they induce. We therefore specialize LRH to vision through the Markov-Field Linear Representation Hypothesis (MF-LRH), which adds the missing spatial dependencies to the LRH assumptions. We thus propose SPATIAL-SAE as an amortized MAP estimator under the MF-LRH. SPATIAL-SAE consistently outperforms standard SAEs in concept recovery and interpretability. Across four variants, it achieves a 96% average win rate on synthetic concept recovery and improves interpretability on DINOv2 activations, at a reconstruction cost concentrated in high spatial frequencies.

1 INTRODUCTION

Sparse autoencoders (SAEs) have become the standard tool for extracting human-interpretable concepts from the activations of foundation models (Bricken et al., 2023; Huben et al., 2024; Lieberum et al., 2024; Deng et al., 2026). Applied to vision models, an SAE learns a dictionary of features, each meant to correspond to a visual concept (Fel et al., 2026; Stevens et al., 2025). In practice, however, the features recovered from images' latent representations are often *spatially fragmented*: a single concept is split across several dictionary atoms, its activation map has holes inside the object it should cover, and neighboring patches of the same region are explained by different atoms (Figure 1) (Chanin et al., 2025b; Ayonrinde et al., 2024; Chanin et al., 2025a; Bhalla et al., 2026). Fragmented features are harder to interpret and harder to trust, since the atom that nominally detects a concept fails to fire on parts of it.

Two properties of the SAE objective make fragmentation likely. The first is the rate–distortion–polysemanticity (RDP) tradeoff (Mencattini et al., 2026). At a fixed sparsity budget, the dictionary that best reconstructs activations can be the one whose atoms are polysemantic, so pieces of a concept may reconstruct better than the concept itself. The second, which we address in this paper, is an implicit prior. SAEs are justified by the Linear Representation Hypothesis (LRH), under which each activation is a sparse linear combination of dictionary directions (Elhage et al., 2022; Park et al., 2024; Fel et al., 2026), and SAE training amortizes maximum a posteriori (MAP) inference under this model (Lubana et al., 2026). LRH, however, describes one activation at a time; the SAE objective adds an inductive bias via a prior according to which neighboring patches select their concepts *independently* (Lubana et al., 2026). This assumption does not hold for vision-model activations. Objects, textures, and regions extend across many adjacent patches, and we show that this dependence carries over from the image to the activations of a frozen vision transformer (Figure 2). An objective that cannot express these dependencies has no spatial criterion for preferring a feature that covers a region over one that covers scattered pieces of it, and existing remedies (Fel et al., 2025; Costa et al., 2025) leave this assumption in place.

We replace the independence prior with one that matches vision-model activations. The **Markov-Field Linear Representation Hypothesis (MF-LRH)** keeps every assumption of LRH and adds a single spatial one: the code of a patch is the sum of a *smooth* component, shared with its neighbors

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.

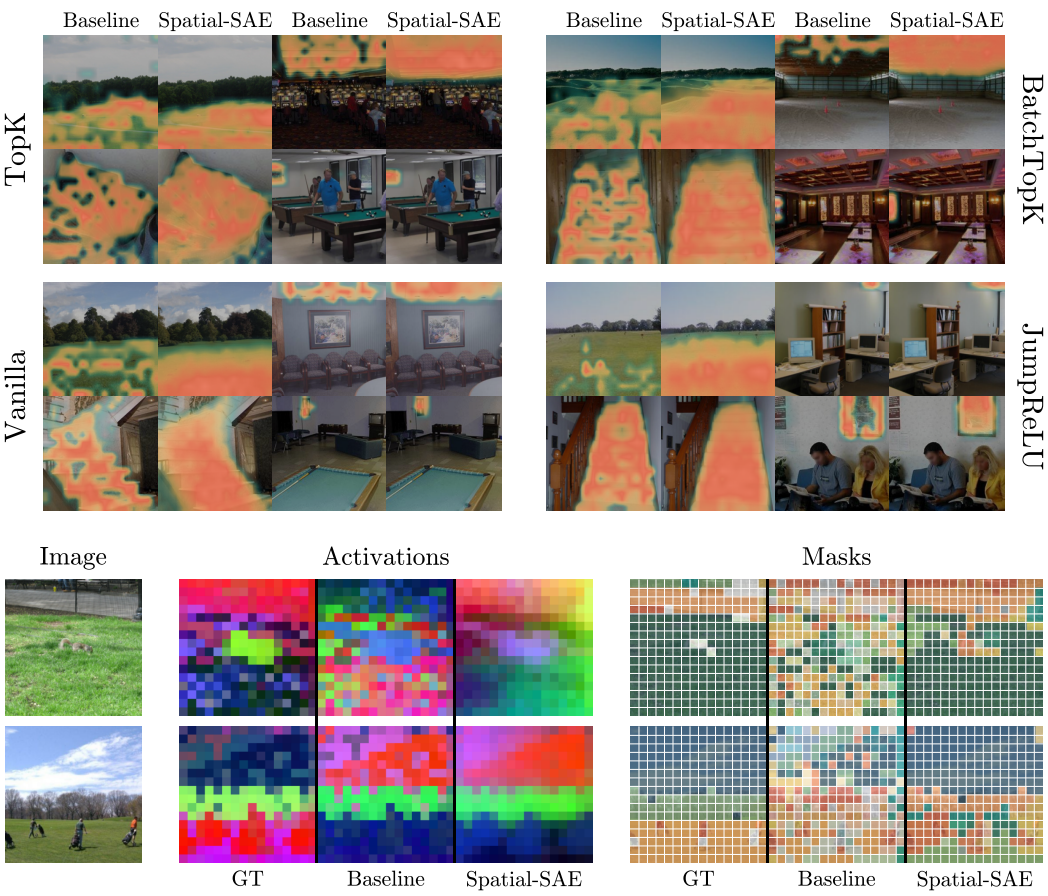

Figure 1: **Spatial organization at three levels. Top:** Baseline and SPATIAL-SAE feature maps across four sparsifiers, illustrating feature support for grass, ceiling, staircase, and painting (reading left to right, then top to bottom). **Bottom:** PCA-RGB views of original and reconstructed activations, alongside ground-truth segments and strongest-atom assignments. Together, these views assess feature support, spatial atom organization, and reconstruction structure.

on the patch grid, and a sparse *innovation*[1] component that carries genuinely local evidence such as boundaries and small objects.

Under MF-LRH, MAP inference is a graph-regularized sparse-coding problem, and amortizing it yields **SPATIAL-SAE**, whose encoder estimates the smooth code of a patch from its neighbors and explains the remainder locally. The spatial prior enters only through the training loss, so SPATIAL-SAE is a drop-in extension of existing SAE families, where fragmented explanations, previously free, now carry a cost. On synthetic data with known dictionaries, concept recovery improves in 288 out of 300 settings; on frozen DINOv2 activations, features become more concept-aligned and region-shaped for all four sparsifiers we test, at a reconstruction cost that is concentrated in the high spatial frequencies where concepts do not live. In practice, Figure 1 shows the outcome: tokenwise SAEs learn incomplete, incoherent, and uninterpretable features whose supports are fragmented across the image; SPATIAL-SAE learns concept-aligned features that are interpretable at a glance.

Contributions. (i) We show that tokenwise SAEs complete LRH with a patch-independence prior, and that vision-model activations violate it (Section 3). **(ii)** We introduce MF-LRH, a spatial specialization of LRH, and derive SPATIAL-SAE as amortized MAP inference under it (Section 4). **(iii)** We show that SPATIAL-SAE recovers the ground-truth concepts of synthetic data more reliably than standard SAEs, and learns more concept-aligned features on real DINOv2 activations (Section 5).

[1]We borrow the term *innovation* from the state-space and Kalman-filtering literature, where it denotes the part of an observation that cannot be predicted from past observations (Kailath, 1968; Kalman, 1960).

2 BACKGROUND

Notation. Matrices are denoted by uppercase bold symbols ($\mathbf{X}$), vectors by lowercase bold symbols ($\mathbf{x}$), scalars by unbolded letters (x), and sets by uppercase calligraphic symbols ($\mathcal{X}$).

Linear Representation Hypothesis. The Linear Representation Hypothesis (LRH) models foundation model activations as sparse linear combinations of directions in an overcomplete, approximately non-redundant dictionary (Elhage et al., 2022; Park et al., 2024; Jiang et al., 2024; Fel et al., 2026). Following Costa et al. (2025); Fel et al. (2026), one can define it formally:

Definition 2.1 (Linear Representation Hypothesis)**.** The activation $\mathbf{h} \in \mathbb{R}^d$ satisfies LRH if there exists a dictionary $\mathbf{D} \in \mathbb{R}^{c \times d}$ and a sparse coefficient vector $\mathbf{z} \in \mathbb{R}^c$ such that $\mathbf{h} \approx \mathbf{z}\mathbf{D}$ and

$$\begin{cases} \text{(i) Overcompleteness:} & c \gg d, \\ \text{(ii) Quasi-orthogonality:} & \mu(\mathbf{D}) := \max\limits_{a \neq b} \left|\mathbf{d}_a \mathbf{d}_b^\top\right| \leq \varepsilon, \quad \|\mathbf{d}_a\|_2 = 1 \quad \forall a, \\ \text{(iii) Sparsity:} & |\mathrm{supp}(\mathbf{z})| \leq k \ll c. \end{cases} \tag{1}$$

Applied row-wise, this gives $\mathbf{H} \approx \mathbf{Z}\mathbf{D}$. Concept extraction under LRH is therefore an inverse problem: recover the dictionary directions $\mathbf{D}$ and the sparse activation matrix $\mathbf{Z}$ (Hindupur et al., 2025).

Sparse coding and SAEs. Sparse coding addresses this inverse problem by jointly optimizing the dictionary and one sparse code per activation (Olshausen & Field, 1996; 1997; Elad, 2010):

$$(\mathbf{Z}^\star, \mathbf{D}^\star) \in \arg\min_{\mathbf{Z},\mathbf{D}} \frac{1}{n} \sum_{i=1}^{n} \left[\|\mathbf{h}_i - \mathbf{z}_i \mathbf{D}\|_2^2 + \lambda \mathcal{R}(\mathbf{z}_i) \right]. \tag{2}$$

where $\mathcal{R}$ promotes sparsity, for example $\mathcal{R}(\mathbf{z}) = \|\mathbf{z}\|_1$. Solving Equation 2 requires a separate optimization for every activation. An SAE amortizes this cost by training a shared parametric encoder $f_{\boldsymbol{\theta}}$ to predict $\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i)$ in a forward pass (Bricken et al., 2023; Huben et al., 2024). The encoder replaces repeated optimization with a forward pass, but may return a suboptimal sparse code; its approximation cost is measured by the amortization gap (O'Neill et al., 2025).

3 SPATIAL STRUCTURE IN VISION-MODEL ACTIVATIONS

The LRH specifies how each activation is composed, but not how the codes of different patches should relate. In Section 3.1, we identify the patch-independence prior implicit in tokenwise SAEs, examine its consequences for dictionary learning, and contrast it with the spatial dependencies observed in vision-model activations. In Section 3.2, we introduce MF-LRH, specializing LRH with a spatial prior over complementary smooth and innovation fields.

3.1 SAES ADD A PATCH-INDEPENDENCE PRIOR

The LRH calls for sparsity and overcompleteness, but the SAE objective also encodes an implicit prior of independence between tokens. Lubana et al. (2026) make this prior explicit through a probabilistic interpretation of SAE training. They show that, under a Gaussian noise model, minimizing the SAE objective is equivalent to finding the *maximum a posteriori (MAP) estimate of the latent codes* under a sparsity-favoring prior, $p(\mathbf{Z}) \propto \exp(-\lambda \mathcal{R}(\mathbf{Z}))$ (e.g., $\mathcal{R}$ could be the ℓ_1 norm). Crucially, the sparsity penalty in Equation 2 applies separately to each token: $\mathcal{R}(\mathbf{Z}) = \sum_i \mathcal{R}(\mathbf{z}_i)$. The corresponding prior therefore factorizes as $p(\mathbf{Z}) = \prod_{i=1}^{p} p(\mathbf{z}_i)$, treating different tokens' latent codes as independent (Proposition B.6). Thus, beyond encouraging accurate reconstruction and sparse representations, the objective *imposes a prior that does not account for dependencies across tokens*. Lubana et al. (2026) argue that this assumption is detrimental for language models, since an i.i.d. prior over tokens ignores their temporal structure.

Similarly, we argue that this prior is mismatched to dictionary learning on vision models. Although activations may already encode context, the tokenwise SAE objective contains no explicit interaction between neighboring patch codes. For a fixed encoder and dictionary, permuting the rows of $\mathbf{H}$ before encoding simply permutes $\mathbf{Z}$: each activation selects the same atoms and incurs the same

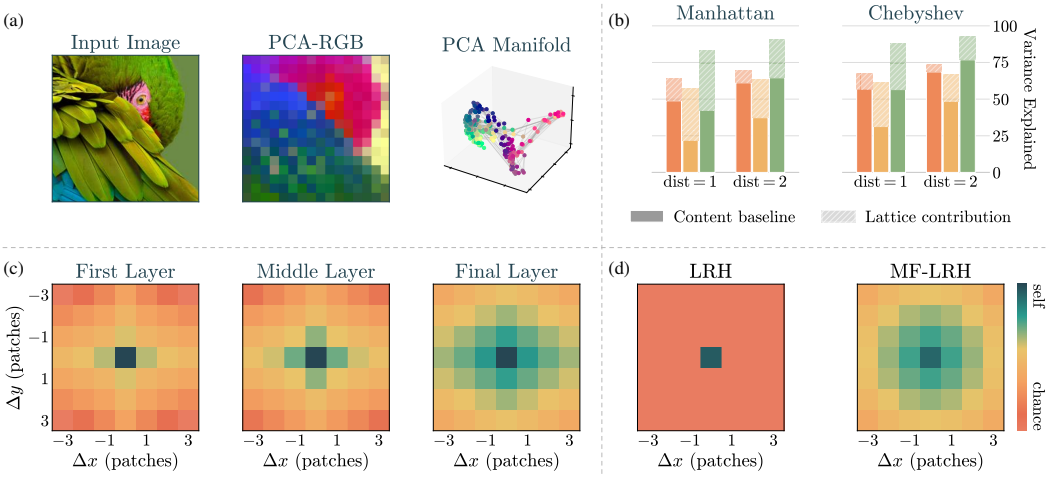

Figure 2: **Patch activations retain spatial structure.** (a) PCA-RGB and 3D PCA views of DINOv2 activations show spatial organization. (b) Variance explained by Manhattan and Chebyshev neighborhoods. Solid bars are within-image shuffled controls; hatching is the additional contribution of the true layout. (c) Measured offset-similarity kernels across depth. (d) Measured offset-similarity kernels from an LRH model and the MF-LRH model used in our synthetic experiments.

cost. The objective therefore assigns no additional cost based on whether a patch's explanation fits with its neighborhood. Thus, the independence prior supplies no criterion for preferring explanations that fit together spatially, making it a mismatched inductive prior for obtaining spatially consistent explanations of vision activations.

Proposition 3.1 (Permutation invariance of the tokenwise objective)**.** *Let* $\mathbf{P} \in \mathbb{R}^{p\times p}$ *be any permutation matrix. For the row-additive tokenwise* SAE *objective defined above,* $\mathcal{L}_{\text{SAE}}(\mathbf{PZ}, \mathbf{D}\,; \mathbf{PH}) = \mathcal{L}_{\text{SAE}}(\mathbf{Z}, \mathbf{D}; \mathbf{H})$. *Consequently, a tokenwise* SAE *objective does not distinguish a coherent patch layout from an arbitrarily permuted layout, provided the activations and codes are permuted together.*

The proof is given in Appendix B.2. To see why this invariance may matter, consider how recognizing a tree helps us interpret a nearby green region as foliage rather than grass: the interpretations of the parts support one another (Wagemans et al., 2012; Biederman, 1972; Palmer, 1975). Just as we interpret image regions, an SAE interprets their activation vectors through dictionary features. If these activations retain spatial dependencies, the features explaining neighboring activations should inform those selected for the current one. Although each activation already carries context, the objective does not coordinate these choices. We next show that these dependencies persist in vision-model activations, motivating replacing the mismatched independence prior implied by SAEs.

Patch activations are spatially dependent. Figure 2 provides three complementary pieces of evidence against independence: **(a) The visualizations reveal recognizable spatial patterns**. Coloring patches by their activations reveals coherent image regions. Even after averaging activations across images at each patch position, their PCA projection still resembles the image grid. **(b)** True neighbors explain more of a patch's activation than equally sized neighborhoods of randomly chosen patches from the same image. The shuffled neighborhoods measure what shared image content alone explains, and the *lattice contribution* is the additional variance explained only when neighbors are taken from the true grid layout. This gap shows that spatial proximity provides information beyond shared image content. **(c) Nearby patches remain more similar than distant ones** from the first to the final layer, showing that this spatial structure persists throughout the network. These observations agree with prior work showing that self-supervised ViT features organize around object regions (Caron et al., 2021) and that ViTs trained by gradient descent provably learn to group spatially related patches through their positional attention (Jelassi et al., 2022). Thus, neighboring patch activations provide predictive evidence about one another that a factorized prior cannot exploit. Next, we introduce an extension of LRH that incorporates this dependence while preserving a separate channel for genuinely local variation.

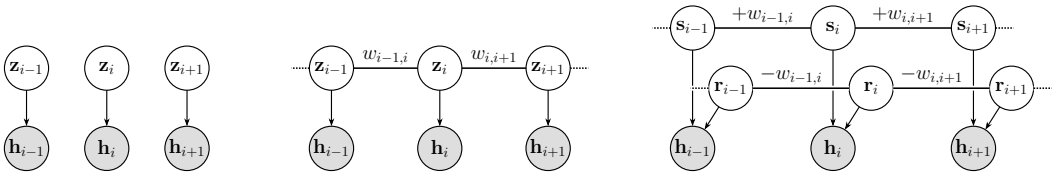

Figure 3: **From independent codes to attractive–repulsive coupling.** *Left:* LRH+SAE treats patch codes $\mathbf{z}_i$ independently. *Middle:* attractive coupling of the full codes promotes spatial persistence but can oversmooth local departures. *Right:* MF-LRH decomposes each code as $\mathbf{z}_i = \mathbf{s}_i + \mathbf{r}_i$, coupling neighboring smooth codes attractively and neighboring innovation codes repulsively.

3.2 Markov-Field Linear Representation Hypothesis

To replace the independence prior, we need a model of how patch codes depend on each other. In images, this dependence is mostly local: an object or region typically spans several adjacent patches, so a concept present at one patch is likely present at its neighbors. Markov random fields (MRFs) have long formalized such local interactions in classical computer vision (Prince, 2012), and they combine naturally with sparsity: an MRF prior can favor sparse supports that cluster spatially over equally sparse but scattered ones (Cevher et al., 2008).

We therefore introduce the **Markov-Field Linear Representation Hypothesis (MF-LRH)**, which splits each patch's code into two complementary fields (Figure 3). Natural images have stronger low-frequency components (Field, 1987), motivating a spatially coupled **smooth field** $\mathbf{S}$ for content shared across patches. Fine details, however, need not persist spatially, motivating an **innovation field** $\mathbf{R}$ for local departures. We combine them additively as $\mathbf{Z} = \mathbf{S} + \mathbf{R}$, both for simplicity and to mirror the structure of the residual stream.

Definition 3.2 (Markov-Field Linear Representation Hypothesis)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ be a patch graph over the p image patches with symmetric nonnegative weights w_{ij}, and let $\mathbf{L} \in \mathbb{R}^{p\times p}$ be its graph Laplacian. A patch-activation matrix $\mathbf{H} \in \mathbb{R}^{p\times d}$ satisfies MF-LRH if there exist a dictionary $\mathbf{D} = (\mathbf{d}_1, \ldots, \mathbf{d}_c)^\top \in \mathbb{R}^{c\times d}$, latent fields $\mathbf{S}, \mathbf{R} \in \mathbb{R}^{p\times c}$, and $\mathbf{Z} = \mathbf{S} + \mathbf{R}$ such that (i) the LRH conditions (Definition 2.1) hold, (ii) $\mathbf{H} \approx (\mathbf{S} + \mathbf{R})\mathbf{D}$, and (iii) $\mathbf{S}$ and $\mathbf{R}$ are respectively attractive and repulsive pairwise MRFs on $\mathcal{G}$. For inference, we relax hard sparsity and use the following continuous prior:

$$p(\mathbf{S}, \mathbf{R} \mid \mathcal{G}) \propto \exp\left\{ -\lambda_S \|\mathbf{S}\|_1 - \lambda_R \|\mathbf{R}\|_1 - \frac{\mu}{2}\left[\kappa_{\mathcal{G}}\|\mathbf{R}\|_{\mathrm{F}}^2 + \sum_{\{i,j\}\in\mathcal{E}} w_{ij}\left(\|\mathbf{s}_i - \mathbf{s}_j\|_2^2 - \|\mathbf{r}_i - \mathbf{r}_j\|_2^2\right)\right]\right\}, \tag{3}$$

where $\kappa_{\mathcal{G}} = \lambda_{\max}(\mathbf{L})$, $\lambda_S, \lambda_R > 0$, and $\mu \geq 0$.

Conditions (i)–(ii) retain the sparse linear representation of LRH while decomposing its code into smooth and innovation fields. Condition (iii) replaces patch independence with complementary local dependencies: neighboring smooth codes favor agreement, whereas neighboring innovation codes favor contrast (Proposition B.8). In Equation (3), the ℓ_1 terms promote sparsity, while the edge terms realize these attractive–repulsive interactions, and the magnitude term $\kappa_{\mathcal{G}}\|\mathbf{R}\|_{\mathrm{F}}^2$ keeps the repulsive quadratic bounded below (Corollary A.4). This reflects the intended role of $\mathbf{R}$ as the local variation left outside the spatially persistent component, consistent with our observations in Figure 2b.

At $\mu = 0$, the spatial prior disappears and the model factorizes across patches; ordinary sparse coding is recovered by additionally taking $\mathbf{S} \equiv \mathbf{0}$ and $\mathbf{R} = \mathbf{Z}$. Thus, MF-LRH retains LRH while replacing its implicit patch-independence completion with an attractive–repulsive Markov prior.

4 Spatial Sparse Autoencoders

Having motivated MF-LRH as a model of spatial dependencies between patch codes, we now develop a method that incorporates this prior. In Section 4.1, we extend the Bayesian interpretation of tokenwise SAEs to derive SPATIAL-SAE from MAP inference under MF-LRH. In Section 4.2, we describe the encoder architecture of SPATIAL-SAE that makes this objective usable in practice, when the precise decomposition of $\mathbf{Z}$ into $(\mathbf{S}, \mathbf{R})$ is not known.

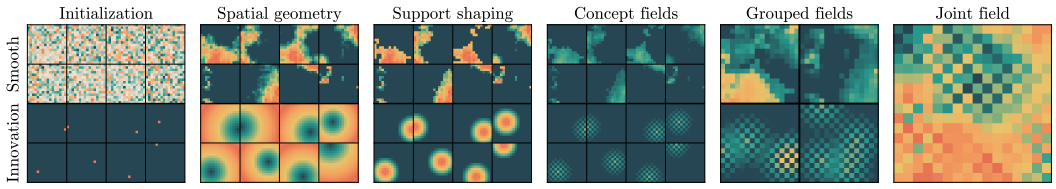

Figure 4: **Generating synthetic data.** From left to right: initialization (Gaussian noise / sampled center), spatial geometry (graph diffusion and rectification / distance from center), support shaping (quantile thresholding / compact radial envelope), concept construction (peak normalization / bipartite masking), grouped aggregation, and the final joint field $\mathbf{S} + \mathbf{R}$. In each parenthetical pair, the smooth-stream operation is listed first and the innovation-stream operation second.

4.1 Spatial-SAE Is Amortized MAP Inference Under MF-LRH

To turn MF-LRH into a practical method, we first need an objective for inferring its latent codes. We derive it through the Bayesian approach of Section 3, keeping the observation model unchanged and adopting the spatial prior of Definition 3.2. The resulting MAP objective combines reconstruction and sparsity with two additional terms that encourage spatial agreement in the smooth field and spatial disagreement in the innovation field.

Theorem 4.1 (Spatial sparse coding is MAP inference for MF-LRH)**.** *Let* $\mathbf{H} = (\mathbf{S} + \mathbf{R})\mathbf{D} + \mathbf{\Psi}$ *with* $\mathbf{\Psi}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$ *independent across patches, and* $(\mathbf{S}, \mathbf{R})$ *following the* MF-LRH *prior of Definition 3.2. For fixed* $\mathbf{D}$ *and an externally specified graph* $\mathcal{G}$ *with Laplacian* $\mathbf{L}$*, let* $\widetilde{\mathbf{L}} = \kappa_{\mathcal{G}} \mathbf{I}_p - \mathbf{L}$*. Absorbing* σ^2 *into the weights, a* MAP *estimate of* $(\mathbf{S}, \mathbf{R})$ *solves the spatial sparse-coding problem:*

$$(\mathbf{S}^\star, \mathbf{R}^\star) \in \arg\min_{\mathbf{S},\mathbf{R}} \tfrac{1}{2} \|\mathbf{H} - (\mathbf{S} + \mathbf{R})\mathbf{D}\|_F^2 + \lambda_S \|\mathbf{S}\|_1 + \lambda_R \|\mathbf{R}\|_1 + \tfrac{\mu}{2} \left[\mathrm{Tr}(\mathbf{S}^\top \mathbf{L}\mathbf{S}) + \mathrm{Tr}(\mathbf{R}^\top \widetilde{\mathbf{L}}\mathbf{R}) \right]. \tag{4}$$

The proof is in Appendix B.4. The trace identities in Lemma A.3 convert the prior's attractive–repulsive interactions into the two quadratic terms in Equation (4). The local dependencies and innovation magnitude penalty encoded by the prior yield complementary spectral regularization on the patch graph. The smooth term penalizes variation across graph edges, suppressing high-frequency components and promoting agreement between neighboring codes. Conversely, the innovation term uses the positive-semidefinite operator $\widetilde{\mathbf{L}}$ to suppress low-frequency components and promote spatial contrast. This objective connects to graph-regularized sparse coding (Yankelevsky & Elad, 2020); related sparsity–smoothness tradeoffs appear in compressed manifold modes (Neumann et al., 2014). Finally, we can define the Spatial-SAE:

Definition 4.2 (Spatial Sparse Autoencoder)**.** Let $\mathcal{X} = \{\mathbf{H}^{(n)} \in \mathbb{R}^{p \times d}\}_{n=1}^N$ be a dataset of patch activations from N images and $\mathbf{L} \in \mathbb{R}^{p \times p}$ be a fixed patch-graph Laplacian and $\widetilde{\mathbf{L}} = \kappa_{\mathcal{G}} \mathbf{I}_p - \mathbf{L}$. A Spatial-SAE pairs an encoder $F_{\boldsymbol{\theta}}$ predicting both fields, $(\mathbf{S}_{\boldsymbol{\theta}}, \mathbf{R}_{\boldsymbol{\theta}}) = F_{\boldsymbol{\theta}}(\mathbf{H}; \mathbf{L})$, with a decoder dictionary $\mathbf{D} \in \mathbb{R}^{c \times d}$ reconstructing $\widehat{\mathbf{H}} = (\mathbf{S}_{\boldsymbol{\theta}} + \mathbf{R}_{\boldsymbol{\theta}})\mathbf{D}$, and is trained by:

$$\min_{\boldsymbol{\theta},\mathbf{D}} \frac{1}{|\mathcal{X}|} \sum_{\mathbf{H} \in \mathcal{X}} \left[\tfrac{1}{2} \|\mathbf{H} - (\mathbf{S}_{\boldsymbol{\theta}} + \mathbf{R}_{\boldsymbol{\theta}})\mathbf{D}\|_F^2 + \lambda_S \|\mathbf{S}_{\boldsymbol{\theta}}\|_1 + \lambda_R \|\mathbf{R}_{\boldsymbol{\theta}}\|_1 + \tfrac{\mu}{2} \Big(\mathrm{Tr}\big(\mathbf{S}_{\boldsymbol{\theta}}^\top \mathbf{L}\mathbf{S}_{\boldsymbol{\theta}}\big) + \mathrm{Tr}\big(\mathbf{R}_{\boldsymbol{\theta}}^\top \widetilde{\mathbf{L}}\mathbf{R}_{\boldsymbol{\theta}}\big) \Big) \right]. \tag{5}$$

Like ordinary sparse coding, Equation 4 requires solving an optimization problem for each image. Spatial-SAE replaces this with a learned encoder, amortizing MAP inference under MF-LRH just as a vanilla SAE amortizes ordinary sparse coding. Furthermore, the spatial prior enters through the training loss, making Spatial-SAE a *plug-and-play extension* compatible with the main SAE families (details in Appendix D.1). Lastly, we prove that *the graph terms break the permutation invariance of tokenwise objectives* (Proposition 3.1) while preserving invariance to patch indexing. With the graph fixed, the remaining permutation symmetries are those that preserve weighted adjacency, as formalized in Proposition B.9. The MF-LRH prior thus encodes the spatial dependencies documented in Section 3 but ignored by the patch-independent SAE prior: the cost of explaining a patch now depends on how its neighbors are explained. The graph terms encourage agreement between neighboring smooth codes and disagreement between neighboring innovation codes. We next describe how the encoder estimates the smooth and innovation fields from real-world activations.

4.2 Estimating the Smooth and Innovation Fields

In real-world models, we observe only the activations $\mathbf{H}$. We therefore need an expressive encoder that infers the smooth and innovation fields $(\mathbf{S}, \mathbf{R})$ from $\mathbf{H}$ alone. Our approach follows a simple idea: use neighboring patches to estimate the shared content, then explain what remains locally. We encode the neighbors' mean activation as an attention query, summarizing the local context. Attention uses this query to weight and combine the neighboring activations into the smooth code $\mathbf{S}_i$, without seeing the target patch. A second encoder estimates the innovation code $\mathbf{R}_i$ from the residual of the smooth reconstruction. This builds on the spatial predictability observed in Figure 2; Algorithm 1 summarizes the procedure. In practice, we partition the dictionary between the smooth and innovation streams, each supported only on its assigned atoms. Algorithm 1 uses the corresponding active code blocks and sub-dictionaries $\mathbf{D}_S$ and $\mathbf{D}_R$. We compute the residual as $\mathbf{H}_i - \mathrm{sg}(\mathbf{S}_i\mathbf{D}_S)$, where the stop-gradient prevents the innovation stream from backpropagating through the smooth reconstruction.

Algorithm 1: SPATIAL-SAE's forward pass.

Require: Patch activations $\mathbf{H} \in \mathbb{R}^{p \times d}$.

for each patch i **do**
  $\mathbf{N} \leftarrow \textsc{Stack}(\{\mathbf{H}_j : (i,j) \in \mathcal{E}\})$
  $\mathbf{q} \leftarrow \overline{\mathbf{N}}\mathbf{E}_S, \quad \mathbf{KV} \leftarrow \mathbf{N}\mathbf{E}_S$
  $\mathbf{S}_i \leftarrow \textsc{Sparsify}(\mathrm{Att}(\mathbf{q}, \mathbf{KV}, \mathbf{KV}))$
  $\mathbf{R}_i \leftarrow \textsc{Sparsify}((\mathbf{H}_i - \mathbf{S}_i\mathbf{D}_S)\mathbf{E}_R)$
  $\widehat{\mathbf{H}}_i \leftarrow \mathbf{S}_i\mathbf{D}_S + \mathbf{R}_i\mathbf{D}_R$
end for
return $\mathbf{S}$, $\mathbf{R}$, and $\widehat{\mathbf{H}}$

5 Experiments

We now test whether a spatial prior improves the features obtained by SAEs. Before presenting the experiments, we need to be clear about what should be evaluated, because the frequently used criterion, reconstruction (Fel et al., 2026; Costa et al., 2025), is not the right one (Chanin et al., 2025b;a). An SAE communicates each activation through a sparse bottleneck, and ideally the resulting code is sparse, faithful, and interpretable at once; standard objectives, however, optimize only the first two and never reward monosemanticity directly. Mencattini et al. (2026) formalize this tension as the **rate–distortion–polysemanticity (RDP) tradeoff**: at a fixed sparsity budget, better reconstruction can be achieved with more polysemantic dictionaries, and thus, it does not necessarily mean a better dictionary. Since our goal is to recover interpretable concepts rather than minimize reconstruction alone, we treat interpretability as the primary metric and report reconstruction as a secondary tradeoff. Under RDP, a loss in reconstruction when improving interpretability is therefore expected.

We do so in two settings. In Section 5.1, we use synthetic data with known dictionaries and codes to measure polysemanticity (Mencattini et al., 2026) and concept recovery under spatial structure. Crucially, data is not trivially sampled from the MF-LRH prior itself, but from a spatially structured generator intended to mimic the smooth and local variation we observed in vision activations (Figure 2). In Section 5.2, we test whether the same inductive bias helps on frozen DINOv2 activations. In both settings and for all four sparsifiers we test, SPATIAL-SAE learns more interpretable dictionaries and more coherent feature maps, at a modest reconstruction cost.

5.1 Concept Recovery on Synthetic Data

To measure concept recovery directly, we generate data whose concepts are known and compare SPATIAL-SAE with a standard SAE on them.

Synthetic spatial benchmark. We add spatial structure to existing sparse linear benchmarks (Chanin & Garriga-Alonso, 2026) by constructing two nonnegative latent streams, $\mathbf{S}$ and $\mathbf{R}$, on a 16×16 grid, both associated with a shared dictionary $\mathbf{D}$. Each stream is formed by normalizing and aggregating concept-specific subfields, but with different spatial geometry: smooth subfields are obtained by diffusing, rectifying, and thresholding Gaussian noise, whereas innovation subfields are compact radial envelopes centered at sampled locations and restricted to alternating grid sites (see Figure 4 for an overview). Our data-generating process deliberately does not sample exactly from the MF-LRH prior, since in that setting an advantage for the corresponding MAP estimator would be unsurprising. Instead, we test whether the prior's inductive bias remains useful under misspecification, using data that retain the qualitative spatial structure observed in real concept fields: concepts tend to persist across neighboring patches, while localized innovations capture patch-specific deviations. Appendix C.1 gives the full generating process and its parameters.

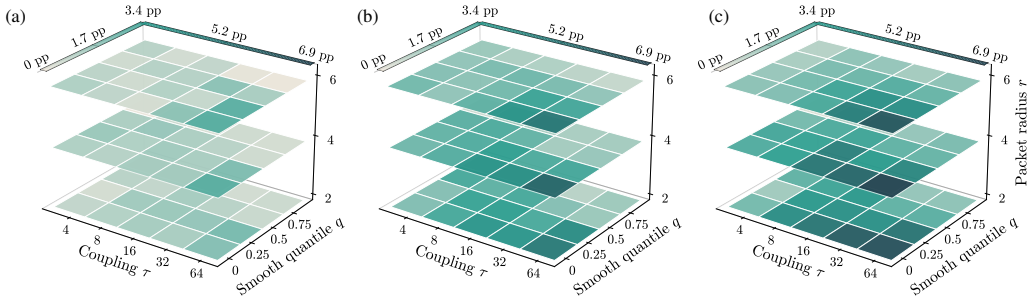

Figure 5: **Isolating the prior's effect on synthetic data.** We compare spectral regularization at (a) $\mu = 3 \times 10^{-4}$, (b) $\mu = 10^{-3}$, and (c) $\mu = 3 \times 10^{-3}$ with the same two-stream model at $\mu = 0$. Color intensity shows F1 gain; the prior helps across **all** smoothness and innovation settings.

Main experimental setting. Since the dictionary and codes are known, we can directly evaluate concept recovery. We measure MCC, support F1, and held-out R^2 following Chanin & Garriga-Alonso (2026), and polysemanticity (POLY.) following Mencattini et al. (2026). Our main experiment studies how SPATIAL-SAE behaves as we vary the DGP parameters governing the smoothness and innovation structure of the generated concept fields. We perform a grid sweep over these parameters, yielding 300 matched runs per architecture, and report the win rate of SPATIAL-SAE over the corresponding baseline in Table 1. Appendix C reports additional sweeps over the generating-dictionary geometry and sparsity budget. It also ablates the two components of SPATIAL-SAE, the smooth–innovation decomposition and spectral regularization, to quantify the contribution of each.

Results. For ReLU, TopK, and BatchTopK, we observe the rate–distortion–polysemanticity tradeoff predicted by Mencattini et al. (2026): SPATIAL-SAE improves concept recovery and reduces polysemanticity while conceding reconstruction. It wins every MCC, F1, and polysemanticity comparison for TopK and BatchTopK, and wins 85.7%, 85.0%, and 87.7% of the corresponding ReLU comparisons, respectively, while never winning on R^2. JumpReLU is the notable exception: SPATIAL-SAE wins every F1 and R^2 comparison, and 98.7% of MCC and 96.0% of polysemanticity comparisons. Thus, the spatial prior need not inherently trade reconstruction for concept recovery; with a compatible sparsifier, it can move the model to a better region of the RDP frontier. These gains are not explained by the two-stream decomposition alone; relative to an otherwise identical unregularized two-stream model, the spatial prior strictly increases F1 across the same parameter sweep and, for an appropriately tuned μ, SPATIAL-SAE improves up to 6.9 percentage points (Figure 5).

Sparsifier	MCC	POLY.	F1	R^2
TopK	100%	100%	100%	0%
BatchTopK	100%	100%	100%	0%
ReLU	85.7%	87.7%	85.0%	0%
JumpReLU	98.7%	96.0%	100%	100%

Table 1: Win rate of SPATIAL-SAE against baseline SAEs on matched pairs (same generator configuration and seed); SPATIAL-SAE wins the three identification metrics, conceding reconstruction.

5.2 CONCEPT DISCOVERY IN VISION-MODEL ACTIVATIONS

Section 5.1 tests SPATIAL-SAE on synthetic data containing both smooth and local concepts. We next turn to real vision-model activations, which need not follow MF-LRH exactly and for which neither the ground-truth dictionary nor the latent codes are observed.

Setup and evaluation. We compare SPATIAL-SAE against standard SAEs on frozen DINOv2 ViT-B/14 activations from ADE20K across four sparsifiers with matched dictionary widths and sparsity budgets. We use dense annotations to measure semantic and spatial alignment, and a blinded autointerpretability protocol to assess feature interpretability. For each class, we identify its best detector and normalize its F1 by that of the corresponding linear probe (*Normalized F1*); we measure spatial alignment by comparing within- and across-segment code similarity at matched distances (*Region Coherence*); and we assess feature interpretability through blinded LLM ratings of activation sup-

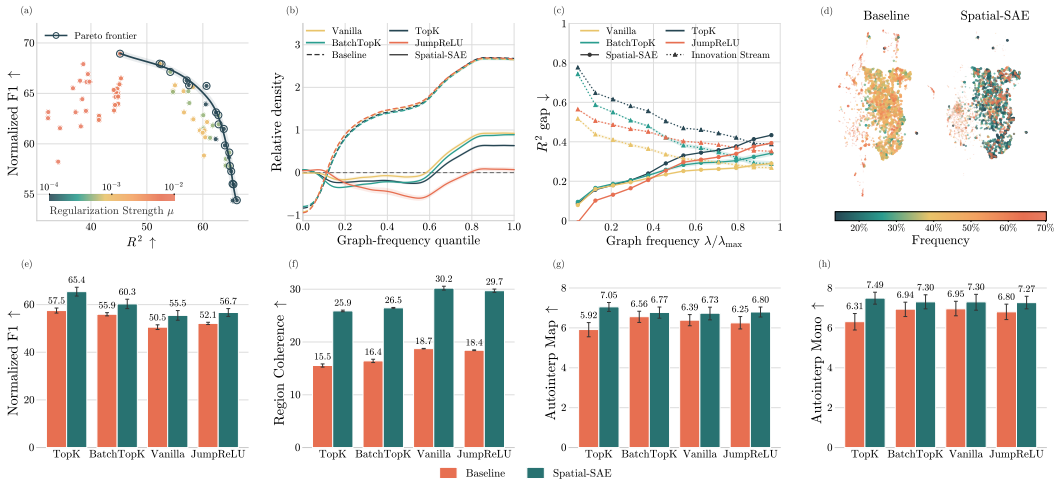

Figure 6: **Concept alignment and spectral organization.** (a) Normalized F1–R^2 frontier over SPATIAL-SAE configurations. (b) Code spectrum relative to concept masks: zero denotes agreement with the masks, dashed curves the baseline, and solid curves SPATIAL-SAE. (c) The baseline's R^2 advantage over SPATIAL-SAE (solid) and its innovation stream alone (dotted), by frequency. (d) TopK decoder atoms in a joint UMAP, colored by the graph frequency of their activation maps. (e–h) Normalized F1, Region Coherence, and automated map-quality and monosemanticity ratings. Error bars show one stddev over 5 seeds in (e–f) and 95% class-bootstrap intervals from one seed in (g–h).

port (*Map Quality*) and consistency across images (*Monosemanticity*). Complete training details, general evaluation protocols, and additional COCO-STUFF results appear in Appendix D.

Concept alignment and spatial organization. SPATIAL-SAE improves all interpretability measures for all sparsifiers (Figure 6e–h). For TopK, Normalized F1 increases from 57.5 to 65.4 and Region Coherence from 15.5 to 25.9, while Map Quality and Monosemanticity improve for every sparsifier. These gains trade off against reconstruction, exactly as RDP predicts; varying the regularization strength traces a clear Normalized-F1–R^2 frontier rather than producing an isolated operating point (Figure 6a). The spectral analysis then connects this tradeoff to fragmentation: low graph frequencies encode concepts that persist across neighboring patches, yet every baseline underrepresents them, instead increasingly overrepresenting higher frequencies. Conversely, SPATIAL-SAE brings the code spectrum substantially closer to the concept mask spectrum (Figure 6b). The reconstruction tradeoff is concentrated in the same high-frequency regime: the baseline's R^2 advantage over the combined SPATIAL-SAE code grows from 0.11 at low frequencies to 0.37 at high frequencies, where its spectral excess is largest. Meanwhile, its advantage over the innovation stream alone narrows, consistent with that stream specializing in local, high-frequency variation (Figure 6c). At the feature level, SPATIAL-SAE contains distinct low- and high-frequency populations, whereas the baseline is concentrated at higher frequencies and lacks a comparable low-frequency population (Figure 6d). The qualitative results in Figure 1 show the same structure: SPATIAL-SAE better preserves regional organization in PCA-RGB reconstructions, produces more stable strongest-atom assignments within segments, and learns class-best maps with significantly more continuous support.

6 CONCLUSION

We showed that tokenwise SAEs implicitly assume independence between patch codes, an assumption that vision-model activations violate. To replace it, we introduced MF-LRH, which specializes the LRH by coupling smooth codes across neighboring patches while retaining sparse innovations for local details, and derived SPATIAL-SAE as amortized MAP inference under this hypothesis. On synthetic data, SPATIAL-SAE improves concept recovery and reduces polysemanticity; on real vision-model activations, where these quantities cannot be measured without a ground-truth dictionary, it improves every proxy we evaluate. SPATIAL-SAE therefore moves along the RDP frontier toward interpretability, trading reconstruction for more interpretable features and coherent activation maps.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods (training, evaluation, and data-generation code), to design and provide feedback on experiments, and to assist in the writing of proofs. A language model is also part of our evaluation methodology: Claude Haiku 4.5 serves as the blinded rater in the automated interpretability protocol (Section 5.2 and Appendix D.3), and the full rater instructions are included in the supplementary material. We have not used generative AI tools to develop the theoretical framework, formulate mathematical claims, propose hypotheses, or interpret results, and no data were generated, translated, or cleaned by generative AI (the synthetic datasets are produced by our own generation code). Additionally, we used generative AI tools for brainstorming, to create and edit software code, create scientific figures, draft parts of the paper, and edit the paper for readability. We have reviewed all AI-assisted work. All AI-generated code was checked and tested for correctness by the authors, every AI-assisted proof was verified step by step by the authors, every experimental result was verified by the authors, and all AI-assisted text was reviewed and revised by the authors against the underlying derivations and results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects: no human annotators or participants were recruited, and all interpretability ratings were produced by a language model under a blinded protocol (Appendix D.3). All experiments use publicly available models and datasets (DINOv2, ADE20K, and COCO-Stuff), used under their respective licenses and terms of use for research. Some of these images contain people; we do not attempt to identify individuals, and we release no new images or annotations. The backbone and datasets may encode demographic and cultural biases, and dictionary features learned from their activations can inherit them; our method aims to make such internal structure easier to inspect, not to correct it. We foresee no direct harmful applications beyond those of interpretability research on vision models in general, and we believe more faithful concept extraction primarily benefits the auditing and understanding of deployed models.

REPRODUCIBILITY STATEMENT

We will publicly release the code for training and evaluating SPATIAL-SAE and all baselines, the synthetic data generator and datasets, and the trained models upon publication. The implementation extends the public `overcomplete` library, and the software versions and compute used are listed in Appendix D. On the theoretical side, all assumptions are stated in the main text and the complete proofs of Proposition 3.1 and Theorem 4.1, along with the supporting results, are given in Appendices B.2–B.4, with the required background on patch graphs and Markov random fields in Appendices A.1–A.2. For the synthetic benchmark, Appendix C.1 specifies the data-generating process, the dictionary and stream construction, the sweeps, and the training and evaluation protocol. For natural images, we use only the publicly available DINOv2 backbone and the ADE20K and COCO-STUFF datasets; Appendix D describes activation extraction and preprocessing, the patch graph, all architectures and hyperparameters (Section D.1), the evaluation metrics (Section D.2), and the model-selection and seeding protocol (Section D.4). The automated interpretability protocol, including panel construction, rater assignment, and aggregation, is described in Appendix D.3, and the full rater instructions are included in the supplementary material.

REFERENCES

Anthropic. Introducing Claude Haiku 4.5, October 2025. URL https://www.anthropic.com/news/claude-haiku-4-5. 41

Kola Ayonrinde, Michael T. Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering SAE explanations of neural activations with MDL-SAEs. *arXiv preprint arXiv:2410.11179*, 2024. 1

Francis Bach. Submodular functions: From discrete to continuous domains. *Mathematical Programming*, 175:419–459, 2019. doi: 10.1007/s10107-018-1248-6. URL https://arxiv.org/abs/1511.00394. 18, 19

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pp. 585–591, 2001. URL https://papers.nips.cc/paper/1961-laplacian-eigenmaps-and-spectral-techniques-for-embedding-and-clustering. 16

Usha Bhalla, Thomas Fel, Can Rager, Sheridan Feucht, Tal Haklay, Daniel Wurgaft, Siddharth Boppana, Matthew Kowal, Vasudev Shyam, Jack Merullo, Atticus Geiger, and Ekdeep Singh Lubana. Do sparse autoencoders capture concept manifolds? *arXiv preprint arXiv:2604.28119*, 2026. 1

Irving Biederman. Perceiving real-world scenes. *Science*, 177(4043):77–80, 1972. doi: 10.1126/science.177.4043.77. 4

Trenton Bricken et al. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023. 1, 3

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1209–1218, 2018. doi: 10.1109/CVPR.2018.00132. 35

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951. 4

Volkan Cevher, Marco F. Duarte, Chinmay Hegde, and Richard G. Baraniuk. Sparse signal recovery using Markov random fields. In *Advances in Neural Information Processing Systems*, volume 21, 2008. URL https://papers.neurips.cc/paper/3487-sparse-signal-recovery-using-markov-random-fields. 5

David Chanin and Adrià Garriga-Alonso. SynthSAEBench: Evaluating sparse autoencoders on scalable realistic synthetic data. *arXiv preprint arXiv:2602.14687*, 2026. 7, 8, 29, 33, 37

David Chanin, Tomáš Dulka, and Adrià Garriga-Alonso. Feature hedging: Correlated features break narrow sparse autoencoders. *arXiv preprint arXiv:2505.11756*, 2025a. 1, 7

David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 82318–82355, 2025b. doi: 10.52202/085713-2755. URL https://arxiv.org/abs/2409.14507. 1, 7

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 34381–34424, 2025. doi: 10.52202/085713-1154. URL https://arxiv.org/abs/2506.03093. 1, 3, 7, 20

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1. 35

Boyi Deng, Xu Wang, Yaoning Wang, Yu Wan, Yubo Ma, Baosong Yang, Haoran Wei, Jialong Tang, Huan Lin, Ruize Gao, Tianhao Li, Qian Cao, Xuancheng Ren, Xiaodong Deng, An Yang, Fei Huang, Dayiheng Liu, and Jingren Zhou. Qwen-Scope: Turning sparse features into development tools for large language models, 2026. URL https://arxiv.org/abs/2605.11887. 1

Michael Elad. *Sparse and Redundant Representations*. Springer, 2010. 3

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. Transformer Circuits Thread, 2022. https://transformer-circuits.pub/2022/toy_model/index.html, arXiv:2209.10652. 1, 3

Thomas Fel, Ekdeep Singh Lubana, Jacob Prince, Matthew Kowal, Victor Boutin, Isabel Papadimitriou, Binxu Wang, Martin Wattenberg, Demba Ba, and Talia Konkle. Archetypal SAE: Adaptive and stable dictionary learning for concept extraction in large vision models. In *International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 16543–16572. PMLR, 2025. 1, 37

Thomas Fel, Binxu Wang, Michael A. Lepori, Matthew Kowal, Andrew Lee, Randall Balestriero, Sonia Joseph, Ekdeep S. Lubana, Talia Konkle, Demba Ba, and Martin Wattenberg. Into the rabbit hull: From task-relevant concepts in DINO to Minkowski geometry. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.08638. 1, 3, 7, 20, 35

David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, 4(12):2379–2394, 1987. 5

Alkis Gotovos, S. Hamed Hassani, and Andreas Krause. Sampling from probabilistic submodular models. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper/5744-sampling-from-probabilistic-submodular-models. 18

Sai Sumedh R. Hindupur, Ekdeep Singh Lubana, Thomas Fel, and Demba Ba. Projecting assumptions: The duality between sparse autoencoders and concept geometry. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 11649–11699, 2025. doi: 10.52202/085713-0391. URL https://arxiv.org/abs/2503.01822. 3, 20, 21

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK. 1, 3

Samy Jelassi, Michael E. Sander, and Yuanzhi Li. Vision transformers provably learn spatial structure. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 37822–37836, 2022. doi: 10.52202/068431-2741. 4

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. On the origins of linear representations in large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21879–21911. PMLR, 2024. URL https://arxiv.org/abs/2403.03867. 3

Thomas Kailath. An innovations approach to least-squares estimation—part I: Linear filtering in additive white noise. *IEEE Transactions on Automatic Control*, 13(6):646–655, 1968. doi: 10.1109/TAC.1968.1099025. 2

Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 2

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on Gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 278–300. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.19. 1

Ekdeep Singh Lubana, Can Rager, Sai Sumedh R. Hindupur, Valérie Costa, Greta Tuckute, Oam Patel, Sonia Krishna Murthy, Thomas Fel, Daniel Wurgaft, Eric J. Bigelow, Johnny Lin, Demba Ba, Martin Wattenberg, Fernanda Viegas, Melanie Weber, and Aaron Mueller. Priors in time: Missing inductive biases for language model interpretability. In *International Conference on Learning Representations (ICLR)*, pp. 42231–42290, 2026. URL https://arxiv.org/abs/2511.01836. 1, 3, 22, 24, 25

Charles C. Margossian and David M. Blei. Amortized variational inference: When and why? In *Proceedings of the 40th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 244 of *Proceedings of Machine Learning Research*, pp. 2434–2449. PMLR, 2024. 21

Tommaso Mencattini, Francesco Montagna, and Francesco Locatello. The rate-distortion-polysemanticity tradeoff in SAEs. *arXiv preprint arXiv:2605.14694*, 2026. 1, 7, 8, 33

T. Neumann, K. Varanasi, C. Theobalt, M. Magnor, and M. Wacker. Compressed manifold modes for mesh processing. *Computer Graphics Forum*, 33(5):35–44, 2014. doi: 10.1111/cgf.12429. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12429. 6

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. 3

Bruno A. Olshausen and David J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. 3

Charles O'Neill, Alim Gumran, and David Klindt. Compute optimal inference and provable amortisation gap in sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 46877–46896. PMLR, 2025. URL https://arxiv.org/abs/2411.13117. 3, 21, 22

Maxime Oquab et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 35

Stephen E. Palmer. The effects of contextual scenes on the identification of objects. *Memory & Cognition*, 3(5):519–526, 1975. doi: 10.3758/BF03197524. 4

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 2024. 1, 3

Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, Cambridge, 2012. ISBN 9781107011793. doi: 10.1017/CBO9780511996504. 5, 17

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013. doi: 10.1109/MSP.2012.2235192. 16

Samuel Stevens, Wei-Lun Chao, Tanya Berger-Wolf, and Yu Su. Interpretable and testable vision features via sparse autoencoders, 2025. URL https://arxiv.org/abs/2502.06755. 1

Johan Wagemans, James H. Elder, Michael Kubovy, Stephen E. Palmer, Mary A. Peterson, Manish Singh, and Rüdiger von der Heydt. A century of Gestalt psychology in visual perception: I. perceptual grouping and figure-ground organization. *Psychological Bulletin*, 138(6):1172–1217, 2012. doi: 10.1037/a0029333. 4

Adrian Weller. Revisiting the limits of MAP inference by MWSS on perfect graphs. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 1061–1069. PMLR, 2015. URL https://proceedings.mlr.press/v38/weller15.html. 18

Yael Yankelevsky and Michael Elad. Theoretical guarantees for graph sparse coding. *Applied and Computational Harmonic Analysis*, 49(2):698–725, 2020. ISSN 1063-5203. doi: 10.1016/j.acha.2019.03.003. URL https://www.sciencedirect.com/science/article/pii/S106352031830232X. 6

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International journal of computer vision*, 127(3):302–321, 2019. 35

APPENDIX OVERVIEW

A Preliminaries **15**

A.1 Patch Graphs, Laplacian, and Dirichlet Energy 15

A.2 Undirected Graphical Models and Markov Random Fields 17

B Theory of SPATIAL-SAEs **20**

B.1 From LRH to Sparse Coding and Sparse Autoencoders 20

B.2 The MAP Interpretation of Vanilla SAEs 22

B.3 The MF-LRH Prior 25

B.4 From MF-LRH to SPATIAL-SAE 27

C Synthetic Experiments **29**

C.1 Experimental Setup 29

C.2 Effects of Decomposition and Spectral Regularization 33

D Experiments with Natural Images **35**

D.1 Implementation Details 36

D.2 Evaluation Metrics and Spectral Diagnostics 38

D.3 Automated Interpretability 41

D.4 Ablations, Model Selection, and Additional Results 42

A PRELIMINARIES

This appendix develops the mathematical language used to express spatial dependence in MF-LRH. In Section A.1, we represent image patches as vertices of a graph and introduce the Laplacian and Dirichlet energies that distinguish spatial agreement from spatial contrast. In Section A.2, we turn these graph interactions into probabilistic priors by reviewing undirected graphical models, pairwise Markov random fields, and MAP estimation. Together, these preliminaries support the theoretical development of SPATIAL-SAE in Section B.

A.1 PATCH GRAPHS, LAPLACIAN, AND DIRICHLET ENERGY

This subsection introduces the graph notation used to express spatial locality. The purpose is simple: patch activations in a vision model are not indexed by an arbitrary set, but by positions in an image. To make this spatial structure available to the prior, we represent the patch layout as a weighted graph. The graph specifies which patches interact. These interactions can favor agreement or contrast between neighboring codes.

> **Why a graph.** A graph encodes which patches are neighbors. Once patches are vertices and adjacency is encoded by edge weights, we can measure how much a latent concept field changes across edges. Penalizing these differences encourages smoothness; favoring them, while controlling code magnitude, encourages spatial contrast.

We first define the object that carries spatial information. Generally, the edge weights may depend on patch adjacency, attention, feature affinity, or a mixture of spatial and representation-space similarity. The formal arguments below only require an undirected graph with symmetric nonnegative weights.

Definition A.1 (Patch graph and graph Laplacian)**.** Let the image patches be equipped with an undirected weighted graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W}),$$

where $\mathcal{V} = \{1, \dots, p\}$ is the set of patch indices and $\mathcal{E}$ consists of unordered pairs of distinct vertices. The weight matrix $\mathbf{W} = (w_{ij}) \in \mathbb{R}^{p\times p}$ satisfies $w_{ij} = w_{ji} \geq 0$, with $w_{ii} = 0$ and $w_{ij} = 0$ whenever $\{i, j\} \notin \mathcal{E}$. Each undirected edge is counted once in sums and products over $\mathcal{E}$. The degree matrix is

$$\mathbf{\Delta} = \operatorname{diag}(\Delta_{11}, \dots, \Delta_{pp}), \qquad \Delta_{ii} = \sum_{j=1}^{p} w_{ij}.$$

The graph Laplacian associated with $\mathcal{G}$ is

$$\mathbf{L} = \mathbf{\Delta} - \mathbf{W}.$$

The Laplacian measures differences between neighboring patch codes through the graph Dirichlet energy. Large edge weights give the corresponding differences more influence. This measure of spatial variation applies to any latent field; its role in the objective depends on whether variation is penalized or favored.

Definition A.2 (Graph Dirichlet energy)**.** For a matrix $\mathbf{X} \in \mathbb{R}^{p\times c}$, whose row $\mathbf{x}_i$ denotes the latent code at patch i, the graph Dirichlet energy of $\mathbf{X}$ on $\mathcal{G}$ is

$$\operatorname{Dir}(\mathbf{X}) := \operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}).$$

This definition is compact, but its meaning is not obvious from the trace form alone. The quantity $\operatorname{Dir}(\mathbf{X})$ measures how much the concept field varies over the graph: it is small when neighboring patches have similar codes, and large when the code changes abruptly across high-weight edges. In this sense, it is the graph analogue of the classical continuous Dirichlet energy

$$\int_\Omega \|\nabla s(x)\|_2^2 \, dx,$$

which measures the spatial variation of a function s over a continuous domain Ω. The next lemma expresses both the Laplacian quadratic and its shifted counterpart in terms of squared differences between neighboring patch codes. These identities describe attractive and repulsive interactions on the graph.

Lemma A.3 (Laplacian trace identities). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ be a patch graph as in Definition A.1, and let $\mathbf{L} = \mathbf{\Delta} - \mathbf{W}$ be its graph Laplacian. Then, for every $\mathbf{X} \in \mathbb{R}^{p\times c}$ and $\beta \in \mathbb{R}$,*

$$\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) = \sum_{\{i,j\}\in\mathcal{E}} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2,$$

$$\operatorname{Tr}\big(\mathbf{X}^\top(\beta\mathbf{I}_p - \mathbf{L})\mathbf{X}\big) = \beta \left\|\mathbf{X}\right\|_F^2 - \sum_{\{i,j\}\in\mathcal{E}} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2.$$

Proof. Using the definition of the graph Laplacian,

$$\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) = \operatorname{Tr}(\mathbf{X}^\top \mathbf{\Delta}\mathbf{X}) - \operatorname{Tr}(\mathbf{X}^\top \mathbf{W}\mathbf{X}).$$

The first term expands as

$$\operatorname{Tr}(\mathbf{X}^\top \mathbf{\Delta}\mathbf{X}) = \sum_{i=1}^{p} \Delta_{ii} \left\|\mathbf{x}_i\right\|_2^2,$$

and the second term expands as

$$\operatorname{Tr}(\mathbf{X}^\top \mathbf{W}\mathbf{X}) = \sum_{i,j=1}^{p} w_{ij} \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle.$$

Since $\Delta_{ii} = \sum_j w_{ij}$ and $\mathbf{W}$ is symmetric, we can rewrite the difference as

$$\begin{aligned}
\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) &= \sum_i \Delta_{ii} \left\|\mathbf{x}_i\right\|_2^2 - \sum_{i,j} w_{ij} \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \\
&= \frac{1}{2} \sum_{i,j} w_{ij} \left(\left\|\mathbf{x}_i\right\|_2^2 + \left\|\mathbf{x}_j\right\|_2^2 - 2 \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle \right) \\
&= \frac{1}{2} \sum_{i,j} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 \\
&= \sum_{\{i,j\}\in\mathcal{E}} w_{ij} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2.
\end{aligned}$$

Each undirected edge contributes twice to the ordered sum, giving the edge-sum form of the first identity. For the second identity, use

$$\operatorname{Tr}\big(\mathbf{X}^\top(\beta\mathbf{I}_p - \mathbf{L})\mathbf{X}\big) = \beta \left\|\mathbf{X}\right\|_F^2 - \operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X})$$

and substitute the first identity. □

Proof intuition. The degree and adjacency terms regroup into weighted squared differences across edges. The shifted quadratic reverses these edge contributions and adds a term proportional to the squared field magnitude.

Minimizing $\operatorname{Dir}(\mathbf{X})$ encourages neighboring patches to have similar concept activations. This is the graph smoothness penalty used in spectral graph methods and graph signal processing (Belkin & Niyogi, 2001; Shuman et al., 2013). Conversely, at a fixed magnitude, larger $\operatorname{Dir}(\mathbf{X})$ favors differences between neighboring codes. A negative Dirichlet term alone, however, would be unbounded below on any graph with a positive-weight edge, since increasing the magnitude of a varying field would decrease the energy without limit.

Corollary A.4 (Stabilized repulsion). *Under the assumptions of Lemma A.3, if $\beta \geq \lambda_{\max}(\mathbf{L})$, then for every $\mathbf{X} \in \mathbb{R}^{p\times c}$,*

$$\operatorname{Tr}\big(\mathbf{X}^\top(\beta\mathbf{I}_p - \mathbf{L})\mathbf{X}\big) = \beta \left\|\mathbf{X}\right\|_F^2 - \operatorname{Dir}(\mathbf{X}) \geq 0.$$

Proof. The equality follows from Lemma A.3. For nonnegativity, set $\mathbf{A} = \beta\mathbf{I}_p - \mathbf{L}$. Its eigenvalues are $\beta - \lambda_k \geq 0$ under the assumed bound on β, so $\mathbf{A} \succeq 0$. Write $\mathbf{X} = [\mathbf{v}_1, \ldots, \mathbf{v}_c]$ column-wise, where $\mathbf{v}_a \in \mathbb{R}^p$. Then

$$\operatorname{Tr}(\mathbf{X}^\top \mathbf{A}\mathbf{X}) = \sum_{a=1}^{c} (\mathbf{X}^\top \mathbf{A}\mathbf{X})_{aa} = \sum_{a=1}^{c} \mathbf{v}_a^\top \mathbf{A}\mathbf{v}_a \geq 0,$$

because each summand is nonnegative when $\mathbf{A} \succeq 0$. □

For $\beta \geq \lambda_{\max}(\mathbf{L})$, the negative edge term favors contrast, while the magnitude term keeps the shifted quadratic nonnegative. The two operators give complementary spectral preferences: $\mathbf{L}$ penalizes a graph mode with eigenvalue λ_k in proportion to λ_k, whereas $\beta\mathbf{I}_p - \mathbf{L}$ penalizes it in proportion to $\beta - \lambda_k$. Thus, at a fixed field magnitude, the Laplacian quadratic favors low graph frequencies and the shifted quadratic favors high graph frequencies.

The graph therefore supports two complementary spatial preferences. The next subsection explains how attractive and repulsive interactions are expressed through potentials in an undirected graphical model.

A.2 Undirected Graphical Models and Markov Random Fields

This subsection introduces the probabilistic language needed to turn spatial locality into a prior. The graph preliminaries above tell us which patches are neighbors. We now explain how a graph can also describe dependencies between random variables attached to those patches. This is the role of undirected graphical models. We introduce local dependence, potential functions, pairwise Markov random fields, and the MAP principle that turns a probabilistic model into an optimization objective.

Why graphical models. A patch graph says which patches are adjacent. An undirected graphical model uses this adjacency structure to describe which latent variables interact directly. For the quadratic difference interactions used here, attraction favors neighboring agreement, while repulsion favors contrast. Both preferences can be expressed on the same patch adjacency structure.

Following the classical computer-vision view of graphical models (Prince, 2012), the point of the graph is not merely to draw connections between variables. The graph is a modeling assumption: it says that the joint distribution should be built from local interactions rather than from arbitrary global dependencies. This is particularly natural for images, where pixels or patches live on a spatial grid.

Definition A.5 (Undirected graphical model)**.** An undirected graphical model consists of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a joint distribution over its vertex-indexed variables $\{X_i\}_{i\in\mathcal{V}}$ satisfying the Markov property in Definition A.6. The graph specifies which variables may interact directly: an edge $\{i, j\} \in \mathcal{E}$ means that X_i and X_j may have a direct probabilistic interaction.

In this appendix, the undirected graphical model of interest is a Markov random field. The Markov property makes the locality assumption explicit: once the neighbors of a variable are known, the rest of the graph does not directly change the conditional distribution of that variable.

Definition A.6 (Markov random field)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and let $\{X_i\}_{i\in\mathcal{V}}$ be random variables indexed by its vertices. The collection $\{X_i\}_{i\in\mathcal{V}}$ is a Markov random field on $\mathcal{G}$ if, for every $i \in \mathcal{V}$, the following equality holds almost surely:

$$p(x_i \mid x_{\mathcal{V}\setminus\{i\}}) = p(x_i \mid x_{\mathcal{N}(i)}),$$

where

$$\mathcal{N}(i) = \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$$

is the set of neighbors of i.

The definition says that the prior is local. A patch code may interact directly with neighboring patch codes, but it does not need a separate direct interaction with every other patch in the image. This is the same modeling principle used in classical grid-based vision models. Locality specifies which variables interact; the interaction potentials determine whether they favor spatial agreement or contrast.

To specify which local configurations are likely or unlikely, graphical models use potential functions. There are two equivalent conventions. One can use positive compatibility potentials, whose larger values make a configuration more probable. Or one can use energy potentials, whose larger values make a configuration less probable through a Gibbs density. In this appendix we use the energy convention because it connects directly to MAP minimization.

Definition A.7 (Pairwise Markov random fields and potentials)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with variables $\{X_i\}_{i\in\mathcal{V}}$.

A *compatibility potential* is a strictly positive function. Unary compatibility potentials $\psi_i(x_i) > 0$ score individual states, and pairwise compatibility potentials $\psi_{ij}(x_i, x_j) > 0$ score joint states of neighboring variables. In sums and products over $\mathcal{E}$, each undirected edge is counted once. In compatibility form, a pairwise Markov random field has density

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \psi_i(x_i) \prod_{\{i,j\} \in \mathcal{E}} \psi_{ij}(x_i, x_j).$$

Larger compatibility values increase the probability of the corresponding configuration.

Equivalently, define unary and pairwise *energy potentials*

$$U_i(x_i) = -\log \psi_i(x_i), \qquad P_{ij}(x_i, x_j) = -\log \psi_{ij}(x_i, x_j).$$

Then

$$p(\mathbf{x}) = \frac{1}{Z} \exp(-E(\mathbf{x})), \qquad E(\mathbf{x}) = \sum_{i \in \mathcal{V}} U_i(x_i) + \sum_{\{i,j\} \in \mathcal{E}} P_{ij}(x_i, x_j).$$

In energy form, lower potential values are preferred by the prior, while higher potential values decrease probability. Individual energy potentials need not be nonnegative: a repulsive potential can decrease as neighboring codes differ. The complete energy must define a finite, positive normalizing constant $0 < Z < \infty$ for the density to be well-defined.

Since MAP inference minimizes negative log probability, the energy convention directly yields optimization objectives for both attractive and repulsive interactions.

Unary potentials encode preferences about individual variables, such as sparsity or magnitude. Pairwise potentials determine how neighboring variables interact. To define attraction and repulsion for continuous variables, we first recall submodularity on products of ordered domains (Bach, 2019, Sections 2.1–2.2).

Definition A.8 (Submodularity and supermodularity)**.** Let $\mathcal{X} = \prod_{k=1}^{n} I_k$, where each $I_k \subseteq \mathbb{R}$ is an interval. For $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, let $\mathbf{x} \wedge \mathbf{y}$ and $\mathbf{x} \vee \mathbf{y}$ denote their coordinatewise minimum and maximum. A function $f : \mathcal{X} \to \mathbb{R}$ is *submodular* if

$$f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \wedge \mathbf{y}) + f(\mathbf{x} \vee \mathbf{y}) \qquad \text{for all } \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

It is *supermodular* if the inequality is reversed, and *modular* if equality holds for all $\mathbf{x}, \mathbf{y}$. A positive function $\psi : \mathcal{X} \to (0, \infty)$ is *log-submodular* or *log-supermodular* when $\log \psi$ has the corresponding property.

The following definition adapts the binary attraction and repulsion conventions of Weller (2015, Section 3.1.1) and Gotovos et al. (2015) to continuous variables, using the same relation between energy and log-compatibility.

Definition A.9 (Attractive and repulsive pairwise MRFs)**.** Consider a continuous pairwise MRF on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with density

$$q(\mathbf{x} \mid \mathcal{G}) = \frac{1}{Z} \exp\left(-\sum_{i \in \mathcal{V}} U_i(x_i) - \sum_{\{i,j\} \in \mathcal{E}} P_{ij}(x_i, x_j) \right),$$

on a product of intervals, where $0 < Z < \infty$ and each undirected edge is counted once. The functions U_i and P_{ij} are unary and pairwise energy potentials, respectively.

The field is *attractive* if every P_{ij} is submodular, and *repulsive* if every P_{ij} is supermodular. Equivalently, the edge compatibility factors $\exp(-P_{ij})$ are respectively log-supermodular and log-submodular. We include zero interactions as a boundary case.

Attraction lets neighboring patches reinforce a feature: holding the other variables fixed, a larger feature value at one patch makes larger values at its neighbor relatively more likely. Repulsion reverses this preference. For nonnegative codes, it discourages strong neighboring activations of the same feature.

For twice continuously differentiable pairwise energies, attraction and repulsion are equivalently characterized by

$$\frac{\partial^2 P_{ij}}{\partial x_i \, \partial x_j} \leq 0 \quad \text{and} \quad \frac{\partial^2 P_{ij}}{\partial x_i \, \partial x_j} \geq 0,$$

respectively (Bach, 2019, Section 2.2). Scalar unary terms are separable and therefore modular: adding them to an energy changes neither classification. The lattice definition itself requires no differentiability, so this observation also applies to nonsmooth unary penalties such as $\lambda|x_i|$.

For the negative Dirichlet energy considered here, Corollary A.4 supplies a unary magnitude term that makes the combined quadratic nonnegative. Adding a positive ℓ_1 penalty controls directions with zero quadratic cost and yields a normalizable density.

The last probabilistic object we need is MAP inference. This is the bridge from a graphical model to an optimization objective. Once the prior and likelihood are written in energy form, MAP inference becomes minimization of a data-fit term plus a prior energy.

Definition A.10 (MAP estimator)**.** Given observed data $\mathbf{y}$ and latent variables $\mathbf{x}$, the maximum a posteriori estimator is any configuration

$$\mathbf{x}_{\mathrm{MAP}} \in \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y}).$$

Equivalently, if

$$p(\mathbf{x} \mid \mathbf{y}) \propto \exp(-E(\mathbf{x}; \mathbf{y})) \,,$$

then

$$\mathbf{x}_{\mathrm{MAP}} \in \arg\min_{\mathbf{x}} E(\mathbf{x}; \mathbf{y}).$$

On a product domain, with respect to a product reference measure, a prior whose energy contains only unary terms factorizes across variables. Pairwise potentials can introduce local dependencies, which appear as graph-dependent terms in the MAP objective. For continuous patch codes, the quadratic energies above can express either spatial agreement or spatial contrast, with unary terms promoting sparsity and controlling magnitude.

An undirected graphical model therefore specifies which variables interact, potential functions specify which configurations are preferred, and MAP inference turns the posterior into an energy minimization problem.

B THEORY OF SPATIAL-SAES

Having established the required graph and probabilistic preliminaries, we now give the full theoretical account of SPATIAL-SAE. In Section B.1, we connect the LRH to sparse coding and its amortization by an SAE. In Section B.2, we derive the MAP interpretation of the vanilla SAE objective and expose its factorized prior over patch codes. We then replace that prior with the spatially coupled MF-LRH prior in Section B.3. In Section B.4, we derive the resulting SPATIAL-SAE objective, relate it to amortized inference, and show that it depends on spatial layout rather than an arbitrary ordering of the patches.

B.1 FROM LRH TO SPARSE CODING AND SPARSE AUTOENCODERS

This subsection fixes the algebraic starting point of the appendix. We first state the Linear Representation Hypothesis in the notation used here. We then explain why the hypothesis turns concept extraction into sparse dictionary learning, and finally define the vanilla sparse autoencoder as the amortized version of that sparse-coding problem. At this stage no spatial assumption is being made; the goal is only to make clear what classical LRH and vanilla SAEs do and do not specify.

> **From representations to inference.** The LRH says that an activation is a sparse linear combination of dictionary atoms. Once this is assumed, interpretability becomes an inverse problem: infer the sparse coefficients and the dictionary. A sparse autoencoder replaces the per-example sparse-coding solve by a learned encoder $f_{\boldsymbol{\theta}}$.

We begin by restating the LRH as stated in Costa et al. (2025, Def. 2.1), in the notation of Fel et al. (2026, Def. 1):

Definition B.1 (Linear Representation Hypothesis)**.** A representation $\mathbf{h} \in \mathbb{R}^d$ satisfies the Linear Representation Hypothesis, abbreviated LRH, if there exists a sparsity constant k, an overcomplete dictionary $\mathbf{D} = (\mathbf{d}_1, \dots, \mathbf{d}_c) \in \mathbb{R}^{c \times d}$, and a coefficient vector $\mathbf{z} \in \mathbb{R}^c$ such that $\mathbf{h} \approx \mathbf{z}\mathbf{D}$, under the following conditions:

$$\begin{cases} \text{(i) Overcompleteness:} & c \gg d, \\ \text{(ii) Quasi-orthogonality:} & \mu(\mathbf{D}) := \max_{a \neq b} \left|\mathbf{d}_a \mathbf{d}_b^\top\right| \leq \varepsilon, \quad \|\mathbf{d}_a\|_2 = 1 \; \forall a, \\ \text{(iii) Sparsity:} & |\mathrm{supp}(\mathbf{z})| \leq k \ll c. \end{cases}$$

The definition is local: it describes one activation vector $\mathbf{h} \in \mathbb{R}^d$. In practice, the object of interest is a collection of activations $\mathbf{H} \in \mathbb{R}^{n \times d}$. If $\mathbf{H}$ collects them row by row, applying Definition B.1 to each row gives a sparse code matrix $\mathbf{Z} \in \mathbb{R}^{n \times c}$ and the approximation $\mathbf{H} \approx \mathbf{Z}\mathbf{D}$. Thus, *under* LRH*, concept extraction becomes the problem of recovering both the dictionary and the sparse code matrix*.

This inverse problem is a dictionary-learning problem. As argued by Fel et al. (2026), different concept extraction methods correspond to different constraints on the code $\mathbf{Z}$ and dictionary $\mathbf{D}$: clustering methods constrain codes to select prototypical examples as atoms, orthogonal factorization constrains the dictionary to be orthogonal, nonnegative factorization imposes nonnegativity, and sparse autoencoders constrain codes to be produced by an encoder with a sparsity-inducing mechanism (Hindupur et al., 2025). *These are not neutral algorithmic choices: each constraint encodes a different prior over what kind of decomposition should count as a good explanation.*

Under LRH, the relevant assumptions are **overcompleteness** and **sparsity**. Therefore the dictionary-learning problem aligned with LRH is sparse coding: learn an overcomplete dictionary and assign each activation a sparse coefficient vector.

Definition B.2 (Sparse coding induced by LRH)**.** Given activations $\mathbf{H} \in \mathbb{R}^{n \times d}$, a latent width c, and a sparsity regularizer $\mathcal{R}$, the sparse-coding problem induced by LRH is

$$(\mathbf{Z}^\star, \mathbf{D}^\star) \in \arg\min_{\mathbf{Z}, \mathbf{D}} \frac{1}{n} \sum_{i=1}^{n} \left[\|\mathbf{h}_i - \mathbf{z}_i \mathbf{D}\|_2^2 + \lambda \mathcal{R}(\mathbf{z}_i) \right],$$

where $\mathbf{z}_i$ is the sparse code assigned to activation $\mathbf{h}_i$, $\mathbf{D}$ is the learned dictionary, and $\lambda \geq 0$ controls the sparsity strength.

Sparse coding is faithful to the algebraic content of LRH: it searches for an overcomplete dictionary and sparse coefficients that reconstruct the observed activations. However, it is computationally expensive at scale. For a fixed dictionary $\mathbf{D}$, the code $\mathbf{z}_i$ of each activation $\mathbf{h}_i$ is itself the solution of a sparse inference problem:

$$\mathbf{z}^\star(\mathbf{h}_i;\mathbf{D}) \in \arg\min_{\mathbf{z}\in\mathbb{R}^c} \left[\|\mathbf{h}_i - \mathbf{z}\mathbf{D}\|_2^2 + \lambda\mathcal{R}(\mathbf{z})\right].$$

Thus sparse coding requires solving, or approximately solving, a new optimization problem for every activation. This is the computational motivation for amortized inference: instead of solving the sparse inference problem separately for each $\mathbf{h}_i$, one learns a shared function $f_{\boldsymbol{\theta}}$ that predicts the code directly from the activation, $\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i)$.

Definition B.3 (Amortized sparse inference)**.** Let

$$\mathcal{Q}(\mathbf{z};\mathbf{h},\mathbf{D}) := \|\mathbf{h} - \mathbf{z}\mathbf{D}\|_2^2 + \lambda\mathcal{R}(\mathbf{z})$$

be the sparse inference objective for the code $\mathbf{z}$ of an activation $\mathbf{h}$ under dictionary $\mathbf{D}$. The exact sparse-coding solution is

$$\mathbf{z}^\star(\mathbf{h};\mathbf{D}) \in \arg\min_{\mathbf{z}\in\mathbb{R}^c} \mathcal{Q}(\mathbf{z};\mathbf{h},\mathbf{D}).$$

Sparse inference is *amortized* when this per-activation optimization is replaced by a shared inference function $f_{\boldsymbol{\theta}}$,

$$\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i) \qquad \forall i.$$

The parameters $\boldsymbol{\theta}$ of $f_{\boldsymbol{\theta}}$ are learned across the dataset, so the cost of inference is shared across activations.

This definition adapts the standard idea of amortized inference to sparse coding. In amortized variational inference, one learns a common inference function that maps each observation to the approximate latent object associated with that observation (Margossian & Blei, 2024). In the sparse-coding setting, the latent object is not a posterior distribution but a sparse coefficient vector. The benefit is speed: once $f_{\boldsymbol{\theta}}$ is learned, codes can be produced by a forward pass rather than by solving the sparse-coding problem above. The cost is approximation: the encoder may not return the exact sparse coding solution. Following O'Neill et al. (2025), we can measure this cost by the amortization gap

$$\Delta_{\text{amort}}(\boldsymbol{\theta};\mathbf{h},\mathbf{D}) := \mathcal{Q}(f_{\boldsymbol{\theta}}(\mathbf{h});\mathbf{h},\mathbf{D}) - \min_{\mathbf{z}\in\mathbb{R}^c} \mathcal{Q}(\mathbf{z};\mathbf{h},\mathbf{D}) \geq 0.$$

For a dataset, the corresponding average gap is $\Delta_{\text{amort}}(\boldsymbol{\theta};\mathbf{H},\mathbf{D}) = \frac{1}{n}\sum_{i=1}^n \Delta_{\text{amort}}(\boldsymbol{\theta};\mathbf{h}_i,\mathbf{D})$. This gap is zero only when the encoder family is expressive enough, and trained well enough, to return an optimal sparse code for each activation. In standard SAEs, the encoder is usually a restricted linear–nonlinear map, so the amortization gap can be nonzero even when better sparse codes exist (O'Neill et al., 2025).

We can now formalize the central point: a sparse autoencoder is an amortized solution to the sparse inference problem induced by LRH. Instead of solving

$$\arg\min_{\mathbf{z}\in\mathbb{R}^c} \mathcal{Q}(\mathbf{z};\mathbf{h}_i,\mathbf{D})$$

separately for every activation $\mathbf{h}_i$, an SAE learns an encoder $f_{\boldsymbol{\theta}}$ that returns the code directly. The bilevel formulation of Hindupur et al. (2025) makes this precise: the outer problem is the sparse dictionary-learning objective, while the inner constraint represents the architecture-dependent encoder map that determines which codes can be returned by the SAE.

Definition B.4 (Sparse autoencoder as amortized sparse inference)**.** A sparse autoencoder, abbreviated SAE, for activations $\mathbf{H} \in \mathbb{R}^{n\times d}$ consists of an encoder $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^c$ and a decoder dictionary $\mathbf{D} \in \mathbb{R}^{c\times d}$. It solves the constrained sparse-coding problem

$$\arg\min_{\mathbf{D},\mathbf{Z},\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^n \left[\|\mathbf{h}_i - \mathbf{z}_i\mathbf{D}\|_2^2 + \lambda\mathcal{R}(\mathbf{z}_i)\right], \qquad \text{s.t.} \quad \mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i) \in \arg\min_{\boldsymbol{\pi}\in\mathcal{S}_{\boldsymbol{\theta}}} F_{\boldsymbol{\theta}}(\boldsymbol{\pi};\mathbf{h}_i) \quad \forall i,$$

under the following conditions:

$$\begin{cases} \text{(i) Overcomplete latent width:} & c \gg d, \\ \text{(ii) Amortized inference:} & \mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i), & \forall i \in \{1, \dots, n\}, \\ \text{(iii) Encoder-induced constraint:} & \mathbf{z}_i \in \arg\min_{\boldsymbol{\pi} \in \mathcal{S}_{\boldsymbol{\theta}}} F_{\boldsymbol{\theta}}(\boldsymbol{\pi}; \mathbf{h}_i), & \forall i \in \{1, \dots, n\}, \\ \text{(iv) Sparse codes:} & \mathcal{R}(\mathbf{z}_i) \text{ penalizes dense codes}, & \text{for example } \mathcal{R}(\mathbf{z}_i) = \|\mathbf{z}_i\|_1, \\ \text{(v) Linear decoding:} & \widehat{\mathbf{h}}_i = \mathbf{z}_i \mathbf{D}. \end{cases}$$

Here $F_{\boldsymbol{\theta}}$ is a variational formulation of the encoder, and $\mathcal{S}_{\boldsymbol{\theta}}$ is the architecture-induced feasible set for the encoder output. For a standard one-layer SAE, this constraint can be instantiated by

$$f_{\boldsymbol{\theta}}(\mathbf{h}) = g(\mathbf{h}\mathbf{W} + \mathbf{b}),$$

with the choice of nonlinearity g determining the encoder family (e.g., ReLU or TopK SAEs).

The definition makes precise how an SAE relates to sparse coding. The outer objective is the same reconstruction-versus-sparsity tradeoff that appears in sparse dictionary learning. The difference is the constraint $\mathbf{z}_i = f_{\boldsymbol{\theta}}(\mathbf{h}_i)$. This constraint makes inference fast, because the code is obtained by a learned encoder rather than by solving $\arg\min_{\mathbf{z}} \mathcal{Q}(\mathbf{z}; \mathbf{h}_i, \mathbf{D})$ for every activation. But it also changes the solution class. The encoder can be a computationally restricted sparse-inference procedure, so its output can differ from the code selected by exact or more expressive sparse inference. This is the amortization cost of SAEs: speed is obtained by restricting inference to the range of $f_{\boldsymbol{\theta}}$ (O'Neill et al., 2025).

This subsection introduced the standard setting behind SAEs and the LRH. LRH motivates sparse, overcomplete dictionary learning; sparse coding is the dictionary-learning solution aligned with that hypothesis; and an SAE is an amortized approximation to sparse coding. The next subsection asks what probabilistic prior is implicitly imposed by the vanilla SAE objective. *We want to understand which additional assumptions it makes beyond the* LRH*, and whether they are realistic.*[2] The next subsection builds on Lubana et al. (2026).

B.2 The MAP Interpretation of Vanilla SAEs

This subsection gives the probabilistic version of the previous subsection's claim. There, sparse coding was introduced as the dictionary-learning objective aligned with LRH: under Definition B.1, activations are assumed to be sparse linear combinations of atoms in an overcomplete dictionary, and sparse coding is the optimization problem that recovers such codes. Here we show that this same objective arises as the MAP estimator, in the sense of Definition A.10, for a noisy version of LRH.[3] *The* MAP *perspective is the precise Bayesian sense in which sparse coding, and consequently* SAEs *as amortized approximations to sparse coding, are natural tools for* LRH. The Bayesian reading is also useful because it exposes an assumption that is hidden in the purely algebraic view. LRH *asks for sparsity and overcompleteness, but the vanilla* SAE *objective adds another choice: its sparsity prior is factorized over indexed activations, imposing an independence assumption between feature activations of the same input*. Thus the MAP view both justifies why sparse coding matches LRH and reveals why vanilla SAEs may still be mismatched when the indexed activations have structure, such as time in language or space in vision.

MAP alignment and a hidden independence prior. Sparse coding is the MAP formalization of the optimization problem suggested by LRH. A noisy linear reconstruction model gives the data-fit term, and a sparsity-promoting prior gives the sparse-code term. But once the sparsity cost is written as a sum over indexed activations, the corresponding prior becomes a product prior. For patch tokens, this means the vanilla SAE prior does not know which patches are neighbors.

[2] This is the key passage of the first part of the theory: SAEs arise as an approximation to the natural solution of concept discovery under LRH (namely sparse coding), as sparse coding matches the assumptions of LRH on the data-generating process. However, SAEs add hidden assumptions on top of those of the LRH, which are unrealistic for vision and make SAEs no longer the natural solution. These assumptions become apparent only when a Bayesian approach is adopted.

[3] Noise is required for a Bayesian interpretation of the objective: it turns the deterministic approximation into a likelihood.

We start from the noisy version of LRH. The deterministic statement $\mathbf{H} \approx \mathbf{ZD}$ is replaced by a generative model in which activations are noisy linear reconstructions from sparse latent codes:

$$\mathbf{h}_i = \mathbf{z}_i\mathbf{D} + \boldsymbol{\psi}_i, \qquad \boldsymbol{\psi}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d), \qquad i = 1, \dots, n.$$

Equivalently, at the matrix level,

$$\mathbf{H} = \mathbf{ZD} + \boldsymbol{\Psi}.$$

The next proposition shows that sparse coding is the MAP estimator of this noisy LRH model when the prior over codes is sparsity-promoting.

Proposition B.5 (Sparse coding as MAP inference for noisy LRH)**.** *Assume the noisy* LRH *model*

$$\mathbf{h}_i = \mathbf{z}_i\mathbf{D} + \boldsymbol{\psi}_i, \qquad \boldsymbol{\psi}_i \sim \mathcal{N}(0, \sigma^2\mathbf{I}_d),$$

with conditionally independent noise across indexed activations. Treat $\mathbf{D}$ *as fixed, and assume a sparsity-promoting prior over the code matrix, namely a prior whose density decreases as the sparsity penalty* $\mathcal{R}(\mathbf{Z})$ *increases. For example, we may take the Gibbs prior*

$$p(\mathbf{Z}) \propto \exp(-\lambda\mathcal{R}(\mathbf{Z})), \qquad \lambda \geq 0.$$

Then the MAP *estimator of* $\mathbf{Z}$ *is*

$$\mathbf{Z}_{\mathrm{MAP}} \in \arg\min_{\mathbf{Z}} \frac{1}{2\sigma^2} \|\mathbf{H} - \mathbf{ZD}\|_F^2 + \lambda\mathcal{R}(\mathbf{Z}).$$

After absorbing σ^2 *into the regularization weight, this is the sparse-coding objective in Definition B.2.*

Proof. Because the noise variables are conditionally independent across indexed activations, the likelihood factorizes over rows:

$$p(\mathbf{H} \mid \mathbf{Z}, \mathbf{D}) = \prod_{i=1}^{n} p(\mathbf{h}_i \mid \mathbf{z}_i, \mathbf{D}).$$

For a single activation, the Gaussian noise assumption gives

$$p(\mathbf{h}_i \mid \mathbf{z}_i, \mathbf{D}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{h}_i - \mathbf{z}_i\mathbf{D}\|_2^2\right).$$

Multiplying these terms over $i = 1, \dots, n$ gives

$$p(\mathbf{H} \mid \mathbf{Z}, \mathbf{D}) = \frac{1}{(2\pi\sigma^2)^{nd/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} \|\mathbf{h}_i - \mathbf{z}_i\mathbf{D}\|_2^2\right).$$

Since

$$\sum_{i=1}^{n} \|\mathbf{h}_i - \mathbf{z}_i\mathbf{D}\|_2^2 = \|\mathbf{H} - \mathbf{ZD}\|_F^2,$$

the likelihood can be written as

$$p(\mathbf{H} \mid \mathbf{Z}, \mathbf{D}) \propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{H} - \mathbf{ZD}\|_F^2\right).$$

By Bayes' rule,

$$p(\mathbf{Z} \mid \mathbf{H}, \mathbf{D}) \propto p(\mathbf{H} \mid \mathbf{Z}, \mathbf{D})p(\mathbf{Z}).$$

Substituting the likelihood and the Gibbs sparsity prior gives

$$p(\mathbf{Z} \mid \mathbf{H}, \mathbf{D}) \propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{H} - \mathbf{ZD}\|_F^2 - \lambda\mathcal{R}(\mathbf{Z})\right).$$

Therefore maximizing the posterior is equivalent to minimizing the negative log posterior:

$$-\log p(\mathbf{Z} \mid \mathbf{H}, \mathbf{D}) = \frac{1}{2\sigma^2} \|\mathbf{H} - \mathbf{ZD}\|_F^2 + \lambda\mathcal{R}(\mathbf{Z}) + \text{const}.$$

Dropping the constant independent of $\mathbf{Z}$ gives the stated MAP objective. □

Proof intuition. The proof only translates noisy LRH into a Bayesian inverse problem. Gaussian noise says that reconstructions with smaller squared error are more likely. A sparsity-promoting prior says that code matrices with lower sparsity penalty are more likely. The posterior combines these two preferences, so its negative logarithm is exactly a reconstruction term plus a sparsity term. This is the probabilistic sense in which sparse coding is the natural optimization tool for LRH.

This is also the point at which the Bayesian reading becomes useful for critique. From the algebraic perspective, the intended assumptions are sparsity and overcompleteness. From the Bayesian perspective, we can inspect the full prior that the objective imposes on $\mathbf{Z}$. The next proposition shows that the standard tokenwise sparsity term does more than impose sparsity: it also imposes independence across indexed activations.

We now specialize to the case where the indexed activations are **patch tokens from one image**. In this case, $\mathbf{H} \in \mathbb{R}^{p \times d}$ and $\mathbf{Z} \in \mathbb{R}^{p \times c}$, where p is the number of patches.

Proposition B.6 (Spatial independence prior of tokenwise SAEs)**.** *Let $\mathbf{H} \in \mathbb{R}^{p \times d}$ be the patch-token activation matrix of one image, and let $\mathbf{Z} \in \mathbb{R}^{p \times c}$ be the corresponding code matrix. Suppose the vanilla* SAE *objective has the row-additive form*

$$\mathcal{L}_{\text{SAE}}(\mathbf{Z}, \mathbf{D}\,;\, \mathbf{H}) = \sum_{i=1}^{p} \ell(\mathbf{h}_i, \mathbf{z}_i; \mathbf{D}) + \lambda \sum_{i=1}^{p} \mathcal{R}(\mathbf{z}_i),$$

where ℓ is the reconstruction loss for patch i. If this objective is interpreted as a MAP *objective, then the induced prior over patch codes factorizes as*

$$p(\mathbf{Z}) = p(\mathbf{z}_1, \ldots, \mathbf{z}_p) = \prod_{i=1}^{p} p(\mathbf{z}_i).$$

If the same regularizer is used at every patch, the marginal prior has the same form at every patch.

The proof is the spatial analogue of the temporal factorization argument in Lubana et al. (2026, App. I.1); we adapt their proof strategy from the time index of language-model activations to the patch index of vision-model activations.

Proof. In a MAP objective, the part of the loss that does not depend on the observed activation is the negative log prior. In the row-additive SAE objective, this prior contribution is

$$\lambda \sum_{i=1}^{p} \mathcal{R}(\mathbf{z}_i).$$

Thus the corresponding negative log prior has the form

$$-\log p(\mathbf{Z}) = \lambda \sum_{i=1}^{p} \mathcal{R}(\mathbf{z}_i) + \text{const}.$$

Exponentiating both sides gives

$$p(\mathbf{Z}) \propto \exp\left(-\lambda \sum_{i=1}^{p} \mathcal{R}(\mathbf{z}_i)\right).$$

Since the exponential of a sum is the product of exponentials,

$$p(\mathbf{Z}) \propto \prod_{i=1}^{p} \exp(-\lambda \mathcal{R}(\mathbf{z}_i)).$$

Define the one-patch marginal prior

$$p(\mathbf{z}_i) \propto \exp(-\lambda \mathcal{R}(\mathbf{z}_i)).$$

Then

$$p(\mathbf{Z}) = \prod_{i=1}^{p} p(\mathbf{z}_i),$$

up to the normalizing constant. Therefore the patch codes are independent under the prior. If the same $\mathcal{R}$ is used for every patch, these marginals have the same functional form. □

Proof intuition. The proof uses the same energy-to-probability conversion as the MAP derivation. A sum of one-patch sparsity costs becomes a product of one-patch priors. Thus the prior knows how to make each individual patch code sparse, but it has no term that can couple neighboring patches.

Remark B.7 (Credit and scope of the argument). This is the spatial analogue of the temporal-prior argument of Lubana et al. (2026). Their result concerns the time index in language-model activations: standard SAEs impose priors that are independent across time. Here the same Bayesian calculation is applied to the patch index of a vision model. The conclusion is therefore spatial: a tokenwise vision SAE imposes an independent prior over patch codes. For the SAE-wise prior characterization, see Appendix I.2 of Lubana et al. (2026), which analyzes ReLU, TopK, BatchTopK, and JumpReLU SAEs.

The previous proposition used a row-additive penalty for clarity. The same structural point applies more broadly to common SAE variants. ReLU SAEs impose a sparsity prior through an ℓ_1 penalty. TopK and JumpReLU SAEs impose sparsity through constraints or pseudo-hard thresholding. BatchTopK couples examples through a batch-level sparsity rule. These variants differ in how they shape sparsity inside each code vector, but none of these tokenwise mechanisms introduces a spatial edge interaction between neighboring patches.

The same limitation can also be seen from a symmetry perspective. If the objective only contains rowwise reconstruction and rowwise sparsity terms, then it cannot distinguish the original patch layout from a layout in which all patch rows have been permuted together.

This is Proposition 3.1 in the main text, which we restate and prove here. Let $\mathbf{P} \in \mathbb{R}^{p\times p}$ be a permutation matrix, and write $\mathcal{L}_{\text{SAE}}(\mathbf{Z}, \mathbf{D}; \mathbf{H})$ for the tokenwise objective. If the reconstruction and sparsity terms are row-additive, then

$$\mathcal{L}_{\text{SAE}}(\mathbf{PZ}, \mathbf{D}; \mathbf{PH}) = \mathcal{L}_{\text{SAE}}(\mathbf{Z}, \mathbf{D}; \mathbf{H}),$$

so a tokenwise SAE objective does not distinguish a coherent patch layout from an arbitrarily permuted layout, provided the activations and codes are permuted together.

Proof of Proposition 3.1. The Frobenius reconstruction term is invariant under left multiplication by a permutation matrix:

$$\|\mathbf{PH} - \mathbf{PZD}\|_F^2 = \|\mathbf{P}(\mathbf{H} - \mathbf{ZD})\|_F^2 = \|\mathbf{H} - \mathbf{ZD}\|_F^2 .$$

The row-additive sparsity term is also unchanged because $\mathbf{P}$ only reorders rows:

$$\sum_i \mathcal{R}((\mathbf{PZ})_i) = \sum_i \mathcal{R}(\mathbf{z}_i).$$

Combining the two equalities proves the claim. □

Proof intuition. A permutation changes the names and order of the patch rows, but not the multiset of patch losses. Since the vanilla objective contains no graph term, it has no way to know which patches were neighbors before the permutation.

This subsection explains what was hidden in the previous one. Sparse coding is the MAP formalization of the noisy LRH model, and vanilla SAEs approximate that sparse-coding estimator through amortized inference. However, the Bayesian reading shows that the vanilla objective also adds a factorized prior over indexed codes. For vision, this means an independent prior over patch codes. The next subsection replaces that spatially factorized prior with a Markov-field prior over neighboring patch codes.

B.3 THE MF-LRH PRIOR

The previous subsection identified the factorized prior supplied by the vanilla tokenwise objective. The MF-LRH of Definition 3.2 replaces that choice with local interactions on the patch graph, while retaining the linear representation $\mathbf{H} \approx (\mathbf{S} + \mathbf{R})\mathbf{D}$. The smooth field models spatially persistent content, and the innovation field models local departures. Both fields are continuous, with attractive interactions for $\mathbf{S}$ and stabilized repulsive interactions for $\mathbf{R}$.

For the patch graph of Definition A.1, write $\kappa_{\mathcal{G}} = \lambda_{\max}(\mathbf{L})$ and $\widetilde{\mathbf{L}} = \kappa_{\mathcal{G}}\mathbf{I}_p - \mathbf{L}$. By the trace identities in Lemma A.3, the prior in Equation (3) is $p(\mathbf{S}, \mathbf{R} \mid \mathcal{G}) \propto \exp[-\mathcal{J}_{\mathbf{L}}(\mathbf{S}, \mathbf{R})]$, where

$$\mathcal{J}_{\mathbf{L}}(\mathbf{S}, \mathbf{R}) = \lambda_S \|\mathbf{S}\|_1 + \lambda_R \|\mathbf{R}\|_1 + \frac{\mu}{2}\left[\mathrm{Tr}(\mathbf{S}^\top \mathbf{L}\mathbf{S}) + \mathrm{Tr}(\mathbf{R}^\top \widetilde{\mathbf{L}}\mathbf{R})\right], \quad \lambda_S, \lambda_R > 0, \quad \mu \geq 0. \tag{6}$$

The first quadratic penalizes variation between neighbors. The second favors contrast at a fixed field magnitude, with the shift by $\kappa_{\mathcal{G}}$ keeping its energy nonnegative. The ℓ_1 terms relax the hard sparsity condition of LRH: the continuous prior can yield sparse posterior modes, but its samples are not exactly sparse. Proposition B.8 below verifies normalizability and the interaction signs on both the real code space and the nonnegative orthant.

Proposition B.8 (Attraction and repulsion in MF-LRH)**.** *Consider the continuous prior in Equation (3), with $\lambda_S, \lambda_R > 0$, $\mu \geq 0$, symmetric nonnegative graph weights, and $\kappa_{\mathcal{G}} = \lambda_{\max}(\mathbf{L})$, where each undirected edge is counted once. The fields $\mathbf{S}, \mathbf{R} \in \mathbb{R}^{p\times c}$ range over the full real domain, or both are restricted to the nonnegative orthant, and the ℓ_1 penalties are entrywise. The prior is normalizable, and its smooth and innovation fields are respectively attractive and repulsive pairwise* MRF*s, coordinatewise in the dictionary-code basis.*

Shared content and local departures. In MF-LRH, these preferences encourage the two fields to specialize. The smooth field $\mathbf{S}$ explains content shared across patches, such as objects, materials, and regions, so attraction encourages its features to persist spatially. The innovation field $\mathbf{R}$ explains local departures from that shared content. For example, neighboring patches of a wall suggest a common background, while patches intersecting a thin pole require an additional local explanation. An innovation feature that activates at the pole and falls away in the surrounding wall expresses precisely this spatial contrast.
Together with sparsity, stabilized repulsion encourages the innovation stream to concentrate on such departures. The spatial penalties thus favor persistent structure in $\mathbf{S}$ and local variation in $\mathbf{R}$, guiding how information is allocated between the streams. This is a modeling preference for complementary explanations; reconstruction supplies the evidence for which innovations are needed.

Proof. The prior energy separates between streams and across feature coordinates. For each feature coordinate a, the scalar edge energies are

$$P_{ij}^S(s_{ia}, s_{ja}) = \frac{\mu w_{ij}}{2}(s_{ia} - s_{ja})^2, \qquad P_{ij}^R(r_{ia}, r_{ja}) = -\frac{\mu w_{ij}}{2}(r_{ia} - r_{ja})^2.$$

Their mixed derivatives satisfy

$$\frac{\partial^2 P_{ij}^S}{\partial s_{ia}\,\partial s_{ja}} = -\mu w_{ij} \leq 0, \qquad \frac{\partial^2 P_{ij}^R}{\partial r_{ia}\,\partial r_{ja}} = \mu w_{ij} \geq 0.$$

Thus the smooth edge energies are submodular and the innovation edge energies are supermodular. The remaining contributions are unary:

$$U_i^S(s_{ia}) = \lambda_S |s_{ia}|, \qquad U_i^R(r_{ia}) = \lambda_R |r_{ia}| + \frac{\mu\kappa_{\mathcal{G}}}{2} r_{ia}^2.$$

These scalar unary terms are modular and do not change either classification. Each feature coordinate therefore has attractive smooth interactions and repulsive innovation interactions on $\mathcal{G}$. Grouping coordinates by patch preserves the pairwise Markov structure. Zero interactions, including the case $\mu = 0$, satisfy both inequalities.

It remains to verify normalizability. By Lemma A.3, the combined quadratic contribution to the prior energy is

$$Q(\mathbf{S}, \mathbf{R}) = \frac{\mu}{2}\left[\mathrm{Tr}(\mathbf{S}^\top \mathbf{L}\mathbf{S}) + \mathrm{Tr}\big(\mathbf{R}^\top(\kappa_{\mathcal{G}}\mathbf{I}_p - \mathbf{L})\mathbf{R}\big)\right].$$

The Laplacian term is nonnegative by its squared-difference identity, and the shifted term is nonnegative by Corollary A.4. Since $\mu \geq 0$, we have $Q(\mathbf{S}, \mathbf{R}) \geq 0$. Writing the full energy as $\mathcal{J}_{\mathbf{L}}(\mathbf{S}, \mathbf{R}) = \lambda_S \|\mathbf{S}\|_1 + \lambda_R \|\mathbf{R}\|_1 + Q(\mathbf{S}, \mathbf{R})$, the unnormalized density satisfies

$$0 < \exp\big(-\mathcal{J}_{\mathbf{L}}(\mathbf{S}, \mathbf{R})\big) \leq \exp(-\lambda_S \|\mathbf{S}\|_1 - \lambda_R \|\mathbf{R}\|_1).$$

The upper bound is integrable over the full real domain and hence over the nonnegative orthant. Strict positivity gives a positive normalizing constant, so $0 < Z < \infty$ on either domain. □

B.4 FROM MF-LRH TO SPATIAL-SAE

We now keep the Gaussian observation model from Section B.2 and derive the inference objective under the MF-LRH prior. We then obtain the practical SPATIAL-SAE objective by amortizing this estimator and establish that its graph dependence is invariant to arbitrary relabelings of the patches.

Proof of Theorem 4.1. Fix the dictionary $\mathbf{D}$ and the externally specified graph $\mathcal{G}$. Under $\mathbf{H} = (\mathbf{S} + \mathbf{R})\mathbf{D} + \mathbf{\Psi}$ with independent $\boldsymbol{\psi}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)$, Bayes' rule gives the negative log posterior, up to a constant independent of the fields, as

$$E_{\mathbf{L}}(\mathbf{S}, \mathbf{R}; \mathbf{H}, \mathbf{D}) := \frac{1}{2\sigma^2} \|\mathbf{H} - (\mathbf{S} + \mathbf{R})\mathbf{D}\|_F^2 + \mathcal{J}_{\mathbf{L}}(\mathbf{S}, \mathbf{R}). \tag{7}$$

The independent Gaussian likelihood supplies the reconstruction term, and Equation (6) supplies the prior term. The MAP estimator therefore minimizes $E_{\mathbf{L}}$. Multiplying this energy by $\sigma^2 > 0$ preserves its minimizers. Absorbing that factor into λ_S, λ_R, and μ gives exactly Equation (4). □

Amortizing the spatial estimator. As in Definition B.3, an encoder replaces a separate optimization for each image by a shared prediction, $(\mathbf{S}_{\boldsymbol{\theta}}, \mathbf{R}_{\boldsymbol{\theta}}) = F_{\boldsymbol{\theta}}(\mathbf{H}; \mathbf{L})$. Substituting these fields into the rescaled posterior energy and averaging over images gives the SPATIAL-SAE objective in Equation (5). The dictionary is learned with unit-norm atoms, $\|\mathbf{d}_a\|_2 = 1$. The encoder restricts the available codes, so its predictions approximate the spatial MAP estimator with a possible amortization gap, just as in vanilla sparse coding.

The dictionary partition in Section 4.2 restricts the two fields to complementary coordinate blocks, with zeros elsewhere. This gives $|\mathrm{supp}(\mathbf{z}_i)| = |\mathrm{supp}(\mathbf{s}_i)| + |\mathrm{supp}(\mathbf{r}_i)|$; such additivity need not hold in general. The exact MAP derivation above uses positive ℓ_1 coefficients. TopK and BatchTopK instead enforce sparsity through the encoder; Section D.1 details these variants and the width normalization of the spatial penalties.

Graph layout and patch indexing. The preceding derivation makes the graph dependence of the objective explicit. We now prove the symmetry claim from Section 4.1: this dependence is on the spatial relations encoded by the graph rather than on the arbitrary indices assigned to its vertices.

Proposition B.9 (Spatial layout and patch indexing)**.** *Let $E_{\mathbf{L}}$ be the objective in Equation (7), with $c \geq 1$ and the loop-free patch graph of Definition A.1. The fields range over the real code space or the nonnegative orthant. For any permutation matrix $\mathbf{P} \in \mathbb{R}^{p \times p}$, simultaneously relabeling the activations, fields, and graph preserves the energy:*

$$E_{\mathbf{PLP}^\top}(\mathbf{PS}, \mathbf{PR}; \mathbf{PH}, \mathbf{D}) = E_{\mathbf{L}}(\mathbf{S}, \mathbf{R}; \mathbf{H}, \mathbf{D}).$$

If $\mu > 0$ and the graph is held fixed, the identity

$$E_{\mathbf{L}}(\mathbf{PS}, \mathbf{PR}; \mathbf{PH}, \mathbf{D}) = E_{\mathbf{L}}(\mathbf{S}, \mathbf{R}; \mathbf{H}, \mathbf{D})$$

holds for every activation matrix and pair of fields if and only if $\mathbf{P}^\top \mathbf{L} \mathbf{P} = \mathbf{L}$. Equivalently, $\mathbf{P}$ preserves weighted adjacency: $\mathbf{P}^\top \mathbf{W} \mathbf{P} = \mathbf{W}$.

Proof. Permutation of rows preserves the Frobenius reconstruction loss and the entrywise ℓ_1 penalties. Relabeling the graph also preserves its spectrum, so for $\mathbf{L}' = \mathbf{PLP}^\top$,

$$\widetilde{\mathbf{L}}' = \kappa_{\mathcal{G}} \mathbf{I}_p - \mathbf{L}' = \mathbf{P} \widetilde{\mathbf{L}} \mathbf{P}^\top.$$

For either $\mathbf{A} = \mathbf{L}$ or $\mathbf{A} = \widetilde{\mathbf{L}}$, orthogonality of $\mathbf{P}$ gives

$$\mathrm{Tr}\big((\mathbf{PX})^\top (\mathbf{PAP}^\top)(\mathbf{PX})\big) = \mathrm{Tr}(\mathbf{X}^\top \mathbf{AX}).$$

This proves invariance under simultaneous relabeling. With the graph fixed, $\mathbf{P}^\top \mathbf{L} \mathbf{P} = \mathbf{L}$ also implies $\mathbf{P}^\top \widetilde{\mathbf{L}} \mathbf{P} = \widetilde{\mathbf{L}}$, so both spatial penalties are unchanged.

Conversely, suppose the fixed-graph identity holds for every pair of fields. Set $\mathbf{R} = 0$ and let a single column of $\mathbf{S}$ be $\mathbf{x}$, with all other columns zero. The reconstruction and sparsity terms are invariant under the simultaneous row permutation. Since $\mu > 0$, equality of the spatial penalties requires

$$\mathbf{x}^\top (\mathbf{P}^\top \mathbf{L} \mathbf{P} - \mathbf{L}) \mathbf{x} = 0$$

for every admissible $\mathbf{x}$. Testing the coordinate vectors $\mathbf{e}_i$ and their sums $\mathbf{e}_i + \mathbf{e}_j$ forces every entry of this symmetric matrix to vanish. These tests are valid on both domains. Finally, for a loop-free graph, the off-diagonal entries of $\mathbf{L}$ are $-w_{ij}$ and its diagonal entries are the weighted degrees. Thus preservation of $\mathbf{L}$ is equivalent to preservation of weighted adjacency. □

At $\mu = 0$, all row permutations preserve the objective even with the graph fixed. For $\mu > 0$, the proposition characterizes symmetries of the whole objective; particular fields may still have equal energy under additional permutations. Together with the MAP derivation, this explains how SPATIAL-SAE uses spatial relations without depending on patch indexing.

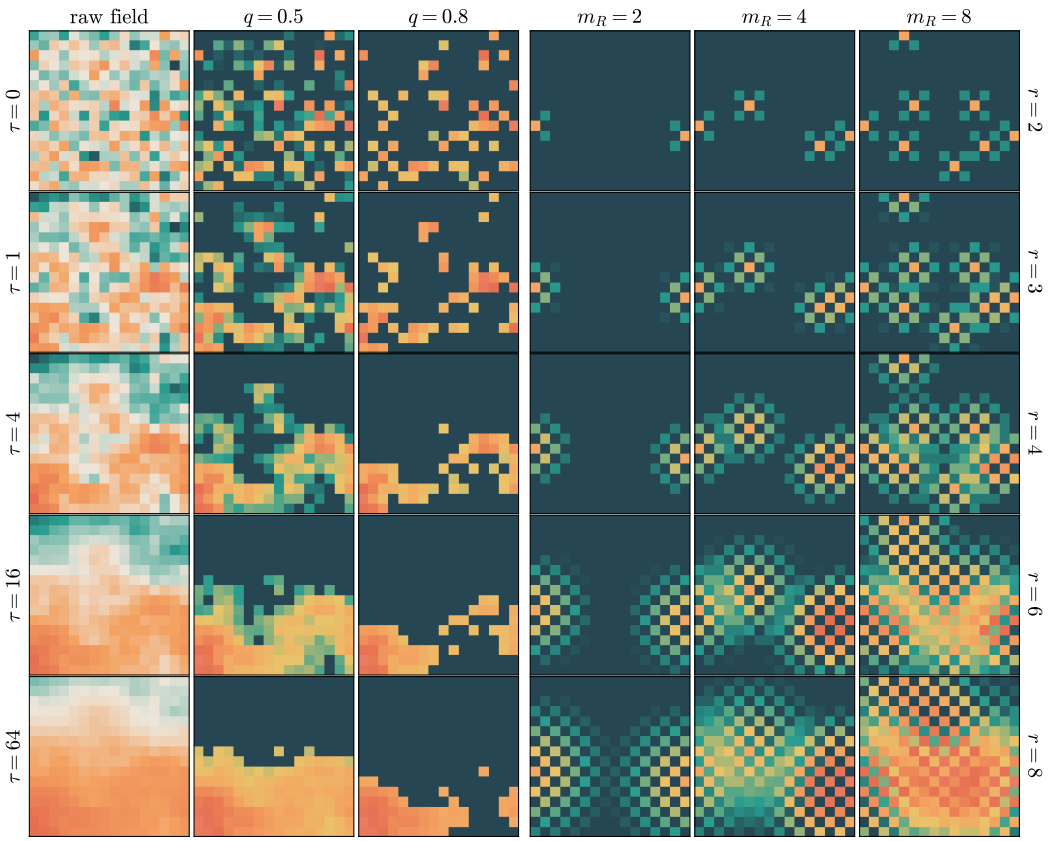

Figure 7: **Controlling smooth- and innovation-stream structure.** Left: rows increase the coupling τ, making the raw field progressively smoother, while the two thresholded columns show how a larger support quantile q reduces its active area. Right: rows increase the packet radius r, expanding each innovation packet, while columns increase the number m_R of active innovation concepts, producing more packets and greater overlap. Random draws and packet centers are held fixed across comparable panels.

C Synthetic Experiments

Real vision-model activations do not reveal the dictionary or latent codes that generated them, so they cannot directly tell us whether an SAE recovers the underlying concepts. We therefore construct a controlled benchmark with known concept directions and firing patterns. We describe the benchmark in Section C.1 and report the component, geometry, and capacity results in Section C.2.

C.1 Experimental Setup

The synthetic activations contain smooth concepts and sparse local innovations, matching the qualitative structure observed in real vision representations (as discussed in Section 3.1), without being sampled from the exact MF-LRH prior. This tests whether the spatial prior remains useful when the data exhibit the relevant structure without matching its assumed distribution exactly.

Concept dictionary and data dimensions. We add spatial structure to existing synthetic SAE benchmarks (Chanin & Garriga-Alonso, 2026) as follows: for each dataset, we sample c concept directions independently from $\mathcal{N}(0, \mathbf{I}_d)$, normalize them to unit length, and fix the resulting dictionary $\mathbf{D} \in \mathbb{R}^{c \times d}$ across images in the linear observation model of Theorem 4.1. Each image has $p = 256$ patches on a 16×16 grid, giving $\mathbf{H} \in \mathbb{R}^{p \times d}$ and smooth and innovation codes $\mathbf{S}, \mathbf{R} \in \mathbb{R}^{p \times c}_{\geq 0}$. We use the unweighted four-neighbor grid graph and denote its combinatorial Laplacian by $\bar{\mathbf{L}}$. Each experiment therefore has activation dimension d, true dictionary width c, and learned SAE width w.

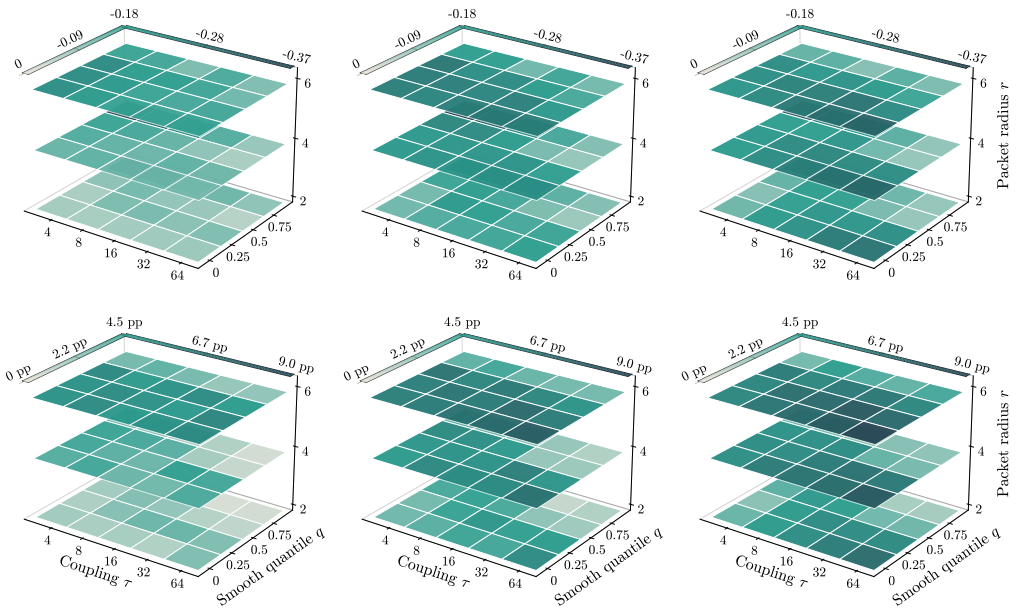

Figure 8: **Geometry sweep relative to B.** Differences between **DT** and **B** across the 60 geometry cells. Columns show L, M, and H; the upper row shows the polysemanticity difference and the lower row the F1 difference in percentage points. Each horizontal slice fixes r, while the in-plane axes vary τ and q. Teal denotes an improvement, while the pale reference color denotes no change. The predominantly teal surfaces show that **DT** improves substantially over **B** throughout the grid.

We vary the superposition ratio $\rho = c/d$, equivalently the overcompleteness of the true dictionary, and narrowness $\nu = w/c$ independently. Since we fix $c = 256$ and vary d, increasing ρ packs the same number of true concepts into a lower-dimensional activation space, while decreasing ν restricts the number of learned atoms available to recover them.

Smooth and innovation streams. For each image, we sample an active set $\mathcal{A}_S \subseteq [c]$ uniformly without replacement, with $|\mathcal{A}_S| = m_S$. For each $a \in \mathcal{A}_S$, we sample $\boldsymbol{\psi}_a^{(S)} \sim \mathcal{N}(0, \mathbf{I}_p)$, diffuse it with $\mathbf{T}_\tau = (\mathbf{I}_p + \tau \mathbf{L})^{-1}$, rectify it, and retain the values above its empirical q-quantile. Writing

$$\mathbf{g}_a = (\mathbf{T}_\tau \boldsymbol{\psi}_a^{(S)})_+, \qquad \mathbf{u}_a = \mathbf{g}_a \odot \mathbb{1}\{\mathbf{g}_a > Q_q(\mathbf{g}_a)\},$$

we set $\mathbf{S}_{:,a} = \rho_S \mathbf{u}_a / \|\mathbf{u}_a\|_\infty$, where $\rho_S > 0$ is the smooth-stream peak amplitude. In practice, we clamp the normalization denominator from below at $\epsilon = 10^{-8}$ and set columns outside $\mathcal{A}_S$ to zero. The left half of Figure 7 shows how τ controls spatial coherence and q controls support area. When the threshold is positive and there are no ties, approximately $1 - q$ of patches remain active. At $\tau = 0$, the sampler has no preference for adjacent patches.

For the innovation stream, we independently sample an active set $\mathcal{A}_R \subseteq [c]$ uniformly without replacement, with $|\mathcal{A}_R| = m_R$. We assign each $a \in \mathcal{A}_R$ a grid center c_a, sampled independently and uniformly from the p patches, with replacement. Let $\delta(i, c_a)$ denote the Euclidean distance from patch i to that center. Writing (x_i, y_i) for the grid coordinates of patch i, define

$$b_i^{(a)} = \mathbb{1}\{(x_i - x_{c_a}) + (y_i - y_{c_a}) \equiv 0 \pmod 2\},$$

so that $\mathbf{b}^{(a)} \in \{0,1\}^p$ indicates the side of the grid's bipartition containing c_a. A compact triangular envelope

$$e_{ia} = \left(1 - \frac{\delta(i, c_a)}{r}\right)_+$$

is restricted by this mask, giving $R_{ia} = \rho_R b_i^{(a)} e_{ia}$ for active concepts and zero otherwise, where $\rho_R > 0$ is the innovation-stream peak amplitude. Hence r controls the spatial extent of each packet

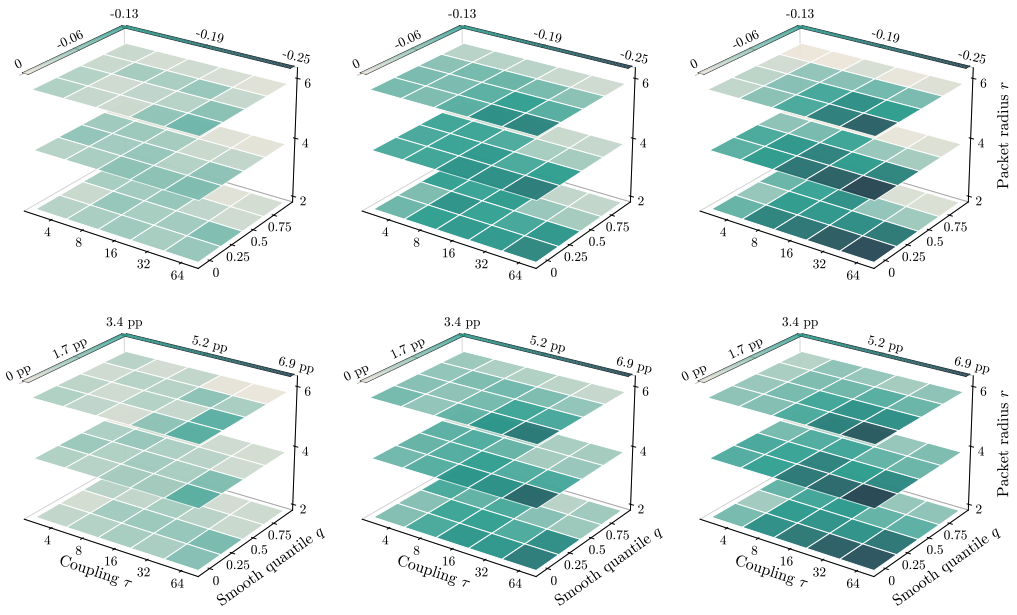

Figure 9: **Geometry sweep relative to D.** Differences between **DT** and **D** across the 60 geometry cells. Columns show L, M, and H; the upper row shows the polysemanticity difference and the lower row the F1 difference in percentage points. Each horizontal slice fixes r, while the in-plane axes vary τ and q. Teal denotes an improvement, while the pale reference color denotes no change. The variation across L, M, and H shows that the benefit over **D** depends on the trace strength.

and the bipartite mask gives the packet its locally alternating structure. Figure 4 illustrates the construction of the smooth and innovation streams side by side, while the right half of Figure 7 visualizes the effects of r and m_R. Because the two streams are sampled independently, their active concept sets and spatial supports may overlap. Both streams are mapped through the same dictionary $\mathbf{D}$. Finally, we form

$$\mathbf{H} = (\mathbf{S} + \mathbf{R})\mathbf{D} + \mathbf{\Psi}, \qquad \mathbf{\Psi}_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2 \mathbf{I}_d).$$

Access levels and sweeps. We compare three models at matched learned width and sparsity. The baseline **B** is a standard single-stream SAE trained on the combined activations. The decomposed model **D** is given the separate smooth and innovation streams but uses no spectral regularization. Finally, **DT** has the same inputs and architecture as **D** and adds the trace penalties of Equation (5).

	D > B	**DT > B**			**DT > D**			**DT > D > B**		
		L	M	H	L	M	H	L	M	H
MCC	99%	100%	99%	89%	96%	83%	64%	96%	82%	64%
Poly.	99%	100%	99%	89%	96%	80%	64%	94%	79%	63%
F1	98%	100%	96%	87%	91%	85%	79%	89%	83%	77%
$\mathbf{R}^2$	13%	10%	1%	0%	15%	1%	0%	4%	0%	0%

L, M, and H denote $\mu_S = 3 \times 10^{-4}$, 10^{-3}, and 3×10^{-3}, respectively.

Table 2: **Ablating SPATIAL-SAE components.** Each entry is the strict win rate of the indicated comparison across the combined Geometry and Capacity Sweeps, rounded to the nearest integer. The no-trace **D** > **B** column isolates decomposition. The remaining groups introduce spectral regularization: **DT** > **D** isolates its incremental effect, **DT** > **B** measures the combined effect, and the final group requires both improvements.

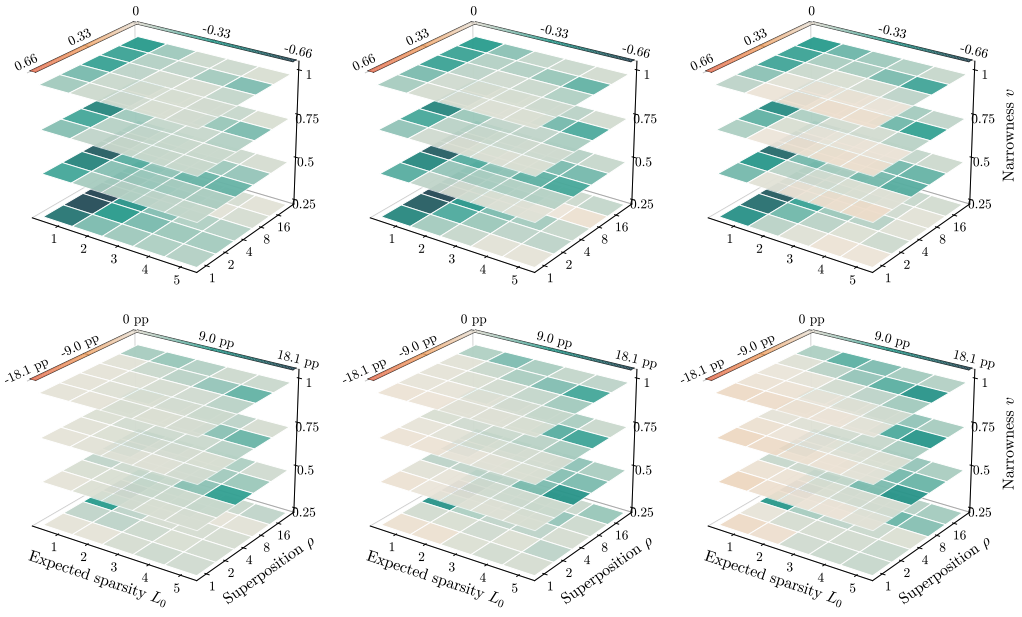

Figure 10: **Capacity sweep relative to B.** Differences between **DT** and **B** across the 100 capacity cells. Columns show L, M, and H; the upper row shows the polysemanticity difference and the lower row the F1 difference in percentage points. Each horizontal slice fixes ν, while the in-plane axes vary L_0 and ρ. Teal denotes an improvement and orange a deterioration.

Consequently, **B** $\rightarrow$ **D** isolates the effect of access to the decomposition, whereas **D** $\rightarrow$ **DT** isolates the effect of the trace prior. In every **DT** experiment, we set $\mu_R = \mu_S/3$.

We use two configuration grids. The first varies the geometry of both latent streams over

$$\tau \times q \times r = \{4, 8, 16, 32, 64\} \times \{0, 0.25, 0.5, 0.75\} \times \{2, 4, 6\}.$$

We exclude $q = 1$ because thresholding above the empirical maximum would set the entire smooth stream to zero. Throughout this geometry grid, we fix the activation dimension at $d = 32$, the superposition ratio at $\rho = 8$, the narrowness at $\nu = 0.5$, the active-concept counts at $m_S = 9$ and $m_R = 4$, and the observation noise at $\sigma = 0.2$. For each cell, we estimate the true mean per-patch L_0 and use its nearest integer as the sparsity target. The second varies the sparsity target, activation dimension, and learned dictionary narrowness over the 100-cell capacity grid

$$L_0 \times d \times \nu = \{1, 2, 3, 4, 5\} \times \{16, 32, 64, 128, 256\} \times \{0.25, 0.5, 0.75, 1\}.$$

Throughout this capacity grid, we fix the true dictionary width at $c = 256$, the spatial geometry at $(\tau, q, r) = (16, 0.5, 4)$, the active-concept counts at $m_S = 9$ and $m_R = 4$, and the observation noise at $\sigma = 0.2$. The swept values of d and ν therefore give

$$\rho = \frac{c}{d} \in \{16, 8, 4, 2, 1\}, \qquad w = \nu c \in \{64, 128, 192, 256\}.$$

The geometry and capacity grids share the default configuration $(\tau, q, r) = (16, 0.5, 4)$ and $(L_0, d, \nu) = (4, 32, 0.5)$. Their union therefore contains $60 + 100 - 1 = 159$ distinct configurations.

We run three matched sweeps:

1. **Sparsifier Generalization Sweep.** On all 60 geometry cells, we compare **DT** with **B** using ReLU, TopK, BatchTopK, and JumpReLU, with five seeds per cell and $\mu_S = 10^{-3}$, giving 2,400 trained SAEs and 300 matched comparisons per sparsifier. The results are presented in Table 1.
2. **Geometry Sweep.** On the same 60 geometry cells, we use only TopK and a single shared seed to compare **B**, **D**, and **DT**. For **DT**, we use three trace strengths, denoted L, M, and H, corresponding to $\mu_S = 3 \times 10^{-4}$, 10^{-3}, and 3×10^{-3}, respectively.
3. **Capacity Sweep.** We repeat the preceding single-seed comparison on all 100 capacity cells.

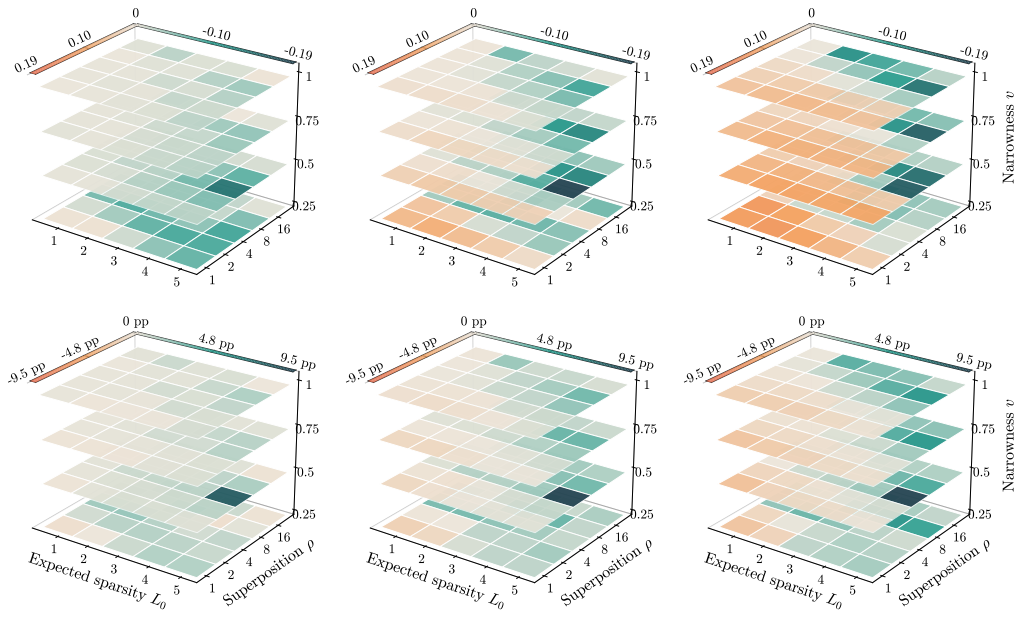

Figure 11: **Capacity sweep relative to D.** Differences between **DT** and **D** across the 100 capacity cells. Columns show L, M, and H; the upper row shows the polysemanticity difference and the lower row the F1 difference in percentage points. Each horizontal slice fixes ν, while the in-plane axes vary L_0 and ρ. Teal denotes an improvement and orange a deterioration.

Training and evaluation. Unless stated otherwise, models are trained for 5,000 Adam steps with batches of 16 images and decoder normalization after each update. Matched models share the generator configuration and seed and are evaluated on the same 2,000 held-out images. We build our implementation on SynthSAEBench (Chanin & Garriga-Alonso, 2026) and reuse its implementations of MCC, support F1, and held-out R^2. We port the polysemanticity metric (POLY.) from Mencattini et al. (2026). A strict increase counts as a win for MCC, F1, and R^2, and a strict decrease for polysemanticity; ties are not wins. The full-ladder event **DT** $>$ **D** $>$ **B** requires both strict improvements in the same configuration.

C.2 Effects of Decomposition and Spectral Regularization

Component ablation. Table 2 summarizes the Geometry and Capacity Sweeps as strict win rates over their 159 distinct configurations. The recovery metrics exhibit a clear ladder: giving the model access to the decomposition is already highly beneficial, and adding spectral regularization on top yields further gains. Specifically, **D** beats **B** in 99%, 99%, and 98% of configurations on MCC, polysemanticity, and F1, respectively. At the low trace strength, **DT** then beats **D** in 96%, 96%, and 91% of configurations, so the full **DT** $>$ **D** $>$ **B** ladder holds in 96%, 94%, and 89%. The ladder weakens as the trace strength increases, while R^2 rarely follows it, emphasizing that the trace coefficient must be tuned for recovery rather than reconstruction.

Geometry Sweep. Across changes in smooth coupling, support, and packet radius, the trace prior provides a robust benefit (Figures 8 and 9). Against **B**, **DT** improves both polysemanticity and F1 throughout the geometry grid at every trace strength, producing the predominantly teal surfaces in Figure 8. This advantage persists after accounting for the decomposition: against **D**, F1 improves throughout, while polysemanticity improves across the full grid at L and M. At H, the polysemanticity benefit becomes less uniform, but the remaining cells show no change rather than a deterioration. Thus, once the data have the intended spatial structure, spectral regularization provides a substantial and robust gain beyond decomposition alone.

Capacity Sweep. Across changes in sparsity, superposition, and narrowness, the benefits of the trace prior depend more strongly on its strength (Figures 10 and 11). At L, **DT** improves over **B** on both plotted metrics in 100% of configurations. By H, however, it performs worse on polysemanticity in 18% of configurations and on F1 in 20%. This sensitivity becomes more pronounced after accounting for the decomposition: against **D**, the high trace strength worsens polysemanticity in 49% of configurations and F1 in 33%, making the mean polysemanticity change slightly unfavorable. Unlike in the Geometry Sweep, a mistuned trace coefficient can therefore erase the advantage over the baseline and become actively detrimental once the decomposition is already available.



Figure 12: **Spatial annotations for concept evaluation.** Images from ADE20K (left) and COCO-STUFF (right), paired with their semantic segmentation masks. The annotations identify the spatial extent of objects and scene regions, allowing us to evaluate whether learned features activate on the concepts they represent.

D EXPERIMENTS WITH NATURAL IMAGES

The natural-image experiments in Section 5.2 test whether spatial structure improves the concepts recovered from vision-model activations. This appendix provides the complete experimental account. We first describe the datasets, frozen backbone, patch graph, and compute environment. In Section D.1, we detail optimization, model architecture, sparsification, and spatial regularization. Section D.2 defines the evaluation metrics and spectral diagnostics, while Section D.3 describes the automated-interpretability protocol. Finally, Section D.4 reports the ablations, model-selection procedure, transfer to COCO-STUFF, and additional qualitative results.

Datasets and backbone. Judging whether an atom captures a concept requires ground truth that is localized in space: an image-level label cannot tell whether an atom fires on the object it is named after or on its surroundings. We therefore evaluate on densely annotated scene-parsing datasets, where every pixel carries a semantic label, and use ADE20K (Zhou et al., 2019) and COCO-STUFF (Caesar et al., 2018). Figure 12 shows example images and their segmentation annotations. ADE20K contains 25,574 training images spanning 365 scene categories and thousands of annotated object categories; its derived SceneParse150 benchmark uses 20,210 training images and 150 semantic classes. COCO-STUFF augments the 118,287 COCO train2017 images with dense pixel-level annotations over 171 semantic classes, plus an unlabeled/void class. Both datasets mix thing classes (person, car) with stuff classes (sky, wall, road), and a typical image contains several concepts at different scales. Concepts therefore occupy both large uniform regions and small or thin structures, which is the mix of smooth and high-frequency content that the two-stream decomposition targets. The pixel-level masks let us assign concepts to individual patches, so the same annotations support patch-level F1, region coherence over annotated segments, and the concept spectra used as spectral references. The two datasets also differ in scale, taxonomy, and domain, so COCO-STUFF tests whether configurations selected on ADE20K transfer without re-tuning. We extract activations from a frozen DINOv2 ViT-B/14 with four register tokens (Oquab et al., 2024; Darcet et al., 2024).[4] ADE20K is one of the semantic segmentation benchmarks used by Oquab et al. (2024) to evaluate frozen patch features with a linear classifier. A 224×224 input yields a 16×16 patch grid, so each image provides $p = 256$ patch activations with dimension $d = 768$. All spatial models use a fixed, unweighted graph connecting distinct patches whose Manhattan distance is at most two, with the same graph used for every image. As in Fel et al. (2026), we take the activations of the final layer after normalization; unlike them, we keep only the 256 patch tokens, discarding the CLS and register tokens, and store the patch activations in half precision (`float16`).

Software and compute. We extend the `overcomplete` library (version 0.3.0) to implement SPATIAL-SAE.[5] The environment uses `Python` 3.12.12, `PyTorch` 2.12.1 with `CUDA` 12.6, `torchvision` 0.27.1, `datasets` 3.6.0, and `timm` 1.0.27. Each run uses a single `NVIDIA`

[4] `timm` model: `vit_base_patch14_reg4_dinov2` (pretrained).

[5] https://github.com/KempnerInstitute/overcomplete

$\mathbb{E}[\mathbf{L_0}]$	Model	TopK		Vanilla		JumpReLU		BatchTopK	
		R^2	Norm. F1	R^2	Norm. F1	R^2	Norm. F1	R^2	Norm. F1
8	Baseline	61.1 ± 0.2	58.0 ± 1.4	40.8 ± 0.8	51.5 ± 1.0	38.4 ± 0.7	52.9 ± 0.7	57.9 ± 0.9	54.2 ± 0.5
	SPATIAL-SAE	45.1 ± 0.1	67.0 ± 1.2	27.7 ± 0.5	55.7 ± 2.0	30.4 ± 0.3	57.2 ± 1.2	43.9 ± 0.3	60.2 ± 1.6
16	Baseline	70.4 ± 0.2	57.5 ± 1.1	53.7 ± 0.7	50.5 ± 1.0	52.2 ± 0.8	52.1 ± 0.5	69.2 ± 0.9	55.9 ± 0.7
	SPATIAL-SAE	52.4 ± 0.3	65.4 ± 1.9	39.1 ± 0.7	55.5 ± 2.0	41.4 ± 0.3	56.7 ± 1.6	52.4 ± 0.4	60.3 ± 2.0
32	Baseline	78.1 ± 0.2	56.1 ± 0.7	66.1 ± 0.6	49.1 ± 0.7	67.5 ± 0.5	49.8 ± 0.3	78.0 ± 0.2	54.9 ± 0.7
	SPATIAL-SAE	57.5 ± 0.1	63.0 ± 0.5	51.1 ± 0.2	52.3 ± 2.0	51.6 ± 0.8	54.3 ± 2.0	58.8 ± 0.4	56.6 ± 0.9
64	Baseline	83.3 ± 0.1	52.6 ± 0.3	74.4 ± 0.6	44.4 ± 0.4	76.6 ± 0.5	43.7 ± 0.5	83.5 ± 0.2	52.3 ± 0.8
	SPATIAL-SAE	61.1 ± 0.3	61.0 ± 2.3	60.5 ± 0.7	48.0 ± 1.0	58.6 ± 0.2	51.2 ± 1.4	63.8 ± 0.3	55.9 ± 1.1
128	Baseline	87.3 ± 0.1	48.3 ± 0.5	80.3 ± 0.4	37.0 ± 0.6	82.7 ± 0.4	36.7 ± 0.5	87.4 ± 0.3	48.3 ± 1.0
	SPATIAL-SAE	64.9 ± 0.1	58.9 ± 0.7	66.1 ± 0.5	44.8 ± 3.8	62.6 ± 0.5	52.3 ± 1.9	68.4 ± 0.3	55.0 ± 0.9
256	Baseline	91.2 ± 0.1	48.0 ± 0.4	85.4 ± 0.2	27.8 ± 0.4	88.3 ± 0.3	29.4 ± 0.4	91.4 ± 0.4	48.3 ± 1.0
	SPATIAL-SAE	69.1 ± 0.1	57.3 ± 1.6	71.0 ± 0.2	45.4 ± 2.5	67.4 ± 0.7	52.2 ± 1.7	73.9 ± 0.3	54.8 ± 1.2

Table 3: **Sparsity-budget ablation on ADE20K.** We measure reconstruction with R^2 and concept recovery with Normalized F1, relative to the corresponding linear probe. Entries are mean $\pm$ standard deviation over five seeds and are reported as percentages. We use an expected sparsity budget of 16: compared with a budget of 8, it improves baseline reconstruction for every sparsifier while retaining at least 98% of each baseline's best Normalized F1, and is itself optimal for BatchTopK. SPATIAL-SAE has higher Normalized F1 throughout the sweep.

`A100-SXM4 GPU` with 80 GB of memory. On ADE20K, ten training epochs take approximately four minutes for static and convolutional spatial models, ten minutes for context attention, and two to three minutes for baselines. On COCO-STUFF, context attention takes 28–35 minutes and linear baselines take six to eight minutes. These times cover training epochs.

D.1 IMPLEMENTATION DETAILS

Optimization. We train all models for ten epochs with Adam, learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. The learning rate increases linearly from zero over the first 100 steps, then decays to 10^{-5} with a cosine schedule. We clip the global gradient norm at 1. Baselines receive batches of 4096 shuffled patch activations; SPATIAL-SAE receives 16 complete patch grids, also totaling 4096 patches. Baselines use float16 autocasting, while spatial models use bfloat16 to avoid overflow in the Laplacian term; steps with a non-finite loss are skipped. Data loading uses four workers, but the order in which batches are consumed is fixed by the seed and independent of worker scheduling: shards are prefetched concurrently and yielded in a precomputed shuffled order, and images within a shard are shuffled with a seeded generator. We do not enforce deterministic GPU kernels. We use the final checkpoint from each run, without early stopping or selection across epochs. Configuration sweeps use seed 0, and the selected models are retrained with seeds 0–4.

Encoders and dictionaries. Encoders and dictionaries are the `MLPEncoder` and `DictionaryLayer` classes of `overcomplete`, which we describe here for completeness. We instantiate the encoder with zero hidden blocks (the `linear` preset), so it applies a single linear projection followed by ReLU, after which the chosen sparsifier is applied. Unless stated otherwise, all models use $c = 3072$ dictionary atoms, which are renormalized to unit ℓ_2 norm on every forward pass.

For SPATIAL-SAE, the dictionary partition in Section 4 assigns c_S atoms to the smooth stream and c_R to the innovation stream, with corresponding decoder blocks $\mathbf{D}_S$ and $\mathbf{D}_R$. Here we write $\mathbf{S} \in \mathbb{R}^{p \times c_S}$ and $\mathbf{R} \in \mathbb{R}^{p \times c_R}$ for the fields' active code blocks. These are sparsified with stream budgets k_S and k_R, enforced by the chosen sparsifier exactly as k is for the baseline. Each stream has its own encoder, $\mathbf{E}_S \in \mathbb{R}^{d \times c_S}$ and $\mathbf{E}_R \in \mathbb{R}^{d \times c_R}$, applied to its respective input. The resulting codes decode through the corresponding blocks $\mathbf{D}_S$ and $\mathbf{D}_R$. To match the baselines, the two streams share the total dictionary and budget, $c_S + c_R = c$ and $k_S + k_R = k$, and atoms are allocated in proportion to budget, $c_S/c = k_S/k$, with largest-remainder rounding. A 14:2 budget split therefore assigns 2688 atoms to the smooth stream and 384 to the innovation stream. We sweep the seven splits 2:14, 4:12, 6:10, 8:8, 10:6, 12:4, and 14:2.

Algorithm 2 Static forward pass.

Require: Patch activations $\mathbf{H} \in \mathbb{R}^{p \times d}$.
1: **for** each patch i **do**
2: $\quad \mathbf{S}_i \leftarrow \textsc{Sparsify}(\mathbf{H}_i \mathbf{E}_S)$
3: $\quad \mathbf{R}_i \leftarrow \textsc{Sparsify}((\mathbf{H}_i - \mathbf{S}_i \mathbf{D}_S)\mathbf{E}_R)$
4: $\quad \widehat{\mathbf{H}}_i \leftarrow \mathbf{S}_i \mathbf{D}_S + \mathbf{R}_i \mathbf{D}_R$
5: **end for**
6: **return** $\mathbf{S}$, $\mathbf{R}$, and $\widehat{\mathbf{H}}$

Algorithm 3 Convolutional forward pass.

Require: Patch activations $\mathbf{H} \in \mathbb{R}^{p \times d}$.
1: **for** each patch i **do**
2: $\quad \widetilde{\mathbf{H}}_i \leftarrow \sum_{j \in \mathcal{N}(i)} \mathbf{W}_{j-i} \odot \mathbf{H}_j$
3: $\quad \mathbf{S}_i \leftarrow \textsc{Sparsify}(\widetilde{\mathbf{H}}_i \mathbf{E}_S)$
4: $\quad \mathbf{R}_i \leftarrow \textsc{Sparsify}((\mathbf{H}_i - \mathbf{S}_i \mathbf{D}_S)\mathbf{E}_R)$
5: $\quad \widehat{\mathbf{H}}_i \leftarrow \mathbf{S}_i \mathbf{D}_S + \mathbf{R}_i \mathbf{D}_R$
6: **end for**
7: **return** $\mathbf{S}$, $\mathbf{R}$, and $\widehat{\mathbf{H}}$

Controls for smooth-field estimation. The static control (left) removes neighbor-based estimation, while the convolutional control (right) aggregates neighboring patches without the content-dependent weighting used by context attention. Both retain the same two-stream innovation and decoding structure, allowing the ablation to isolate how the smooth field is estimated.

Controlling effective sparsity and dead atoms. Unless stated otherwise, we target an expected sparsity of 16 active features per patch; Table 3 reports the full sparsity-budget sweep. Comparisons at a fixed budget require models to use a similar number of active features per patch. TopK enforces the budget directly by retaining the k largest coefficients. BatchTopK applies a shared threshold, set to the (nk)-th largest pre-activation in a batch of n patches; at evaluation, it uses an exponential moving average of the training thresholds with momentum 0.9.

For ReLU and JumpReLU, the number of active features depends on the ℓ_1 penalty, so the same penalty coefficient need not yield the same sparsity across models. We therefore use a simplified variant of the target-L_0 controller of Chanin & Garriga-Alonso (2026), without its gain scheduling, to adjust the coefficient $\lambda_0 \eta$, with $\lambda_0 = 10^{-2}$. Let $\bar{L}_0$ be an exponential moving average of the number of active features per patch, with smoothing coefficient 0.99. Starting at step 100, we update the multiplier after each optimization step as

$$\eta \leftarrow \eta \left[1 + k_i \tanh\left(g \frac{\bar{L}_0 - k}{k}\right)\right], \qquad k_i = 3 \times 10^{-3}, \quad g = 10,$$

and clip η to $[10^{-2}, 10^{2}]$. The penalty increases when $\bar{L}_0 > k$ and decreases when $\bar{L}_0 < k$. For SPATIAL-SAE, each stream has its own controller with target k_S or k_R. JumpReLU additionally learns a threshold for each atom, initialized at e^{-3}, through a straight-through estimator with a Silverman kernel of bandwidth 0.1. In every final run targeting $\mathbb{E}[L_0] = 16$, the mean number of active features per patch on the evaluation pool differs from the target by at most one.

We also use the reanimation regularizer implemented in the `overcomplete` library (Fel et al., 2025) to encourage atoms that never fire in a batch to become active. We subtract their summed pre-activations from the loss, normalized by the batch size and dictionary width. The penalty has weight 10^{-3} for each spatial stream and 5×10^{-3} for baselines; each stream uses its own width for normalization. Fewer than 0.4% of atoms are inactive on the evaluation pool in every final run targeting $\mathbb{E}[L_0] = 16$.

Estimating the smooth field. The encoder in Section 4.2 estimates the smooth content at each patch from its neighbors, without observing the target patch's activation. It projects the neighbors' mean activation to form a query and their individual activations to form keys and values. Attention then combines this information into the smooth code. We call this estimator *context attention* in the ablations below. We arrange the neighbors in 12 fixed positions, add learned relative-position embeddings to the keys and values, and use a learned vector for positions outside the grid. The attention layer has four heads and inner width $c_S/4$, with LayerNorm applied to its inputs. Its output is added to the query, then rectified and sparsified.

We compare this estimator with two alternatives to examine the role of neighboring activations and how they are combined. The *static* estimator obtains the smooth code directly by projecting and sparsifying the target patch's own activation. The *convolutional* estimator applies a depthwise convolution over 12 neighbors with a learned kernel $\mathbf{W} \in \mathbb{R}^{12 \times d}$, before projecting and sparsifying the

result. The kernel weights each coordinate separately, so the convolution combines spatial information without mixing coordinates. We use zero padding and initialize all kernel entries to $1/12$. The row $\mathbf{W}_\delta$ corresponds to relative offset δ. Algorithms 2 and 3 give both forward passes, using the encoders and dictionary blocks defined above and the fixed graph's edge set $\mathcal{E}$.

All three estimators use the same innovation step: the innovation encoder receives $\mathbf{H} - \mathrm{sg}(\mathbf{S}\mathbf{D}_S)$, with gradients stopped through the smooth reconstruction.

Spectral regularization and objective. We implement a width-normalized variant of the objective in Definition 4.2, using separate dictionary blocks for the smooth and innovation streams. For activations $\mathbf{H} \in \mathbb{R}^{p\times d}$, let $(\mathbf{S}_{\boldsymbol{\theta}}, \mathbf{R}_{\boldsymbol{\theta}}) = F_{\boldsymbol{\theta}}(\mathbf{H}\,;\mathbf{L})$ and $\widehat{\mathbf{H}}_{\boldsymbol{\theta}} = \mathbf{S}_{\boldsymbol{\theta}}\mathbf{D}_S + \mathbf{R}_{\boldsymbol{\theta}}\mathbf{D}_R$. We suppress the encoder subscript when defining the loss components. The reconstruction and sparsity losses are

$$\mathcal{L}_{\mathrm{rec}}(\mathbf{H}, \widehat{\mathbf{H}}) = \frac{1}{pd}\left\|\mathbf{H} - \widehat{\mathbf{H}}\right\|_F^2,$$
$$\mathcal{L}_{\ell_1}(\mathbf{S}, \mathbf{R}; \lambda_S, \lambda_R) = \frac{\lambda_S}{p}\left\|\mathbf{S}\right\|_1 + \frac{\lambda_R}{p}\left\|\mathbf{R}\right\|_1.$$

Thus reconstruction is averaged over activation entries, and each sparsity penalty is averaged over patches. For ReLU and JumpReLU, the controller described above adjusts λ_S and λ_R independently. TopK and BatchTopK enforce sparsity through the encoder and set $\lambda_S = \lambda_R = 0$. The normalizability result in Proposition B.8 applies to the continuous prior with positive ℓ_1 coefficients. The TopK and BatchTopK implementations instead use encoder-enforced sparsity with the same spatial regularizers.

Let $\mathbf{L}$ be the Laplacian of the fixed patch graph used throughout the experiments, and define $\widetilde{\mathbf{L}} = \lambda_{\max}(\mathbf{L})\mathbf{I}_p - \mathbf{L}$. The spectral loss is

$$\mathcal{L}_{\mathrm{spec}}(\mathbf{S}, \mathbf{R}; \mathbf{L}, \mu) = \mu\left[\frac{\mathrm{Tr}(\mathbf{S}^\top\mathbf{L}\mathbf{S})}{c_S} + \frac{\mathrm{Tr}(\mathbf{R}^\top\widetilde{\mathbf{L}}\mathbf{R})}{c_R}\right].$$

The two terms penalize high-frequency energy in the smooth stream and low-frequency energy in the innovation stream, respectively, with each normalized by its dictionary width. Here μ weights these normalized terms; setting $\mu = 0$ removes both penalties. When $c_S \neq c_R$, normalization changes the relative weighting of the two spectral penalties compared with Equation (5). These positive rescalings preserve the attraction/repulsion signs and nonnegativity of the quadratic terms. We compute the eigendecomposition of $\mathbf{L}$ once and reuse it for regularization and spectral metrics.

The complete training objective is

$$\min_{\boldsymbol{\theta},\mathbf{D}} \frac{1}{|\mathcal{X}|}\sum_{\mathbf{H}\in\mathcal{X}}\Big[\mathcal{L}_{\mathrm{rec}}(\mathbf{H}, \widehat{\mathbf{H}}_{\boldsymbol{\theta}}) + \mathcal{L}_{\ell_1}(\mathbf{S}_{\boldsymbol{\theta}}, \mathbf{R}_{\boldsymbol{\theta}}; \lambda_S, \lambda_R) + \mathcal{L}_{\mathrm{spec}}(\mathbf{S}_{\boldsymbol{\theta}}, \mathbf{R}_{\boldsymbol{\theta}}; \mathbf{L}, \mu)\Big] + \text{ reanim.} \tag{8}$$

D.2 Evaluation Metrics and Spectral Diagnostics

Because the concepts that generate real vision-model activations are unobserved, no single score can establish SAE quality. We therefore evaluate three complementary properties: reconstruction fidelity, alignment of individual features with annotated concepts, and spatial coherence of the complete code. We then use graph-frequency diagnostics to characterize the energy distribution of the learned codes relative to the true concept masks.

Evaluation protocol. All metrics are computed on held-out data. For an activation matrix $\mathbf{H} \in \mathbb{R}^{n\times d}$, let $\widehat{\mathbf{H}}$ be its reconstruction and let $\mathbf{Z} \in \mathbb{R}^{n\times c}$ be its sparse code. For Spatial-SAE, $\mathbf{Z}$ concatenates the smooth and innovation codes, and feature-level metrics consider features from both streams. We first aggregate each metric over its evaluation samples and then average the resulting scores across training seeds.

Reconstruction fidelity. We measure the fraction of activation variance preserved by the reconstruction. For a batch of patch activations, let $\boldsymbol{\mu}_{\mathbf{H}} = n^{-1}\mathbf{H}^\top\mathbf{1}$ be the mean activation and let

Monosemanticity rubric (integer score from 1 to 10)

How consistently is the active region the same object or concept across the seven rows?

10	One clear concept; every row fits.
8–9	One clear concept; all rows fit except one near miss of that same concept.
6–7	One concept covers most rows; the remainder are unrelated or ambiguous.
4–5	About half the rows share a concept, or two distinct groups have comparable size.
2–3	Only a loose thematic resemblance, or three or more distinct groups.
1	No discernible commonality.

Rubric 1: **Monosemanticity scoring rubric.** Scores measure whether one coherent concept explains the highlighted regions across the panel.

$\overline{\mathbf{H}} = \mathbf{1}\boldsymbol{\mu}_{\mathbf{H}}^{\top}$ repeat that mean at every patch. We define

$$R^2(\mathbf{H}, \widehat{\mathbf{H}}) = 1 - \frac{\left\|\mathbf{H} - \widehat{\mathbf{H}}\right\|_F^2}{\left\|\mathbf{H} - \overline{\mathbf{H}}\right\|_F^2}. \tag{9}$$

We compute this coefficient separately for each evaluation batch, average it across batches, and report the result as a percentage.

Concept alignment. A feature aligns with a concept when its activation mask agrees with the concept's segmentation mask. For an arbitrary image, let $\mathbf{Y} \in \{0,1\}^{p \times r}$ contain the patch-aligned masks of the r semantic classes and let $\mathbf{Z} \in \mathbb{R}^{p \times c}$ be its sparse code. We binarize all feature maps at their native firing threshold, $\mathbf{A} = \mathbb{1}\{\mathbf{Z} > 0\}$. For class ℓ and feature j, we collect the columns $\mathbf{Y}_{:\ell}$ and $\mathbf{A}_{:j}$ across the images that contain class ℓ, concatenate them across patches, and compute

$$\mathrm{F1}_{\ell j} = \frac{2\mathrm{TP}_{\ell j}}{2\mathrm{TP}_{\ell j} + \mathrm{FP}_{\ell j} + \mathrm{FN}_{\ell j}}. \tag{10}$$

Each class is represented by its best-aligned feature, $j_\ell^\star = \arg\max_j \mathrm{F1}_{\ell j}$. Concept F1 macro-averages these class-best scores:

$$\text{Concept F1} = \frac{100}{|\mathcal{C}|} \sum_{\ell \in \mathcal{C}} \mathrm{F1}_{\ell j_\ell^\star}. \tag{11}$$

Here $\mathcal{C}$ is the set of annotated classes represented in the evaluation data. We refer to this aggregate as concept F1, or simply F1 in figures and tables.

Normalized concept F1. Concept F1 depends partly on whether the frozen backbone makes a class linearly accessible at all. We therefore train a linear probe on the same backbone activations and compute its class-wise F1 against the same masks. For the classes that are reliably decoded by the probe, $\mathcal{C}_{\mathrm{dec}} = \{\ell : \mathrm{F1}_\ell^{\mathrm{probe}} > 0.6\}$, we report the ratio between the class-best feature and probe scores:

$$\text{Normalized F1} = \frac{100}{|\mathcal{C}_{\mathrm{dec}}|} \sum_{\ell \in \mathcal{C}_{\mathrm{dec}}} \frac{\mathrm{F1}_{\ell j_\ell^\star}}{\mathrm{F1}_\ell^{\mathrm{probe}}}. \tag{12}$$

Normalized F1, sometimes abbreviated *Norm. F1* in figures and tables, thus measures how much of a class's linearly decodable information is captured by a single SAE feature.

Region coherence. Concept F1 assesses individual features; region coherence asks whether the complete sparse code respects the spatial extent of annotated concepts. We align each segmentation mask to the patch grid, discard unlabeled and boundary patches, and group the remaining patches by semantic region. For an arbitrary image, let $\mathcal{P}$ be the retained patch indices and let $\mathbf{Z}_{\mathcal{P},:}$ be their

Map-quality rubric (integer score from 1 to 10)

How well does the activation map model the object or concept named in the description, across the seven rows? Judge **coverage** of the whole object, **tightness** at its extent, and **consistency** across rows jointly.

10	In essentially every row, the active region covers the object completely and tightly.
8–9	The object is covered well in most rows, with only small spill or gaps.
6–7	The map is on the object in most rows but is clearly partial or loose.
4–5	The map is on the object about half the time and is otherwise diffuse, scattered, or on other things.
2–3	The object is visible, but the map mostly misses it or covers the whole image.
1	No usable localization, such as a whole-image response, border artifact, or noise; alternatively, no consistent object exists to model.

The target is the concept inferred by the rater, not the ground-truth segmentation class. When no consistent concept exists, map quality is judged against whatever the description nevertheless identifies.

Rubric 2: **Map-quality scoring rubric.** Scores measure the coverage, tightness, and consistency of the highlighted regions relative to the concept inferred by the rater.

code matrix. For SPATIAL-SAE, this matrix concatenates the two streams. We center its columns across patches,

$$\widetilde{\mathbf{Z}} = \left(\mathbf{I}_{|\mathcal{P}|} - \frac{1}{|\mathcal{P}|}\mathbf{1}\mathbf{1}^\top\right)\mathbf{Z}_{\mathcal{P},:}. \tag{13}$$

We compare every pair of nonzero rows of $\widetilde{\mathbf{Z}}$ by cosine similarity and group pairs by their grid distance. Within each image, we average similarities at distance δ separately for pairs from the same and different semantic regions. We then average both quantities across images, denoting the results by $s_{\text{within}}(\delta)$ and $s_{\text{between}}(\delta)$. Region coherence is their average matched-distance difference,

$$\text{Coherence} = \frac{100}{|\Delta|}\sum_{\delta\in\Delta}\left[s_{\text{within}}(\delta) - s_{\text{between}}(\delta)\right], \tag{14}$$

where Δ contains the grid distances represented by both pair types. We average within-region similarities equally across regions and between-region similarities equally across region pairs before averaging over images. Matching distance removes the confound that patches from the same region are generally closer; a larger score indicates that the code more faithfully preserves semantic regions.

Concept-relative code spectra (Figure 6b and Figure 15b). The preceding metrics measure SAE performance; we next examine how code energy is distributed across spatial scales. Our hypothesis predicts a specific spectral signature of concept fragmentation: when a spatially coherent concept is split across patch-local features, the code allocates more energy to high graph frequencies than the corresponding concept masks. Comparing the two spectra therefore tests whether the learned code reflects the spatial scales present in the annotated concepts.

Let $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the eigendecomposition of the combinatorial Laplacian of the patch graph, with graph-Fourier modes ordered by increasing eigenvalue. For an arbitrary image, let $\mathbf{Z} \in \mathbb{R}^{p\times c}$ be its code matrix and let $\mathbf{M} \in \mathbb{R}^{p\times r}$ contain its patch-aligned concept masks. The graph-Fourier energy across all features and concepts is

$$\mathbf{e}_{\text{code}}(\mathbf{Z}) = \text{diag}\left(\mathbf{U}^\top\mathbf{Z}\mathbf{Z}^\top\mathbf{U}\right), \tag{15}$$

$$\mathbf{e}_{\text{concept}}(\mathbf{M}) = \text{diag}\left(\mathbf{U}^\top\mathbf{M}\mathbf{M}^\top\mathbf{U}\right). \tag{16}$$

Here diag returns the diagonal as a vector, so each entry records the energy assigned to one graph-Fourier mode. We compute these vectors for every evaluation image and sum them componentwise across images. Denote the resulting dataset-level vectors by $\mathbf{e}_{\text{code}}$ and $\mathbf{e}_{\text{concept}}$; the former includes features from both SPATIAL-SAE streams. After discarding their constant-mode entries, let $\mathbf{e}^{\circ}_{\text{code}}$

and $\mathbf{e}^{\circ}_{\text{concept}}$ denote the remaining vectors. We normalize them to unit mass,

$$\boldsymbol{\pi}_{\text{code}} = \frac{\mathbf{e}^{\circ}_{\text{code}}}{\mathbf{1}^{\top}\mathbf{e}^{\circ}_{\text{code}}}, \qquad \boldsymbol{\pi}_{\text{concept}} = \frac{\mathbf{e}^{\circ}_{\text{concept}}}{\mathbf{1}^{\top}\mathbf{e}^{\circ}_{\text{concept}}}.$$

Let $\mathcal{S}$ denote the common smoothing operator applied to these discrete distributions. The plotted concept-relative spectrum is the vector

$$\mathbf{r} = \log_2(\mathcal{S}(\boldsymbol{\pi}_{\text{code}}) \oslash \mathcal{S}(\boldsymbol{\pi}_{\text{concept}})), \tag{17}$$

where both $\log_2$ and $\oslash$ act elementwise. The entry associated with mode k is plotted at normalized rank $q_k = k/p$. Zero indicates that codes and concept masks allocate the same fraction of nonconstant energy to a spatial frequency; positive values indicate excess code energy, and negative values indicate under-representation. We average $\mathbf{r}$ across training seeds and use its componentwise standard deviation for the uncertainty bands.

Per-feature frequency and dictionary visualization (Figure 6d and Figure 15d). The concept-relative spectrum summarizes the code as a whole, but does not show how spatial frequency is distributed across individual features or decoder directions. We therefore assign each feature an empirical graph frequency. For an arbitrary image with code matrix $\mathbf{Z} \in \mathbb{R}^{p\times c}$, we first remove each feature's spatial mean,

$$\widetilde{\mathbf{Z}} = \left(\mathbf{I} - \frac{1}{p}\mathbf{1}\mathbf{1}^{\top}\right)\mathbf{Z}. \tag{18}$$

Its per-feature Dirichlet and nonconstant energies are the vectors

$$\mathbf{d}(\widetilde{\mathbf{Z}}) = \operatorname{diag}(\widetilde{\mathbf{Z}}^{\top}\mathbf{L}\widetilde{\mathbf{Z}}), \qquad \mathbf{a}(\widetilde{\mathbf{Z}}) = \operatorname{diag}(\widetilde{\mathbf{Z}}^{\top}\widetilde{\mathbf{Z}}). \tag{19}$$

We compute both vectors for every evaluation image and sum them componentwise, obtaining the aggregated vectors $\mathbf{d}_{\text{agg}}$ and $\mathbf{a}_{\text{agg}}$. For features with nonzero accumulated energy, and with $\lambda_{\max}$ denoting the largest eigenvalue of the patch-graph Laplacian, the frequency vector is

$$\mathbf{f} = \frac{1}{\lambda_{\max}}\mathbf{d}_{\text{agg}} \oslash \mathbf{a}_{\text{agg}}, \tag{20}$$

where $\oslash$ denotes elementwise division. Each entry is the normalized, activation-energy-weighted mean graph frequency of one feature. Low values indicate spatially smooth activation maps, whereas high values indicate rapidly varying maps.

D.3 Automated Interpretability

Concept F1 measures alignment with a predefined semantic class, but does not establish whether a feature represents one coherent concept across its strongest examples or whether its activation maps localize that concept cleanly. We therefore complement concept F1 with an automated interpretability evaluation on both ADE20K and COCO-Stuff. The two evaluations use the same feature selection, panel construction, anonymization, rater, and scoring rubrics; they differ only in their class cohorts and held-out image pools.

Panel construction and scoring. For both datasets, we evaluate the seed-0 baseline and Spatial-SAE runs for all four sparsifiers. For each semantic class and model family, we select that model's own class-best feature $j^{\star}_{\ell}$ from Equation (11), taken from its full code (for Spatial-SAE, the concatenated smooth and innovation streams). We then rank the held-out images by the feature's activation summed over patches and retain the seven highest-ranked images, ordered from strongest to weakest.

Each row of a panel places the 224×224 model input beside the same image with its 16×16 activation map overlaid by nearest-neighbor upsampling. Red opacity is proportional to activation after normalization by the maximum activation in that image, and a dark outline marks the active region. Panels contain no class names, captions, or scores and use randomized class identifiers.

Claude Haiku 4.5 (Anthropic, 2025), at low reasoning effort, views each panel and returns a short description, an integer *monosemanticity* score, and an integer *map-quality* score. The rater is instructed to describe the highlighted regions rather than the whole image and is not told the semantic class, feature F1, model family, or sparsifier. Monosemanticity measures whether one inferred

	Two Streams − Single Stream			Norm. F1
μ	ΔR^2	Δ Norm. F1	Δ Region coh.	Win rate ↑
10^{-3}	-0.80 ± 0.63	$\mathbf{+6.12} \pm 1.86$	-2.90 ± 4.07	**100%**
10^{-2}	-7.25 ± 3.19	$+0.23 \pm 3.33$	-4.49 ± 3.79	57%
10^{-1}	-5.12 ± 2.59	-5.60 ± 4.30	$+5.51 \pm 12.21$	14%

Table 4: **Effect of code decomposition at matched regularization strength.** Static TopK on ADE20K, $\mathbb{E}[L_0] = 16$. The single-stream control applies the attractive Laplacian penalty to the complete code; the two-stream model additionally represents an innovation field with the repulsive penalty. Differences are two-stream minus single-stream at the same μ, reported as mean ± sample standard deviation across seven budget allocations and two initializations each. R^2 and Normalized F1 changes are in percentage points; coherence changes are multiplied by 100. The last column reports the percentage with higher Normalized F1, rounded to the nearest integer. Shading marks the selected regularization strength.

concept explains the active regions across the seven images. Map quality measures the coverage, tightness, and consistency of those regions relative to the concept inferred by the rater. It is therefore a self-consistent localization score rather than agreement with the ground-truth segmentation mask. Both scores use an integer 1–10 scale; the scoring criteria are given in Rubrics 1 and 2. Full panel-generation and rater instructions are included in the supplementary material.

Dataset cohorts. For ADE20K, we begin with all 150 semantic classes and exclude the 18 classes whose best-feature F1 is zero for either model family. This leaves 132 classes for each sparsifier and produces $132 \times 4 \times 2 = 1{,}056$ panels, assigned to 132 eight-panel rater instances.

For COCO-STUFF, we use all 171 official semantic classes. Every class has positive best-feature F1 for both model families under every sparsifier, so no classes are excluded. This produces $171 \times 4 \times 2 = 1{,}368$ panels, assigned to 171 eight-panel rater instances. The panels are drawn from a held-out activation-cache split of 6,218 images, constructed as a seeded random 5% subset of the 152 COCO train2017 shards. This split is not the official COCO val2017 split. Inputs and activation maps retain the same 224×224 and 16×16 geometry used for ADE20K.

Assignment and quality control. Within each dataset, panels are assigned to rater instances using a seeded permutation of the class identifiers. COCO-STUFF uses its own seed-0 permutation, independent of the ADE20K permutation. Every eight-panel instance contains distinct classes and therefore never includes both model families for the same class.

The intended protocol assigns one rating to each panel. Every ADE20K panel received exactly one rating. For COCO-STUFF, the procedure produced 1,370 valid ratings for 1,368 panels because two panels were inadvertently rated twice. Each duplicate pair received identical integer scores, and we average the two ratings before aggregation. No malformed records were produced. The rater was forbidden from accessing other files or outputs; although the storage paths contained model and sparsifier tokens, their meanings were not provided.

Aggregation and uncertainty. For each displayed bar, we average over the corresponding class cohort: 132 classes for ADE20K and 171 classes for COCO-STUFF. We construct 95% percentile intervals from 2,000 class-bootstrap resamples, performed independently for each model family. These intervals quantify variation across semantic classes, not variation across training seeds, ratings, or raters. Scores retain their native 1–10 scale, although the figure axes begin at zero for display. The resulting ADE20K and COCO-STUFF evaluations are reported in Figure 6g–h and Figure 15g–h, respectively.

D.4 Ablations, Model Selection, and Additional Results

SPATIAL-SAE introduces four design choices whose contributions we isolate in turn: the two-stream decomposition, the regularization strength, the smooth-field estimator, and the allocation of the

	Transition	Reconstruction		Normalized F1		Coherence	
Regularization Strength μ	$0 \to 10^{-3}$	-5.52 $[-5.77, -5.09]$	0%	$+6.53$ $[+4.77, +6.94]$	**100%**	$+2.51$ $[-2.69, +4.10]$	86%
	$10^{-3} \to 10^{-2}$	-25.91 $[-29.85, -23.46]$	0%	$+6.75$ $[+2.60, +8.61]$	86%	$+0.13$ $[-1.23, +2.10]$	64%
	$10^{-2} \to 10^{-1}$	-27.94 $[-28.20, -27.23]$	0%	-43.14 $[-47.46, -35.75]$	0%	$+16.04$ $[+7.95, +24.17]$	**100%**
Smooth Field Estimator	S $\to$ C	-3.52 $[-6.90, -1.78]$	0%	-0.34 $[-1.79, +1.01]$	43%	$+2.23$ $[+1.21, +3.00]$	86%
	S $\to$ CA	-2.55 $[-5.16, -1.48]$	0%	$+1.95$ $[+0.09, +5.11]$	71%	$+2.69$ $[+2.14, +3.88]$	93%
	C $\to$ CA	$+0.85$ $[+0.42, +1.55]$	79%	$+2.67$ $[+1.51, +5.23]$	86%	$+0.79$ $[+0.16, +0.94]$	79%

Table 5: **Effects of regularization and smooth-field estimation.** TopK on ADE20K, $\mathbb{E}[L_0] = 16$. Differences are destination minus source, matched by budget allocation and initialization (14 pairs). For each metric, the left column reports the median change (Δ), with its interquartile range beneath; the right reports the win rate (WR), the percentage of positive differences rounded to the nearest integer. R^2 and Normalized F1 changes are in percentage points; coherence changes are multiplied by 100. Regularization transitions use the static encoder; estimator transitions fix $\mu = 10^{-3}$. S: static; C: convolutional; CA: context attention.

sparsity budget between the two streams. Unless stated otherwise, these ablations use ADE20K and a TopK sparsifier with expected sparsity 16 and dictionary width $c = 3072$. Each two-stream setting covers all seven budget allocations, with two parameter initializations per allocation; the single-stream control has no allocation parameter. We report R^2 and Normalized F1 as percentages and multiply region coherence by 100.

Effect of code decomposition under spectral regularization. The gains in Section 5.2 could in principle come from smoothing the code rather than from separating it into smooth and innovation fields. To distinguish these explanations, Table 4 compares SPATIAL-SAE with a single-stream control that applies the same attractive Laplacian penalty to the entire code but has neither an innovation stream nor its repulsive penalty. At the selected strength, $\mu = 10^{-3}$, the decomposition increases Norm. F1 by 6.12 ± 1.86 percentage points on average, and this difference is positive in all 14 matched comparisons. The corresponding mean differences are -0.80 ± 0.63 points in R^2 and -2.90 ± 4.07 points in region coherence. The latter is expected: the single-stream control applies the attractive prior to its entire code and thus directly optimizes all of its features for smoothness, whereas SPATIAL-SAE also maintains an innovation stream under the repulsive prior. The purpose of the decomposition is therefore not to make the entire code maximally smooth, but to improve concept alignment while retaining a separate channel for high-frequency residual structure.

This advantage depends on the regularization regime; at $\mu = 10^{-2}$, the mean Normalized F1 difference narrows to $+0.23$ points, and at 10^{-1} it favors the single-stream control. Even so, sweeping $\mu \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ reveals a higher attainable peak for the decomposed model: the best single-stream setting reaches 66.0% Normalized F1 at $R^2 = 41.9\%$, whereas the best two-stream setting reaches 71.0% at $R^2 = 35.1\%$ (both at $\mu = 10^{-2}$). The decomposition can therefore recover more concept-aligned features, although reaching that peak incurs additional reconstruction cost.

Effect of regularization strength. The upper block of Table 5 varies μ for the two-stream model with the static estimator. Introducing spectral regularization at $\mu = 10^{-3}$ yields median changes of $+6.53$ points in Normalized F1, $+2.51$ points in region coherence, and -5.52 points in R^2. Across the matched pairs, the regularized model has higher Normalized F1 in 100% of comparisons and higher region coherence in 86%, but higher R^2 in none. Increasing the strength from 10^{-3} to 10^{-2} yields median changes of $+6.75$ points in Normalized F1, $+0.13$ points in region coherence, and -25.91 points in R^2; the stronger setting is better on these metrics in 86%, 64%, and 0% of matched comparisons, respectively. Pushing to 10^{-1} degenerates the model: the median changes are -43.14

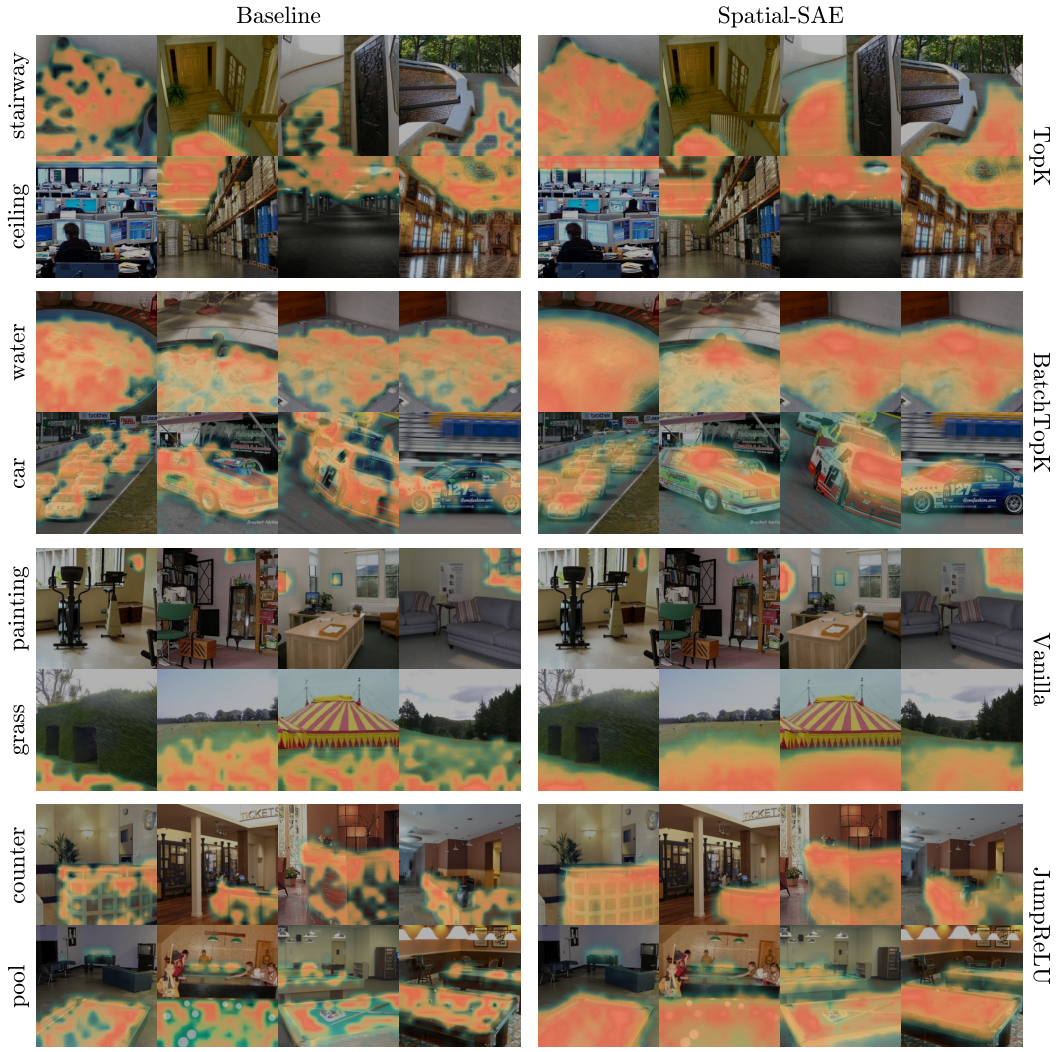

Figure 13: **Class-best feature maps on ADE20K.** For each sparsifier (labels at right), two representative classes (labels at left) compare the baseline (left block) with SPATIAL-SAE (right block). Each model contributes its highest-concept-F1 feature for the labeled class. The four displayed images are selected for high per-image F1 of the baseline feature and then held fixed across models, so the selection deliberately favors the baseline. Warmer colors indicate stronger activation.

points in Normalized F1 and -27.94 points in R^2, with no positive comparisons on either metric, even though region coherence increases by a median 16.04 points and is higher in every comparison. We therefore use $\mu = 10^{-3}$ in subsequent comparisons, choosing the first large and fully consistent improvement in concept alignment without accepting the much steeper reconstruction loss at 10^{-2}.

Effect of the smooth-field estimator. The lower block of Table 5 compares the static, convolutional, and context-attention estimators at $\mu = 10^{-3}$. Relative to the static estimator, convolution yields median changes of -3.52 points in R^2, -0.34 points in Normalized F1, and $+2.23$ points in region coherence. It outperforms the static estimator in 0%, 43%, and 86% of matched comparisons on these three metrics, respectively, so its benefit is largely confined to coherence. Context attention offers a better tradeoff. Relative to the static estimator, its median changes are -2.55 points in R^2, $+1.95$ points in Normalized F1, and $+2.69$ points in coherence; it outperforms the static estimator in 0%, 71%, and 93% of comparisons, respectively. Relative to convolution, all three median differences are positive—$+0.85$, $+2.67$, and $+0.79$ points—and context attention is better in 79%,



Figure 14: **Class-best feature maps on COCO-STUFF.** For each sparsifier (labels at right), two representative classes (labels at left) compare the baseline (left block) with SPATIAL-SAE (right block). Each model contributes its highest-concept-F1 feature for the labeled class. Images are selected for high per-image F1 of the baseline feature and then held fixed across models, so the selection deliberately favors the baseline. Warmer colors indicate stronger activation.

86%, and 79% of comparisons. It therefore provides the most consistent improvement in concept alignment and coherence, and we use it in the final models.

Budget selection for each sparsifier. After fixing context attention and $\mu = 10^{-3}$, we run a separate seven-allocation sweep for each sparsifier on ADE20K, evaluating every split from 2:14 through 14:2 at total budget $k = 16$. We choose from each sweep in two stages: first retain every allocation whose Normalized F1 is within 0.5 percentage points of that sparsifier's best value, then select the candidate with the highest R^2. The procedure selects k_S:k_R = 14:2 for TopK, 12:4 for BatchTopK, 10:6 for ReLU, and 14:2 for JumpReLU. We retrain the selected configurations with seeds 0–4; in the ADE20K sparsity-budget sweep, we preserve these selected proportions while scaling the two stream budgets with the total budget.

Transfer to COCO-STUFF. We transfer the model design and all training choices selected on ADE20K to COCO-STUFF, including the context-attention estimator, $\mu = 10^{-3}$, dictionary width, total sparsity budget, per-sparsifier k_S/k_R allocation, and optimization procedure. Without further

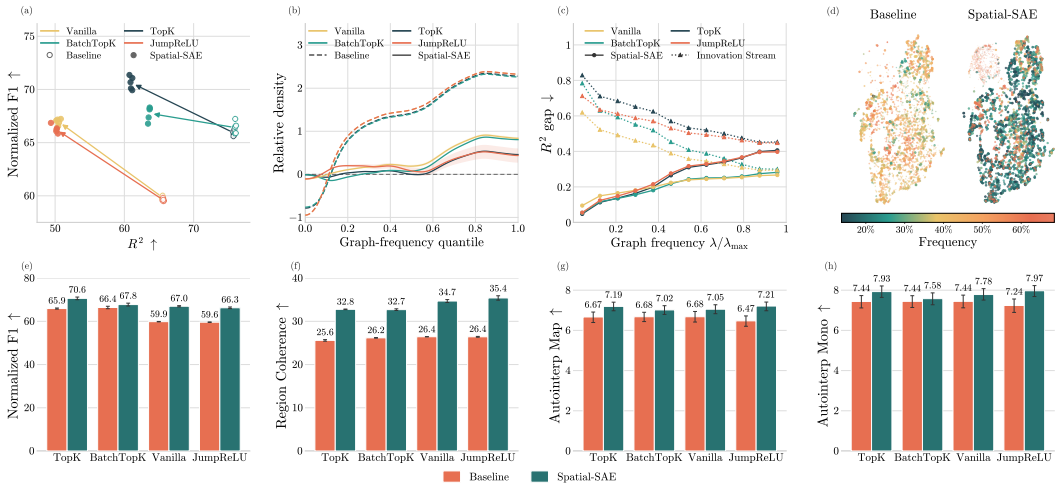

Figure 15: **Quantitative results on COCO-STUFF.** (a) Normalized F1 against reconstruction R^2; points show training seeds and arrows connect model means. (b) Base-2 log ratio between code and concept-mask spectral density; zero denotes agreement, with dashed baseline and solid SPATIAL-SAE curves. (c) The baseline's R^2 advantage over the combined SPATIAL-SAE code (solid) and its innovation stream alone (dotted), resolved by graph frequency. (d) TopK decoder atoms in a joint UMAP, colored by activation-map frequency. (e–h) Normalized F1, region coherence, automated map quality, and monosemanticity. Error bars show one standard deviation across five training seeds in (e–f) and 95% class-bootstrap intervals in (g–h).

tuning on COCO-STUFF, we retrain each transferred configuration from scratch over five random seeds. Under this direct transfer, SPATIAL-SAE improves Normalized F1 for all four sparsifiers by 1.4–7.1 points and region coherence by 6.5–9.0 points (Figure 15). Map quality and monosemanticity also improve in every comparison, by 0.34–0.74 and 0.14–0.73, respectively, on their native 1–10 scales.

Qualitative feature maps. The examples in Figures 13 and 14 are deliberately selected in the baselines' favor. Each family contributes its own class-best feature, but we select the displayed images specifically for high per-image F1 of the *baseline*'s feature and then hold those images fixed for SPATIAL-SAE. This is effectively reverse cherry-picking: the baseline is shown on images where its feature already aligns well with the target, whereas SPATIAL-SAE receives no such selection advantage. Despite this baseline-favoring protocol, SPATIAL-SAE produces visibly more complete and connected maps. On ADE20K, baseline activations for extended regions such as ceiling, water, grass, and pool often contain holes or break into disconnected patches, while SPATIAL-SAE follows a larger continuous portion of the region; the same contrast appears for bounded objects such as cars and counters. On COCO-STUFF, the spatial maps likewise follow more of objects with varied shapes and scales, including cats, giraffes, fruit, zebras, and elephants, instead of isolating a few discriminative fragments. Because image selection is tilted toward the baselines, these qualitative comparisons are a conservative visual counterpart to the aggregate region-coherence and automated-interpretability gains.

A shared spectral signature across datasets. Figure 6b and Figure 15b reveal a remarkably stable failure mode. Across all four sparsifiers and on both datasets, the baseline curves nearly coincide: relative to the concept masks, standard SAE codes underallocate spectral density to the lowest graph frequencies and progressively overallocate it toward higher frequencies. At the upper end of the spectrum, the baseline log ratios exceed 2, corresponding to more than four times the normalized spectral density of the concept masks. SPATIAL-SAE moves every sparsifier substantially toward zero across the spectrum, largely removing the low-frequency deficit and sharply reducing the high-frequency excess. The correction survives tuning-free transfer to COCO-STUFF, showing that it is neither sparsifier- nor dataset-specific: spatial fragmentation has a consistent spectral signature in standard SAEs, and the structured prior directly counteracts it.