WHEN COORDINATES CHANGE THE FEATURES: A SYMMETRY TEST FOR SPARSE AUTOENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

An orthogonal change of coordinates preserves the information in language-model activations, but it can change the features learned by a sparse autoencoder (SAE). We study this dependence with a coupled symmetry test. For TopK SAEs, a rotation of the inputs has an exact counterpart in parameter space that leaves sparse codes and reconstruction loss unchanged. We initialize two SAEs at corresponding parameters and train them on paired minibatches to test whether this correspondence survives training. With Adam, the resulting dictionaries differ substantially, even after latent matching, although their reconstruction quality is similar. The discrepancy begins with Adam's coordinatewise second-moment normalization. We derive a symmetry criterion for adaptive updates and show that group-RMS updates preserve the correspondence in exact arithmetic. In our experiments, these updates reduce dictionary disagreement at comparable reconstruction quality. Thus, differences between SAE dictionaries can come from the coordinates used during training even when the language model and its activation data are fixed. Reconstruction metrics alone do not reveal this dependence. The coupled test measures it against a known correspondence and identifies optimizer choices that respect that correspondence.

1 INTRODUCTION

Sparse autoencoders (SAEs) extract candidate features from language-model activations (Bricken et al., 2023; Cunningham et al., 2024; Templeton et al., 2024; Gao et al., 2024). Researchers name these features, compare them across models, and use them to guide interventions. These uses include model comparison and investigations of model behavior (Jiralerspong & Bricken, 2026; Karvonen et al., 2026). Interpreting a difference between dictionaries therefore requires care: the difference may come from the activations, from choices made during SAE training, or from both.

Reconstruction and sparsity do not settle which features to trust. Better proxy metrics need not mean better practical performance (Karvonen et al., 2025), and SAEs trained on the same data can learn different features (Paulo & Belrose, 2025). Recent work studies how consistently features are learned and whether unstable individual features lie in reproducible subspaces (Song et al., 2026; Gerasimov et al., 2026). Here we test one source of variation by changing the coordinates of a fixed activation dataset. The transformation preserves both the information in the data and the reconstruction objective.

For an orthogonal Q, a TopK SAE with parameters $\theta = (W, b, D, c)$ has an exactly corresponding parameterization

$$T_Q\theta = (WQ^\top,\, b,\, QD,\, Qc). \tag{1}$$

On inputs Qa, it produces identical sparse codes and reconstructions rotated by Q, with the same loss. We use this correspondence to define our *coupled symmetry test* (Figure 1): initialize the two runs at corresponding parameters, use paired minibatches, and check whether training preserves the relationship. We also compare independent starts to put the measured differences in the context of ordinary training variability. As in interpretability sanity checks (Adebayo et al., 2018), the procedure is tested against a controlled change. Here the equivalence of the two training problems gives an exact baseline.

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.

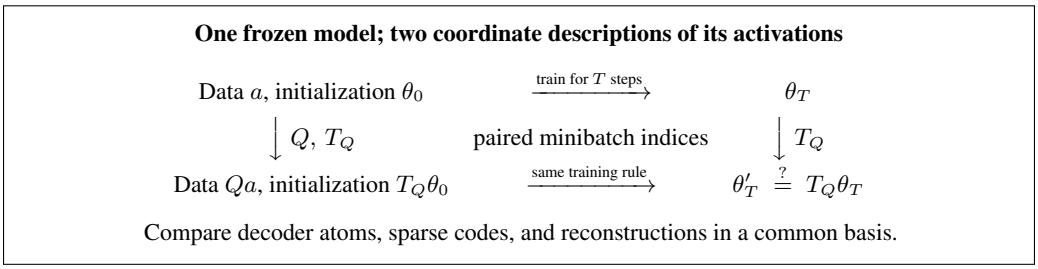


Figure 1: Two training runs describe the same activation data in different coordinates. The model and tokens are fixed, and both the initialization and minibatches correspond exactly. Equivariant training commutes with the parameter transport T_Q. We measure how far the trained pair departs from this correspondence.

Reconstruction quality is similar across the paired runs, but their dictionaries can differ considerably. After 20 epochs, Adam produces decoder mismatch 0.23–0.44 across five GPT-2 Small layers. Optimal latent matching leaves 90–94% of this mismatch. The effect is not limited to random rotations: at layer 6, the PCA eigenbasis gives mismatch 0.306, versus 0.249 for a random dense rotation.

The runs first diverge at Adam's update: its diagonal second moment breaks the correspondence, and the dense-rotation mismatch persists in float64. SGD, momentum SGD, and group-RMS updates are equivariant in exact arithmetic. Their measured decoder mismatch vanishes in the float64 conditions tested. Tensor-RMS Adam also achieves comparable reconstruction quality in the layer-6 sweep and a longer, annealed control. Adam's rotation sensitivity is known in optimization (Zhang et al., 2025); our experiments show how it affects the feature dictionaries used for interpretation. We derive a symmetry criterion for adaptive updates and use it to choose updates that preserve the correspondence.

2 Setup and Objective-Level Symmetry

2.1 Sparse autoencoders

Let $a \in \mathbb{R}^d$ be an activation vector drawn from a distribution $\mathcal{D}$ (here, residual-stream activations of a frozen language model; Section 3). A TopK sparse autoencoder (Makhzani & Frey, 2013; Gao et al., 2024) with m latents has parameters $\theta = (W, b, D, c)$: encoder weights $W \in \mathbb{R}^{m\times d}$ and bias $b \in \mathbb{R}^m$, decoder $D \in \mathbb{R}^{d\times m}$ and bias $c \in \mathbb{R}^d$. It computes

$$z_\theta(a) = \mathrm{TopK}_k\big(\mathrm{ReLU}(Wa + b)\big), \tag{2}$$

$$f_\theta(a) = D\, z_\theta(a) + c, \tag{3}$$

where TopK_k keeps the k largest entries of its input and sets the rest to zero. We call the columns $d_1, \ldots, d_m$ of D the *decoder atoms* and track whether these feature directions agree across runs. Training minimizes the reconstruction loss

$$\mathcal{L}_\mathcal{D}(\theta) = \mathbb{E}_{a\sim\mathcal{D}}\big[\, \|f_\theta(a) - a\|_2^2 \,\big] \tag{4}$$

over the constraint set $\mathcal{C} = \{\theta : \|d_j\|_2 = 1 \text{ for all } j\}$. We enforce this constraint by renormalizing the columns of D after every optimizer step. Since TopK_k imposes sparsity directly, Equation (4) is the full objective.

2.2 Orthogonal reparameterizations of the objective

Let $Q \in \mathbb{R}^{d\times d}$ be orthogonal and $Q_*\mathcal{D}$ the pushforward of $\mathcal{D}$ under $a \mapsto Qa$. The parameter map T_Q in Equation (1) takes an SAE solution in one coordinate system to a solution in the other.

Proposition 1 (Exact transformed solutions). *Let Q be orthogonal and $T_Q\theta = (WQ^\top, b, QD, Qc)$ as in Equation* (1)*. For every θ: (i) $z_{T_Q\theta}(Qa) = z_\theta(a)$ for every $a \in \mathbb{R}^d$; (ii) $f_{T_Q\theta}(Qa) = Qf_\theta(a)$; (iii) $\mathcal{L}_{Q_*\mathcal{D}}(T_Q\theta) = \mathcal{L}_\mathcal{D}(\theta)$ for every $\mathcal{D}$; (iv) T_Q is a bijection of $\mathcal{C}$ onto itself, with inverse $T_{Q^\top}$.*

The key to the proof (Section A) is that the pre-activations $WQ^\top(Qa) + b = Wa + b$ are *identical vectors* in the two coordinate systems. Part (i) therefore holds even at TopK ties. Because T_Q is a loss-preserving bijection of $\mathcal{C}$, the two training problems have identical landscapes up to an isometric reparameterization. This pairs the solutions for $\mathcal{D}$ and $Q_*\mathcal{D}$, but does not determine which solution training will select. Section B checks the construction numerically in float32.

2.3 Algorithmic equivariance

Let $\mathcal{A}_T$ denote the complete training procedure over T steps. This includes the optimizer *and the initialization and update of its internal state*, the learning-rate schedule, decoder renormalization, and the minibatch schedule. We write $\theta_T = \mathcal{A}_T(\mathcal{D}, \theta_0; \xi)$, where ξ is the realized stochasticity, here the minibatch index sequence. In the transformed problem, the same ξ selects minibatches $Q\mathcal{B}_t$.

Definition 1 (Algorithmic equivariance)**.** A training procedure $\mathcal{A}$ is *equivariant* under the orthogonal transform Q if, for every initialization θ_0, every realization ξ, and every horizon T,

$$\mathcal{A}_T\big(Q_*\mathcal{D},\ T_Q\theta_0;\ \xi\big)\ =\ T_Q\,\mathcal{A}_T\big(\mathcal{D},\ \theta_0;\ \xi\big). \tag{5}$$

When Equation (5) holds, the learned dictionaries satisfy $D'_T = QD_T$ column by column and the sparse codes agree along the coupled trajectory. We can therefore compare the pair at fixed latent indices (Section 3). Equivariance here is a *pathwise* property: the correspondence holds at every step for each shared minibatch sequence.

Remark 1 (Scope of the symmetry claim). We apply Q to the SAE's input activations and keep the transformer fixed. PCA preprocessing includes such a change of basis; whitening also rescales the resulting coordinates.

3 The Coupled Symmetry Test

We use the equivalence in Proposition 1 as an experimental control. Starting from θ_0 and $T_Q\theta_0$, we train the two models on paired minibatches $\mathcal{B}_t$ and $Q\mathcal{B}_t$ and measure how far they move from the known correspondence. Latent matching, precision controls, and independent-seed references help distinguish possible causes and assess the size of the discrepancy.

3.1 Data, models, and transforms

Activations. We collect residual-stream activations from a frozen GPT-2 Small (Radford et al., 2019) ($d = 768$), using the inputs to block ℓ for $\ell \in \{0, 3, 6, 9, 11\}$ ($\ell = 0$ is the embedding stream). All layers use the same token rows from 128-token contexts of the Pile (Gao et al., 2020), with $50{,}000$ training and $5{,}000$ evaluation vectors per layer. We shuffle token positions before splitting them, so withheld vectors share contexts with training vectors. We use the withheld positions both to evaluate reconstruction and to select learning rates. The quality comparisons are therefore validation estimates on the collected corpus. To preserve the orthogonal equivalence, we center the activations by the training mean and divide by a *single global* RMS scalar. The experiments cover five layers, up to five seeds, and two numerical precisions.

Transforms. For each seed, we independently draw random signed permutations, Haar rotations, and block rotations. Their error bars therefore include variability across transforms. Every optimizer studied preserves signed permutations (Proposition 4). Dense random rotations mix all coordinates. Block-orthogonal rotations apply an independent Haar rotation within each random block of width w, ranging from sign flips ($w = 1$) to a dense rotation ($w = d$). We measure the density of an orthogonal Q by its mean participation ratio $\frac{1}{d}\sum_i(\sum_j Q_{ij}^4)^{-1}$: the effective number of input coordinates mixed into each output. Two deterministic transforms provide further comparisons: the sign-fixed PCA eigenbasis of the layer-6 training activations and the orthonormal type-II discrete cosine transform (DCT-II). The PCA condition rotates coordinates without whitening's rescaling. We store transforms in float64 (orthogonal to $\sim 10^{-15}$) and cast them to the training precision when applied.

3.2 PAIRED TRAINING AND METRICS

For each condition (layer, seed, transform, optimizer), we sample θ_0, construct $T_Q\theta_0$ exactly, and train both models with identical hyperparameters on paired minibatches $\mathcal{B}_t$ and $Q\mathcal{B}_t$. The models share an index sequence ξ and both start with zero optimizer state (Proposition 2). Unless stated otherwise, we use $m = 1024$, $k = 16$, batch size 2048, and 20 epochs (500 steps) at a constant learning rate. Robustness experiments vary sparsity ($k \in \{8, 64\}$) and width ($m = 2048$, $k = 32$). The core grid, one-step diagnostics, and optimizer ablations use 5 seeds; the remaining multi-seed experiments use 3, as specified in the captions. Tables report mean $\pm$ s.d. unless stated otherwise, and figure captions describe the uncertainty across seeds. Section C lists all hyperparameters, including one layer-specific learning-rate adjustment on the embedding stream. These comparisons concern finite training trajectories at the stated budgets and do not assume convergence. Section E describes the separate 100-epoch schedule control.

Let θ_T, θ'_T be the trained pair and d_j, d'_j their decoder atoms. We compare the parameters directly, then evaluate the codes and reconstructions on the withheld token positions:

$$E_D \;=\; 1 - \frac{1}{m}\sum_{j=1}^{m} \cos\big(Qd_j,\, d'_j\big), \qquad E_z \;=\; \frac{\big\|Z - Z'\big\|_F}{\|Z\|_F}, \qquad E_{\mathrm{rec}} \;=\; \frac{\big\|QF - F'\big\|_F}{\|QF\|_F}, \quad (6)$$

where the columns of Z, Z' are the codes $z_{\theta_T}(a_i), z_{\theta'_T}(Qa_i)$ and those of F, F' the reconstructions. The *decoder mismatch* E_D compares atoms at fixed index j, with no matching step. It directly tests whether training preserves the correspondence in Definition 1. We also report TopK support agreement and the reconstruction quality $\mathrm{FVE}(\theta) = 1 - \sum_i \|f_\theta(a_i) - a_i\|_2^2 / \sum_i \|a_i\|_2^2$ for both runs. If FVE is similar but E_D is large, reconstruction quality alone does not establish feature correspondence. We also measure *movement from initialization*, $\frac{1}{m}\sum_j \cos\big(d_j^{(T)}, d_j^{(0)}\big)$: a dictionary that did not move would preserve the correspondence trivially.

Matched metrics. At a fixed latent index j, mismatch may reflect changed feature directions or a permutation of unchanged directions. Both violate Definition 1, but only the former changes the learned feature *set*. We distinguish these cases by finding the one-to-one assignment π of latents that maximizes $\sum_j \cos(Qd_j, d'_{\pi(j)})$ (Hungarian matching, also used to compare independently trained SAEs). We report the *matched decoder mismatch*

$$E_D^{\mathrm{match}} \;=\; 1 - \frac{1}{m}\sum_{j=1}^{m} \cos\big(Qd_j,\, d'_{\pi(j)}\big) \;\le\; E_D, \quad (7)$$

the *relabeled fraction* $\rho = |\{j : \pi(j) \neq j\}|/m$, and the share of $\sum_j (1 - \cos(Qd_j, d'_j))$ contributed by relabeled latents. For an equivariant procedure, the identity assignment is optimal and $E_D^{\mathrm{match}} = E_D = 0$ in exact arithmetic, although ties can make the optimal assignment nonunique. We recompute the code metrics under π: matched E_z, support agreement, and per-latent activation correlation. Reconstruction mismatch E_{rec} is already invariant to latent relabeling. Each run in the matched-analysis grid saves both final dictionaries and the values of $\cos(Qd_j, d'_j)$, $\cos(Qd_j, d'_{\pi(j)})$, and π for each latent. These metrics compare geometry and reconstruction; they do not evaluate feature semantics or causal roles.

3.3 PRECISION PROTOCOL AND CALIBRATION

Finite-precision drift. The propositions in Section 4 assume exact arithmetic, but most experiments use float32. Rotations change the order of floating-point reductions. Small rounding differences in pre-activations can then change which entries TopK selects, leading to gradient differences that accumulate over training. We use two checks to assess this effect. Sign-flip-only transforms ($w = 1$) leave reduction order unchanged and serve as a numerical control (Section D). We also repeat selected conditions entirely in float64, starting from the same float32-representable initialization. If drift disappears at higher precision, it is consistent with accumulated rounding effects. Persistent drift is evidence of an effect of the update rule.

Matched-quality comparison. We compare feature mismatch at similar reconstruction quality by measuring drift when each run first reaches fixed FVE levels. We also sweep SGD learning rates and train SGD for longer (Section H).

Seed-reference rails. To compare the test's mismatch with ordinary run-to-run variability at the same configuration, we break the coupling in three ways. The transformed model either keeps $T_Q\theta_0$ and uses an independent minibatch order (*order*), starts from an independent initialization $T_Q\tilde{\theta}_0$ with the shared order (*init*), or changes both (*both*; ordinary seed-to-seed variability). The raw model is unchanged in all three cases. We use matched metrics for these comparisons.

4 Optimizer Mechanism and an Equivariant Repair

For gradient blocks $g = (G_W, G_b, G_D, G_c)$, the transformation induced by T_Q is

$$T_Q g \;=\; \big(G_W Q^\top,\; G_b,\; Q\,G_D,\; Q\,G_c\big), \tag{8}$$

which has the same form as the parameter transformation and preserves each block's norm. The propositions below assume exact arithmetic. Section 5 examines behavior at finite precision; Section A gives the proofs.

Lemma 1 (Gradient equivariance)**.** *Let $\ell_{\mathcal{B}}(\theta)$ be the reconstruction loss* (4) *on a minibatch $\mathcal{B}$. Then $\ell_{Q\mathcal{B}}(T_Q\theta) = \ell_{\mathcal{B}}(\theta)$ identically in θ, and wherever $\ell_{\mathcal{B}}$ is differentiable,*

$$\nabla \ell_{Q\mathcal{B}}(T_Q\theta) \;=\; T_Q\, \nabla \ell_{\mathcal{B}}(\theta). \tag{9}$$

At ReLU boundaries and TopK ties, the implemented gradients still correspond if both runs use the same differentiation and tie-breaking convention, since their pre-activations are identical (Proposition 1(i)). Decoder renormalization also commutes with T_Q. Equivariance of the full procedure therefore depends on whether the optimizer update and its state preserve the correspondence.

Proposition 2 (SGD equivariance)**.** *In exact arithmetic, SGD and momentum SGD (with any shared scalar learning-rate schedule, decoder renormalization after each step, and the standard zero initialization of the momentum buffer) satisfy Definition 1 for every orthogonal Q.*

The proof follows by induction: the update is a *linear* function of the gradient history with scalar coefficients shared by the two runs. Warmup, decay, and averaging with shared scalar coefficients preserve this property. Zero state initialization ensures that the states correspond at the start.

Coordinatewise adaptivity breaks the symmetry. Adam (Kingma & Ba, 2015) maintains, per coordinate of each block,

$$m_{t+1} = \beta_1 m_t + (1-\beta_1)\, g_t, \qquad v_{t+1} = \beta_2 v_t + (1-\beta_2)\, g_t \odot g_t, \tag{10}$$

and updates $p_{t+1} = p_t - \eta_t\, \hat{m}_{t+1} \oslash \big(\sqrt{\hat{v}_{t+1}} + \epsilon\big)$, with $\hat{m}, \hat{v}$ the bias-corrected estimates (scalar multiples of m and v). Write R for the orthogonal map T_Q induces on a vectorized block. For corresponding gradient histories, linearity gives $m'_t = R m_t$. But v stores only the *diagonal* of the gradient second-moment matrix. Rotating this matrix mixes its diagonal and off-diagonal entries, so the diagonal in the new basis depends on information that v has discarded. As a result, Adam's updates can fail to correspond even when the parameters correspond and both runs start from zero state.

Proposition 3 (Adam is not orthogonally equivariant)**.** *There exist an orthogonal Q, an initialization, and data such that Adam violates Definition 1 at the first step. In particular, with zero state initialization the first Adam step is*

$$\Delta p \;=\; -\eta\; g \oslash \big(|g| + \epsilon\big), \tag{11}$$

an ϵ-smoothed sign step, and for a 45° rotation R acting in a plane where $g = (\gamma, 0)$ with $\gamma \gg \epsilon$, the transformed step $\Delta p' \approx -\eta\,(1,1)$ deviates from the rotated step $R\,\Delta p \approx -\eta\,(1/\sqrt{2}, 1/\sqrt{2})$ by $\sqrt{2} - 1 \approx 0.41$ in relative ℓ_2.

A rotation can spread a gradient that points along one axis across several coordinates. Sign descent takes a full step in each, making the update longer. Rotating the original update would preserve its length. This geometry of sign descent on the coordinate-aligned ℓ_∞ ball (Balles & Hennig, 2018; Balles et al., 2020; Kunstner et al., 2023) explains the $O(1)$ first-step violation in Section 5. RMSProp (Tieleman & Hinton, 2012) and Adagrad (Duchi et al., 2011) use the same coordinatewise second-moment structure and break equivariance for the same reason. All three methods still respect signed permutations:

Proposition 4 (Signed-permutation equivariance of coordinatewise methods)**.** *Let $S = \mathrm{diag}(\sigma)P$ be a signed permutation, with P a permutation matrix and $\sigma \in \{\pm 1\}^d$. In exact arithmetic, Adam, RMSProp, and Adagrad, with standard zero state initialization, satisfy Definition 1 for $Q = S$.*

The signs cancel when squared: $(Sg) \odot (Sg) = P(g \odot g)$. In the float64 conditions we test, signed-permutation decoder mismatch is zero at the measured resolution, while dense-rotation mismatch persists (Section 5).

Repair by group-level second moments. We can make the update equivariant by replacing the coordinatewise second moment with one scalar for each parameter group.

Proposition 5 (Group-RMS Adam is orthogonally equivariant)**.** *Partition the parameters into groups, each a union of whole parameter blocks (e.g., one group per tensor among W, b, D, c). Replace v in Equation* (10) *by one scalar per group $\mathcal{G}$,*

$$\nu_{t+1}^{\mathcal{G}} \;=\; \beta_2\, \nu_t^{\mathcal{G}} \;+\; (1-\beta_2)\, \tfrac{1}{|\mathcal{G}|}\, \big\| g_t^{\mathcal{G}} \big\|_2^2, \tag{12}$$

keeping the first moment and the update otherwise unchanged, $\Delta p^{\mathcal{G}} = -\eta\, \hat{m}^{\mathcal{G}} / \big(\sqrt{\hat{\nu}^{\mathcal{G}}} + \epsilon\big)$. With zero initial moments and a shared scalar learning-rate schedule, this procedure satisfies Definition 1 for every orthogonal Q in exact arithmetic.

The norm $\|g^{\mathcal{G}}\|_2$ is T_Q-invariant, so the raw and transformed runs have the same scalar second-moment sequence. Earlier work used group-level moments to reduce memory use or improve stability: layer-wise in NovoGrad (Ginsburg et al., 2019), block-wise in Adam-mini (Zhang et al., 2024), and as a single scalar in AdaGrad-Norm (Ward et al., 2019). Any T_Q-invariant grouping satisfies Proposition 5. Our default is *scalar-RMS Adam*, which shares one scalar across all parameters. We also test *tensor-RMS Adam*, with one scalar per parameter tensor, in the learning-rate sweep and schedule control of Section 5.

5 Results

Feature differences at similar reconstruction quality. After 20 epochs with a dense rotation, Adam's decoder mismatch is $E_D = 0.23$–0.44 across all five layers (Table 1). Yet the layer-mean raw and rotated FVEs differ by at most 0.001. Changing the coordinates thus changes the learned decoder directions despite similar reconstruction quality on the paired data. For SGD, momentum SGD, and scalar-RMS Adam, mismatch is at or below 0.016. This is not simply because the parameters stay near their initial values: on layers 3–11, mean atom cosine to initialization is 0.20–0.26 for momentum SGD and 0.19–0.22 for Adam, even though momentum's mismatch is 20–75$\times$ lower.

Disagreement after latent matching. A fixed-index comparison can count the same direction at a different latent index as a disagreement. We therefore use optimal one-to-one matching to measure what remains after relabeling (Figure 2 and Table 15). At epoch 20, 9–16% of Adam's latents are assigned to a different index, but 90–94% of the fixed-index mismatch remains after matching on every layer. At layer 6, $E_D^{\mathrm{match}} = 0.237 \pm 0.010$, compared with $E_D = 0.253 \pm 0.013$: most of the disagreement lies in the dictionary's directions.

The difference also appears in the SAE's input-output behavior: reconstructions disagree by $E_{\mathrm{rec}} = 0.37$ in relative ℓ_2, and matched per-latent activations have mean correlation 0.74 at layer 6. Across the grid, equivariant optimizers have $E_{\mathrm{rec}} \le 0.14$ and activation correlation ≥ 0.96. In the synthetic control, both Adam twins recover the ground truth comparably while their learned dictionaries disagree (Section J).

PCA and rotation density. We also test a basis obtained from the data. The orthogonal PCA eigenbasis of the layer-6 training activations gives $E_D = 0.306$, compared with 0.249 for a random dense rotation, after 20 epochs. The transformed twins have FVEs 0.700 and 0.707, respectively, against raw-twin FVE 0.707. The PCA condition rotates the coordinates without whitening's rescaling.

Under block-orthogonal rotations, Adam's endpoint mismatch increases monotonically with participation ratio (Spearman $r_s = 1.0$), from zero at $w = 1$ to 0.249 at $w = 768$ (Figure 6 and Section G). Mixing about two coordinates per output already gives $E_D = 0.155$. Every block width uses the same raw training trajectory, with FVE 0.707.

Table 1: Decoder mismatch E_D under a dense rotation after 20 epochs (mean $\pm$ s.d. over 5 seeds, independent rotations), and the raw/rotated reconstruction quality of the Adam pair. †Embedding stream: SGD and momentum use a $10\times$ smaller learning rate (Section C).

Layer	SGD	Momentum	Adam	Scalar-RMS	FVE raw/rot.
$\ell = 0^{\dagger}$	$(1.0 \pm 0.8)\times10^{-8}$	$(2.5 \pm 0.8)\times10^{-5}$	0.436 ± 0.019	$(2.3 \pm 1.1)\times10^{-3}$	0.843/0.842
$\ell = 3$	$(7.6 \pm 4.7)\times10^{-5}$	$(1.6 \pm 1.0)\times10^{-2}$	0.315 ± 0.008	$(2.6 \pm 1.8)\times10^{-3}$	0.792/0.793
$\ell = 6$	$(8.7 \pm 3.5)\times10^{-6}$	$(5.5 \pm 1.4)\times10^{-3}$	0.253 ± 0.013	$(1.7 \pm 0.6)\times10^{-3}$	0.707/0.707
$\ell = 9$	$(6.2 \pm 1.8)\times10^{-6}$	$(4.4 \pm 3.1)\times10^{-3}$	0.232 ± 0.011	$(1.3 \pm 0.5)\times10^{-3}$	0.683/0.683
$\ell = 11$	$(5.2 \pm 1.5)\times10^{-6}$	$(3.6 \pm 2.0)\times10^{-3}$	0.268 ± 0.015	$(1.3 \pm 0.3)\times10^{-3}$	0.698/0.698

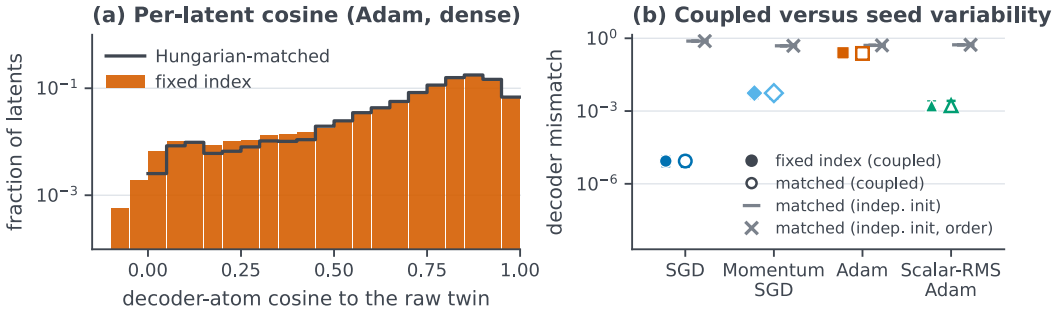


Figure 2: Decoder disagreement after latent matching (layer 6, dense rotation, 20 epochs). (a) Per-latent decoder cosine between the raw and rotated twins under Adam, pooled over 5 seeds (5,120 latents), at fixed index (bars) and after Hungarian matching (line). (b) Fixed-index (filled) and matched (hollow) mismatch of the coupled test for each optimizer, next to the matched mismatch of the independent-initialization rails (gray; 3 seeds); error bars span the observed seed range.

Mechanism and numerical controls. The mismatch is already visible in the first update (Figure 3 and Section D). Under a dense rotation, gradient deviation is 9.4×10^{-7} in float32 and 1.9×10^{-15} in float64, but Adam's update deviation is 0.882 ± 0.0004 at both precisions. The first moment agrees to the same numerical precision as the gradient. The diagonal second moment has deviation 1.4 under the squared-transform metric of Section D. Coordinatewise variants show the same first-step behavior, whereas scalar-RMS Adam tracks SGD.

Repeating the runs at higher precision separates this effect from numerical drift (Table 7 and Section F). Adam's signed-permutation decoder mismatch falls from 7.5×10^{-3} to zero in float64, and SGD's dense-rotation mismatch falls from about 10^{-5} to zero. Adam's dense-rotation mismatch instead remains 0.249 at both precisions. Setting $k = m$ removes the TopK selection discontinuity and reduces this mismatch to 0.031. TopK selection thus amplifies the optimizer's underlying coordinate dependence.

Repair at comparable reconstruction quality. At the first crossing of FVE $= 0.65$, Adam has $E_D = 0.11$, compared with 7.1×10^{-5}–1.1×10^{-3} for the equivariant methods (Figure 8 and Table 12). Momentum SGD reaches FVE $= 0.704$ at epoch 20, close to Adam's 0.707, with $45\times$ lower mean mismatch.

For the layer-6 learning-rate sweep, we give Adam and both group-RMS variants the same grid and 20-epoch budget (Figure 9 and Table 13). The best observed Adam setting reaches FVE $= 0.7321 \pm 0.0003$ and $E_D = 0.50$ at the largest tested rate. Tensor-RMS Adam reaches FVE $= 0.716 \pm 0.001$, a gap of 0.016, with $E_D = 1.7 \times 10^{-2}$; its corresponding float64 run has zero measured decoder mismatch.

Longer training and learning-rate schedules. At layer 6, the epoch-20 default-rate matched mismatch 0.237 is 0.46 of the independent-initialization reference 0.510 (Tables 5 and 17). A separate 100-epoch control at peak rate 3×10^{-3} compares constant and cosine schedules (Table 6). Adam's matched mismatch is still 0.233 and 0.269, respectively. With cosine decay, tensor-RMS has mismatch 0.044 and raw FVE 0.740, versus Adam's 0.743 (paired FVE gap 0.00340 ± 0.00045).

Tensor-RMS therefore achieves similar reconstruction quality with lower dictionary mismatch in this longer, annealed run. By epoch 100, latent matching removes a larger share of Adam's fixed-index difference (Section E).

6 Independent Training as a Reference

We also train independent pairs to compare the coupled mismatch with ordinary run-to-run variation. Because these pairs start with different dictionaries and latent labels, we use Hungarian matching to compare them. In the separate configuration of Sections C and L, pairs trained independently in the same basis have mean matched cosine 0.370, compared with 0.368–0.370 for independently initialized raw-versus-rotated runs (mismatch ratios 1.000–1.003). Adding a rotation changes this mean little relative to the variation between seeds (Paulo & Belrose, 2025; Gerasimov et al., 2026). At the layer-6 test configuration, independent initialization gives $E_D^{\text{match}} \approx 0.5$ for Adam, momentum SGD, and scalar-RMS Adam after 20 epochs (Table 17).

The coupled and independent comparisons answer different questions. With coupling, the initial correspondence is fixed, and an equivariant procedure has zero mismatch in exact arithmetic. Independent runs measure agreement between different starts. In the reconstruction stitching comparisons of Section L, linear combinations also account for more of the reconstruction discrepancy between independent runs than permutations do. These comparisons can be used alongside training methods designed to stabilize features across independent runs (Wang et al., 2025; Brzozowski & Chung, 2026).

7 Related Work

Evaluating interpretability procedures. SAEs extract candidate features from language-model activations (Bricken et al., 2023; Cunningham et al., 2024; Gao et al., 2024). Several evaluations go beyond reconstruction. SAEBench measures practical utility alongside proxy metrics (Karvonen et al., 2025), while CHIVE evaluates explanations by their value for predicting counterfactual behavior (Karvonen et al., 2026). In cross-architecture model diffing, differences in the representations must also be distinguished from differences introduced by the analysis procedure (Jiralerspong & Bricken, 2026). We test the procedure by changing the input basis while preserving the information and objective. This follows the logic of sanity checks for saliency maps (Adebayo et al., 2018): apply a controlled change and check whether the explanation procedure behaves as expected. Coupling the initialization and minibatches gives an exact starting correspondence that we can test throughout training.

Instability and identifiability. SAE dictionaries vary across training runs: Paulo & Belrose (2025) find substantial disagreement across seeds, and Gerasimov et al. (2026) show that individually unstable features can occupy reproducible subspaces. Song et al. (2026) argue that feature consistency should be evaluated routinely and relate dictionary agreement to agreement between feature explanations. Other work modifies training to improve consistency (Wang et al., 2025; Brzozowski & Chung, 2026). In our paired protocol, the initial dictionaries correspond exactly. We then follow whether the optimizer preserves that correspondence after a coordinate change. Independent-seed comparisons put the mismatch in context, and optimal matching removes differences due to latent labels. Network-alignment methods (Entezari et al., 2022; Ainsworth et al., 2023) similarly account for permutation symmetries. Our input-basis transformation specifies a correspondence from the start, so it can be checked throughout training.

Optimizer geometry. Adam's sign-like, coordinate-dependent geometry is well studied (Balles & Hennig, 2018; Balles et al., 2020; Kunstner et al., 2023), and adaptive updates can select different solutions (Wilson et al., 2017). Zhang et al. (2025) study rotation-dependent optimization performance. We examine how this geometry affects feature correspondence at matched reconstruction quality. The first failure occurs in Adam's diagonal second moment. Group-level adaptive methods already exist (Ginsburg et al., 2019; Zhang et al., 2024; Ward et al., 2019); we give a symmetry criterion for choosing groups that preserve correspondence and use these updates as controls. Full-matrix preconditioning (Duchi et al., 2011) and spectral matrix updates such as Shampoo and Muon (Gupta

et al., 2018; Jordan et al., 2024) are candidates for further equivariant baselines. Their bias updates and all other training operations would also need to respect the transformation.

8 Discussion and Limitations

A control for feature discovery. Similar reconstruction scores do not imply similar decoder directions. In the coupled test, we know how the dictionaries should correspond at initialization and can measure whether that relationship holds after training. It can fail even when the coordinate change comes from PCA, a standard preprocessing choice. Comparisons between SAE dictionaries should therefore report the optimizer, input basis, training budget, and seed protocol as well as reconstruction quality.

Mechanism and repair. Adam's diagonal second moment gives each coordinate its own adaptive scale. We instead compute one second-moment scale for each parameter group preserved by the parameter transport. With one scalar per tensor, the update remains adaptive and approaches Adam's reconstruction quality in our layer-6 experiments. After 100 epochs with cosine decay, its matched mismatch is 0.044 versus Adam's 0.269, with raw FVE lower by 0.0034. In this setting, much of the dictionary disagreement can be removed at a small cost in reconstruction quality.

Limitations. We tested compact TopK SAEs on GPT-2 Small using $55{,}000$ activation rows. Training and evaluation positions share contexts, and the learning-rate sweep has no separate test set. To establish whether the quality comparisons generalize, the experiments need to be repeated at larger scale with disjoint training and evaluation documents. The 100-epoch control covers one longer, annealed recipe; it does not establish convergence or robustness across schedules. TopK can amplify float32 rounding, and we ran float64 controls only for selected conditions. We have not tested SAE architectures beyond TopK.

Our measurements concern geometry and reconstruction. They do not test whether matched features have the same meaning or causal role. In the synthetic experiment, the two dictionaries recover the ground truth similarly despite disagreeing with each other, so disagreement alone cannot tell us which dictionary is better. A next step is to ablate matched features and compare their effects on prespecified model decisions, using independent text. This would test whether the geometric differences also change causal attributions.

Conclusion. Changing the coordinates of fixed activations can change the dictionary learned by an SAE, with little change in reconstruction quality. The coupled test traces this effect to Adam's coordinatewise second moment. Group-RMS updates preserve the correspondence in exact arithmetic and reduce the measured mismatch at comparable reconstruction quality.

AI use statement

AI coding assistants (OpenAI Codex and Claude Code) were used to help implement and test the experimental code. Generative AI tools were also used to polish the manuscript for clarity and readability. The research ideas, methodology, analysis, and conclusions were developed by the author. The author reviewed all AI-assisted work and takes responsibility for the final content.

Reproducibility Statement

Proofs and experimental settings appear in Sections A and C. The anonymous code supplement includes implementations, pinned dependencies, per-seed results, tests, and figure-generation scripts.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/2018/hash/294a8ed24b1ad22ec2e7efea049b8737-Abstract.html.

Samuel K. Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git Re-Basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations (ICLR)*, 2023.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning (ICML)*, 2018.

Lukas Balles, Fabian Pedregosa, and Nicolas Le Roux. The geometry of sign gradient descent. *arXiv preprint arXiv:2002.08056*, 2020.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features.

Michał Brzozowski and Neo Christopher Chung. Aligned training: A parameter-free method to improve feature quality and stability of sparse autoencoders. *arXiv preprint arXiv:2605.18629*, 2026.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Rahim Entezari, Hanie Sedghi, Olga Saukh, and Behnam Neyshabur. The role of permutation invariance in linear mode connectivity of neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*, 2024.

Gleb Gerasimov, Timofei Rusalev, Nikita Balagansky, Daniil Laptev, Vadim Kurochkin, and Daniil Gavrilov. Unstable features, reproducible subspaces: Understanding seed dependence in sparse autoencoders. *arXiv preprint arXiv:2606.12138*, 2026.

Boris Ginsburg, Patrice Castonguay, Oleksii Hrinchuk, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Huyen Nguyen, Yang Zhang, and Jonathan M. Cohen. Stochastic gradient methods with layer-wise adaptive moments for training of deep networks. *arXiv preprint arXiv:1905.11286*, 2019.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning (ICML)*, 2018.

Thomas Jiralerspong and Trenton Bricken. Cross-architecture model diffing with crosscoders: Unsupervised discovery of differences between LLMs. *arXiv preprint arXiv:2602.11729*, 2026. URL https://arxiv.org/abs/2602.11729.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. Blog post, 2024. URL https://kellerjordan.github.io/posts/muon/.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264. PMLR, 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Adam Karvonen, Euan Ong, Subhash Kantamneni, and Samuel Marks. Would this change your answer? evaluating explanations of LLM behavior in the wild with counterfactual experiments. *arXiv preprint arXiv:2608.16747*, 2026. URL https://arxiv.org/abs/2608.16747.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between SGD and Adam on transformers, but sign descent might be. *arXiv preprint arXiv:2304.13960*, 2023.

Alireza Makhzani and Brendan Frey. k-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features. *arXiv preprint arXiv:2501.16615*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Technical Report, 2019.

Xiangchen Song, Aashiq Muhamed, Yujia Zheng, Lingjing Kong, Zeyu Tang, Mona T. Diab, Virginia Smith, and Kun Zhang. Mechanistic interpretability should prioritize feature consistency in sparse autoencoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2172–2210. Association for Computational Linguistics, July 2026. doi: 10.18653/v1/2026.acl-long.99. URL https://aclanthology.org/2026.acl-long.99/.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L. Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. Transformer Circuits Thread, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5: RMSProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Sophie L. Wang, Alex Quach, Nithin Parsan, and John J. Yang. Enforcing orderedness to improve feature consistency. *arXiv preprint arXiv:2512.02194*, 2025.

Rachel Ward, Xiaoxia Wu, and Léon Bottou. AdaGrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning (ICML)*, 2019.

Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Tianyue H. Zhang, Lucas Maes, Alan Milligan, Alexia Jolicoeur-Martineau, Ioannis Mitliagkas, Damien Scieur, Simon Lacoste-Julien, and Charles Guille-Escuret. Understanding Adam requires better rotation dependent assumptions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P. Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024.

A Deferred Proofs

Proof of Proposition 1. (i) The pre-activation vectors are identical: $WQ^\top(Qa) + b = Wa + b$. Thus ReLU and TopK_k receive the same input and return the same code under any tie-breaking rule. (ii) Using (i), $f_{T_Q\theta}(Qa) = QD\, z_\theta(a) + Qc = Q\big(Dz_\theta(a) + c\big)$. (iii) By (ii) and orthogonality, $\|f_{T_Q\theta}(Qa) - Qa\|_2 = \|Q(f_\theta(a) - a)\|_2 = \|f_\theta(a) - a\|_2$; take expectations. (iv) Q preserves column norms, $\|Qd_j\|_2 = \|d_j\|_2$, and ${T_Q}^\top T_Q = \mathrm{id}$. □

Proof of Lemma 1. The identity of losses is Proposition 1(iii) applied to the empirical distribution of $\mathcal{B}$. Since T_Q is an orthogonal linear map on parameter space, the chain rule applied to $\ell_{Q\mathcal{B}} \circ T_Q = \ell_{\mathcal{B}}$ gives $T_Q^\top \nabla \ell_{Q\mathcal{B}}(T_Q\theta) = \nabla \ell_{\mathcal{B}}(\theta)$, and ${T_Q^{-}}^\top = T_Q$. □

Proof of Proposition 2. Momentum SGD iterates $u_{t+1} = \mu u_t + \nabla \ell_{\mathcal{B}_t}(\theta_t)$ and $\theta_{t+1} = \mathrm{Norm}\big(\theta_t - \eta_t u_{t+1}\big)$, where Norm denotes decoder renormalization (SGD is $\mu = 0$). Suppose $\theta'_t = T_Q\theta_t$ and $u'_t = T_Q u_t$; both hold at $t = 0$ since the transformed run starts at $T_Q\theta_0$ and $T_Q 0 = 0$. By Lemma 1 and linearity, $u'_{t+1} = \mu T_Q u_t + T_Q \nabla \ell_{\mathcal{B}_t}(\theta_t) = T_Q u_{t+1}$; by linearity of the parameter update and commutation of Norm with T_Q, $\theta'_{t+1} = T_Q\theta_{t+1}$. Induct over t. □

Proof of Proposition 3. At $t = 1$, $\hat{m}_1 = g$ and $\hat{v}_1 = g \odot g$, giving Equation (11). For $g = (\gamma, 0)$ the step is $\Delta p \approx -\eta\,(1, 0)$, while $Rg = \big(\gamma/\sqrt{2}, \gamma/\sqrt{2}\big)$ gives $\Delta p' \approx -\eta\,(1, 1)$: a sign step has unit magnitude in every coordinate carrying a nonzero gradient, regardless of how the rotation distributed the gradient's energy. The stated deviation follows from $\|(1,1) - (1/\sqrt{2}, 1/\sqrt{2})\|_2 = \sqrt{2} - 1$ and $\|R\Delta p\|_2 = 1$ (in units of η). Such gradient configurations are realized by SAE instances (for example through the decoder-bias block, whose gradient can point in any direction of $\mathbb{R}^d$), so the procedure-level definition fails. □

Proof of Proposition 4. T_S acts on each parameter block as a signed permutation of its entries. For any signed permutation, $(Sg) \odot (Sg) = |S|(g \odot g)$ with $|S| = P$: signs cancel in the square. By induction the states transform as $m'_t = Sm_t$ and $v'_t = Pv_t$ (zero initialization corresponds), and the elementwise update respects this pair of actions: coordinate i of $Sm \oslash (\sqrt{Pv} + \epsilon)$ equals $\sigma_i\, m_{\pi(i)} / \big(\sqrt{v_{\pi(i)}} + \epsilon\big) = \big(S\,(m \oslash (\sqrt{v} + \epsilon))\big)_i$. Decoder renormalization commutes as before. Induct over t. □

Proof of Proposition 5. T_Q acts orthogonally on every block (8), so $\|g^{\mathcal{G}}\|_2$ is invariant for any group that is a union of blocks. Hence both runs use identical scalar sequences $\nu_t^{\mathcal{G}}$. Each group's update is a scalar multiple of $\hat{m}^{\mathcal{G}}$, which transforms linearly by Lemma 1. Renormalization also commutes with the transformation, so induction gives Equation (5). □

B Oracle Verification of the Transformed-SAE Construction

Proposition 1 holds in exact arithmetic. To check the construction in float32, we use an SAE trained on GPT-2 Small layer-6 residual-stream activations ($m = 2048$, $k = 32$; training details in Section C). We construct the oracle $T_Q\hat{\theta}$ for both a signed permutation and a dense random orthogonal matrix. On $5{,}000$ held-out tokens, we then compare its codes and reconstructions with the corresponding transformed outputs of the raw model. All calculations use float32 (Table 2).

Table 2: Oracle transformed-SAE check (float32). Relative ℓ_2 errors are computed over all held-out codes and reconstructions; exact-support fraction is the share of evaluation tokens whose TopK active set matches exactly.

Transform	Code rel. ℓ_2	Recon. rel. ℓ_2	Exact support	Min. atom cosine
Signed permutation	5.1×10^{-7}	4.3×10^{-7}	1.0000	> 0.9999995
Dense orthogonal	4.4×10^{-4}	3.7×10^{-4}	0.9998	> 0.9999995

Under the dense rotation, TopK support changes for one of the $5{,}000$ evaluation tokens (exact-support fraction $0.9998 = 4999/5000$). That token accounts for most of the aggregate code error: a rounding-level pre-activation difference crosses the selection boundary. The maximum absolute code deviation is 0.45. Decoder parameters map exactly: the maximum absolute decoder-bias error is 0, and encoder-row and decoder-atom cosines to the oracle are 1 within 5×10^{-7}.

C Experimental Details

Data. We collect activations from GPT-2 Small ($d = 768$) at the inputs to blocks $\ell \in \{0, 3, 6, 9, 11\}$, using 128-token contexts packed from the Pile-10k sample (Gao et al., 2020). We drop position 0 of each context. All layers use the same $55{,}000$ token rows, shuffled once and split into $50{,}000$ training rows and $5{,}000$ evaluation rows. For each layer, we subtract the training-mean vector and divide by a single global training-RMS scalar (Section 3.1).

Architecture and initialization. We use the TopK SAE of Section 2.1. The default settings are $m = 1024$, $k = 16$ (robustness cells: $k \in \{8, 64\}$; $m = 2048$, $k = 32$). We sample decoder atoms from $\mathcal{N}(0, 1/d)$ and normalize each column. The encoder starts at the decoder transpose plus Gaussian noise with standard deviation 10^{-2}, and biases start at zero. We renormalize decoder columns after every optimizer step. The training loss is the per-coordinate mean squared error (MSE) multiplied by d. This rescaling multiplies all gradients by a constant, so it preserves equivariance and only shifts the effective SGD learning rate.

Optimization. The default SGD learning rate is 10^{-2} (the embedding stream $\ell = 0$ and the $k = m$ precision arms use 10^{-3} for stability; divergence details in Section F), with momentum 0.9 for momentum SGD. All adaptive optimizers use learning rate 10^{-3} (the rescue sweep of Section 5, *Repair at comparable reconstruction quality* uses $\{3{\times}10^{-4}, 10^{-3}, 3{\times}10^{-3}\}$ for Adam and both group-RMS variants). The remaining settings are $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, RMSProp $\alpha = 0.99$, and no weight decay. Unless stated otherwise, we use batch size 2048 and train for 20 epochs (≈ 500 steps). The longer SGD arms run for 60 and 100 epochs; the one-step diagnostics use batch size 512 and run for 5 steps. The main coupled grid uses constant learning rates and is evaluated after every epoch. The schedule control of Section E trains Adam and tensor-RMS Adam for 100 epochs, at peak rate 3×10^{-3}, with constant or cosine-decayed rates and evaluations at epochs 0, 1, and every 5 epochs.

Transforms, seeds, and precision. For each random-transform seed, we draw a separate signed permutation and dense rotation using generator seeds $s + 10^4$ and $s + 2 \times 10^4$, respectively. We use the same transform across optimizers and, since all layers have $d = 768$, across layers. For each seed and block width, we draw a separate block-orthogonal transform. To sample dense and block rotations from the Haar distribution, we use QR of a Gaussian matrix with a sign-fixed R diagonal. The matrices are computed and stored in float64 (orthogonality error $\sim 10^{-15}$), then cast to the training precision when applied. The core grid, one-step diagnostics, and optimizer ablations use seeds 0–4; the remaining multi-seed arms use seeds 0–2, as specified in captions. For the float64 arms, we rerun the same configuration on CPU from the same float32-representable initialization. Float32 models train on an Apple-silicon GPU (MPS; float64 on CPU). A coupled pair at the default configuration takes 10–60 seconds to train; all experiments together take a few hours on one laptop.

Independent-run configuration (Section 6). These runs use $m = 2048$, $k = 32$, and AdamW (learning rate 10^{-3}, cosine decay to 10^{-4}), with gradient clipping at norm 1 and 40 epochs of training. The data and preprocessing are the same. Stitching maps are fit on $10{,}000$ tokens with ridge regularization 10^{-2}.

D Optimizer Behavior and One-Step Diagnostics

Table 3 summarizes the exact-arithmetic status of each optimizer family. To measure second-moment error, consider a vectorized parameter block with orthogonal action R. We compare the transformed run's stored diagonal second moment v' with $(R \odot R)v = \mathrm{diag}(R\,\mathrm{Diag}(v)R^\top)$. The reported error

Table 3: Exact-arithmetic equivariance of the optimizer families in the coupled test (Definition 1). Nonzero float32 discrepancies are reported in Section 5.

Optimizer family	Signed permutation	Dense orthogonal
SGD, momentum SGD	preserves (Proposition 2)	preserves (Proposition 2)
Adam, RMSProp, Adagrad	preserves (Proposition 4)	breaks (Proposition 3)
Group-RMS Adam	preserves (Proposition 5)	preserves (Proposition 5)

is $\|v' - (R \odot R)v\|_2 / \|(R \odot R)v\|_2$, concatenated over parameter blocks. For a dense Q, this is equivalent to comparing against $\mathrm{diag}(Q\,\mathrm{Diag}(v)Q^\top)$ along each affected input or output axis.

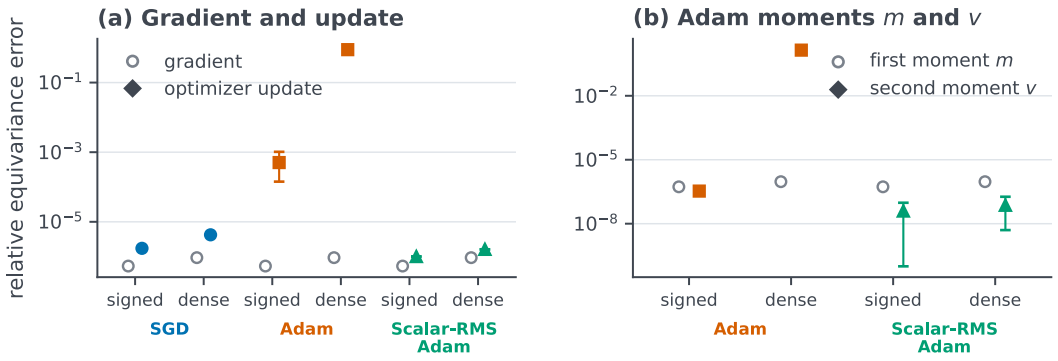


Figure 3: First-step correspondence errors (layer 6, float32; points are means over 5 seeds and error bars span the observed range; Section 5, *Mechanism and numerical controls*). (a) Gradients correspond at the rounding floor for every optimizer, while Adam's update under a dense rotation deviates by $O(1)$. (b) Adam's first moment m corresponds; only the diagonal second moment v breaks, and only under the dense rotation.

Table 4 gives the relative ℓ_2 deviation of the first parameter update for both transforms and precisions (layer 6, batch 512, mean over 5 seeds with independent transforms). The coordinatewise Adam variants give the same first-step deviation because neither β parameter appears in Equation (11). RMSProp differs only in omitting bias correction. With a dense rotation, the deviation stays at 0.882 in float64. Signed-permutation deviations, in contrast, fall by eight to nine orders of magnitude. Group-RMS Adam and SGD reach the same numerical floor.

Table 4: First-step update deviation (relative ℓ_2; mean over 5 seeds). The coordinatewise rows are identical because the first Adam-family step Equation (11) does not depend on β_1 or β_2.

	Signed permutation		Dense rotation	
Optimizer	float32	float64	float32	float64
SGD, momentum SGD	1.7×10^{-6}	3.1×10^{-15}	4.2×10^{-6}	7.9×10^{-15}
Adam, β_1=0, β_2=0, Adagrad	5.0×10^{-4}	1.1×10^{-12}	0.882	0.882
RMSProp	1.8×10^{-4}	4.1×10^{-13}	0.882	0.882
Scalar-RMS Adam	1.0×10^{-6}	1.8×10^{-15}	1.6×10^{-6}	3.1×10^{-15}

Sign flips as a numerical control. At $w = 1$, diagonal sign flips preserve reduction order and yield zero measured decoder mismatch in float32 (Section 5, *PCA and rotation density*).

E Training Horizon and Schedule

At layer 6, Adam's coupled matched mismatch increases over the first 20 epochs while its independent-initialization reference decreases (Table 5 and Figure 4). The ratio of those seed means is 0.11, 0.24, and 0.46 at epochs 5, 10, and 20.

Table 5: Adam dense-rotation mismatch over training (constant learning rate 10^{-3}). Means over 5 GPT-2 coupled seeds and 3 synthetic seeds; Init is matched mismatch for 3 independently initialized pairs at the same GPT-2 epoch. Steps are updates per twin, $T = \text{epochs}\,\lceil N/B \rceil$.

Data	Epochs	Steps	E_D	E_D^{match}	Relabeled (%)	Init
GPT-2 L6	5	125	0.073	0.073	0.0	0.662
GPT-2 L6	10	250	0.139	0.138	1.1	0.587
GPT-2 L6	20	500	0.253	0.237	8.9	0.510
Synthetic	20	640	0.041	0.041	0.0	—
Synthetic	100	3200	0.218	0.083	29.7	—

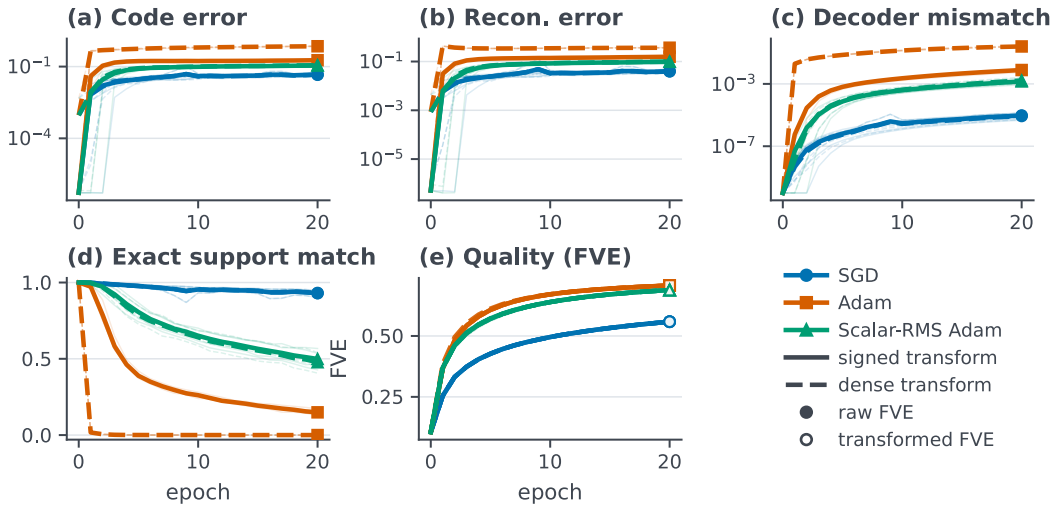


Figure 4: Coupled training trajectories on GPT-2 Small layer-6 activations ($m = 1024$, $k = 16$; thick curves are means over five seeds, faint curves are individual runs with independent transforms). Dense-rotation mismatch grows under Adam while reconstruction quality improves; Table 5 reports selected values. Exact zeros are drawn at 10^{-10} in logarithmic panels.

Longer-horizon schedule control. For layer-6 Adam and tensor-RMS, we extend training at the previously tested peak learning rate 3×10^{-3} to 100 epochs. For each optimizer and seed $s \in \{0, 1, 2\}$, constant and cosine schedules share the activation data, initial dictionaries, dense rotation, and minibatch index sequence. This gives 12 coupled pairs. All runs use $m = 1024$, $k = 16$, batch size 2048, and float32/MPS, as in the original learning-rate sweep.

For the cosine arm, the rate applied to update $t + 1$, with $t = 0, \ldots, T - 1$ and $T = 2500$, is

$$\eta_t = \eta_0 \left[0.1 + 0.9\, \frac{1 + \cos(\pi t / T)}{2} \right].$$

The schedule decays to $0.1\eta_0$ over 100 epochs. Evaluations at epochs 20 and 50 are intermediate snapshots of this 100-epoch schedule. We report means and sample standard deviations over the three seeds; optimizer and schedule differences are paired within each seed.

Results. At epoch 100, Adam's matched mismatch is 0.233 ± 0.015 under a constant rate and 0.269 ± 0.009 under cosine decay (Table 6 and Figure 5). Tensor-RMS has 0.092 ± 0.018 and 0.044 ± 0.006, respectively. Under cosine decay, the paired Adam-minus-tensor-RMS FVE gaps are 0.00340 ± 0.00045 in the raw basis and 0.00314 ± 0.00093 in the rotated basis. Annealing therefore does not remove the coordinate discrepancy. The group-RMS control has substantially lower matched mismatch at similar reconstruction quality.

The share of mismatch explained by relabeling depends on the training recipe. At the cosine endpoint, Adam's fixed mismatch is 0.575 ± 0.007; matching leaves 46.8% of it (ratio of seed means), and $60.2 \pm 0.9\%$ of latents are relabeled. In the 20-epoch default-rate grid, 90–94% of the mismatch remains after matching. Over epochs 80–100, cosine Adam's matched mismatch

changes by $+0.00041 \pm 0.00007$ and raw FVE by -0.00019 ± 0.00005; tensor-RMS changes by $+0.00298 \pm 0.00040$ and $+0.00065 \pm 0.00001$.

The epoch-20 constant-rate snapshots reproduce the corresponding learning-rate-sweep results in Table 13.

Table 6: Schedule control on GPT-2 layer 6 (dense rotations; $m = 1024$, $k = 16$). Mean $\pm$ sample s.d. over three paired seeds; peak learning rate 3×10^{-3}. Both schedules run for 100 epochs; cosine decays to 0.1 of the peak rate. Epochs 20 and 50 are intermediate snapshots of the 100-epoch runs.

Schedule	Optimizer	Epoch	E_D^{match}	FVE raw	FVE rot.
Constant	Adam	20	0.300 ± 0.007	0.7321 ± 0.0003	0.7314 ± 0.0004
Constant	Adam	50	0.252 ± 0.014	0.7411 ± 0.0005	0.7402 ± 0.0009
Constant	Adam	100	0.233 ± 0.015	0.7427 ± 0.0002	0.7418 ± 0.0007
Constant	Tensor-RMS	20	0.016 ± 0.003	0.7159 ± 0.0014	0.7127 ± 0.0029
Constant	Tensor-RMS	50	0.054 ± 0.014	0.7337 ± 0.0027	0.7296 ± 0.0071
Constant	Tensor-RMS	100	0.092 ± 0.018	0.7387 ± 0.0004	0.7352 ± 0.0049
Cosine	Adam	20	0.306 ± 0.005	0.7327 ± 0.0005	0.7316 ± 0.0005
Cosine	Adam	50	0.271 ± 0.009	0.7423 ± 0.0004	0.7418 ± 0.0007
Cosine	Adam	100	0.269 ± 0.009	0.7433 ± 0.0004	0.7431 ± 0.0005
Cosine	Tensor-RMS	20	0.011 ± 0.003	0.7149 ± 0.0024	0.7154 ± 0.0008
Cosine	Tensor-RMS	50	0.031 ± 0.005	0.7352 ± 0.0004	0.7352 ± 0.0001
Cosine	Tensor-RMS	100	0.044 ± 0.006	0.7399 ± 0.0002	0.7399 ± 0.0004

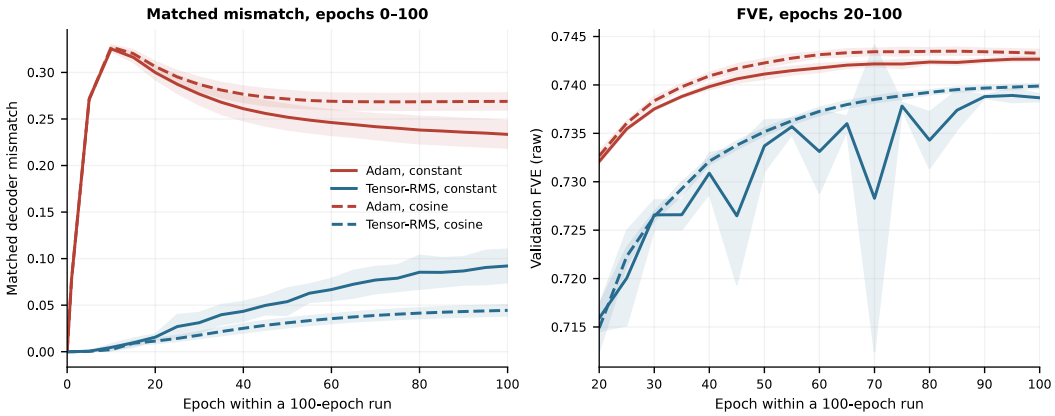


Figure 5: Longer-horizon schedule control at layer 6 (three paired seeds; peak learning rate 3×10^{-3}; float32). Lines show seed means and bands show ± 1 sample s.d. Left: matched decoder mismatch over the full 100-epoch run. Right: raw validation FVE, zoomed to epochs 20–100. The constant tensor-RMS trajectory has larger fluctuations than its cosine counterpart. Both cosine curves use the full 100-epoch decay horizon.

F Full Coupled-Grid Tables

At layer 3, which has the largest float32 deviation among the equivariant grid cells, matched float64 runs give $E_D = 0$ for SGD, momentum SGD, and scalar-RMS Adam under both transforms.

Signed-permutation companion to Table 1. Table 8 reports E_D under signed permutations (Proposition 4). These nonzero float32 values are consistent with finite-precision drift; the selected matched float64 runs give $E_D = 0$. Adam's signed-permutation mismatch is 1–2 orders below its dense-rotation mismatch on every layer.

Embedding-stream divergence. At the shared default lr $= 10^{-2}$, plain SGD diverges on 1 of 5 seeds on $\ell = 0$ and momentum SGD on 5 of 5 (the embedding stream is heavy-tailed after global-RMS normalization). At 10^{-3} both train stably (FVE $= 0.59$ and 0.75, respectively) and preserve

Table 7: Final E_D at two numerical precisions (Section 5, *Mechanism and numerical controls*; layer 6, mean $\pm$ s.d. over 3 seeds), with and without the TopK discontinuity ($k = m$; SGD uses learning rate 10^{-3} there for stability). Entries "0" denote zero measured decoder-cosine mismatch; supports match on all 5,000 evaluation tokens and code errors remain at the float64 floor.

Condition	float32, k=16	float64, k=16	float32, k=m	float64, k=m
SGD, signed perm.	$(1.0 \pm 0.5)\times10^{-5}$	0	$(2.4 \pm 0.7)\times10^{-11}$	0
SGD, dense	$(9.5 \pm 4.5)\times10^{-6}$	0	$(3.4 \pm 0.3)\times10^{-11}$	0
Adam, signed perm.	$(7.5 \pm 1.9)\times10^{-3}$	0	$(1.2 \pm 0.2)\times10^{-5}$	0
Adam, dense	0.249 ± 0.010	0.249 ± 0.009	$(3.1 \pm 0.1)\times10^{-2}$	$(3.1 \pm 0.1)\times10^{-2}$

Table 8: Signed-permutation E_D after 20 epochs (mean $\pm$ s.d., 5 seeds). †SGD-family entries on $\ell = 0$ use the 10^{-3}-learning-rate supplement, as in Table 1.

Layer	SGD	Momentum	Adam	Scalar-RMS
$\ell = 0^\dagger$	$(0.9 \pm 1.0)\times10^{-8}$	$(2.1 \pm 0.6)\times10^{-5}$	$(3.1 \pm 0.6)\times10^{-2}$	$(2.2 \pm 1.1)\times10^{-3}$
$\ell = 3$	$(7.3 \pm 3.9)\times10^{-5}$	$(1.0 \pm 0.5)\times10^{-2}$	$(1.0 \pm 0.6)\times10^{-2}$	$(1.7 \pm 0.5)\times10^{-3}$
$\ell = 6$	$(9.4 \pm 4.4)\times10^{-6}$	$(7.6 \pm 3.6)\times10^{-3}$	$(7.8 \pm 1.4)\times10^{-3}$	$(1.4 \pm 0.5)\times10^{-3}$
$\ell = 9$	$(6.0 \pm 0.6)\times10^{-6}$	$(5.3 \pm 2.3)\times10^{-3}$	$(1.1 \pm 0.6)\times10^{-2}$	$(1.0 \pm 0.6)\times10^{-3}$
$\ell = 11$	$(6.2 \pm 1.4)\times10^{-6}$	$(3.9 \pm 2.4)\times10^{-3}$	$(1.1 \pm 0.8)\times10^{-2}$	$(1.4 \pm 0.3)\times10^{-3}$

the correspondence. Plain SGD moves little from initialization (cosine 0.979), whereas momentum SGD both moves (cosine 0.80) and preserves correspondence ($E_D \leq 3.7 \times 10^{-5}$). Divergent runs are flagged and not averaged.

Optimizer ablation (layer 6, 5 seeds). Adam with $\beta_1 = 0$, RMSProp, and Adagrad have dense-to-signed mismatch ratios of 6–1200$\times$ (Table 9); magnitudes track each schedule's effective step sizes. Adagrad both undertrains (FVE $= 0.49$) and has lower drift. Adam with $\beta_2 = 0$ diverges before an endpoint can be measured: its update $\hat{m}_t \oslash (|g_t| + \epsilon)$ divides accumulated momentum by the *current* gradient magnitude plus the stabilizer. The update can become very large when the current gradient is small relative to the accumulated momentum.

Table 9: Ablation: E_D after 20 epochs and reconstruction quality (layer 6, 5 seeds).

Optimizer	Signed perm.	Dense	FVE
Adam, $\beta_1 = 0$	$(3.4 \pm 0.8)\times10^{-2}$	0.30 ± 0.01	0.708
Adam, $\beta_2 = 0$	diverges (float32)		
RMSProp	0.11 ± 0.02	0.64 ± 0.01	0.717
Adagrad	$(2.3 \pm 1.8)\times10^{-5}$	$(2.9 \pm 0.1)\times10^{-2}$	0.486

G Rotation-Density Results

Relative to the dense random rotation, PCA and DCT give 1.23$\times$ and 1.02$\times$ its mismatch, respectively (Table 11). The endpoint and one-step experiments (Figure 7) share the same per-(seed, width) transform draws.

H Matched-Quality and Batch-Order Comparisons

With independently scrambled minibatch orders (3 seeds, 20 epochs), SGD has final $E_D = 2.0\times10^{-3}$ under either transform, compared with 0.31 (signed) and 0.37 (dense) for Adam. For SGD, the batch-order perturbation is about 200$\times$ larger than a rotation but still 130$\times$ smaller than Adam's rotation-induced mismatch.

Table 10: Robustness cells under a dense rotation (layer 6, 20 epochs, mean ± s.d. over 3 seeds). Mismatch columns report E_D.

Configuration	Adam	SGD	Scalar-RMS	Adam FVE
m=1024, k=8	0.30 ± 0.00	$(3.3 \pm 1.4)\times10^{-6}$	$(2.6 \pm 3.0)\times10^{-3}$	0.672
m=1024, k=64	0.15 ± 0.00	$(2.3 \pm 0.6)\times10^{-5}$	$(1.3 \pm 0.1)\times10^{-3}$	0.768
m=2048, k=32	0.20 ± 0.00	$(1.0 \pm 0.2)\times10^{-5}$	$(4.0 \pm 0.7)\times10^{-3}$	0.771

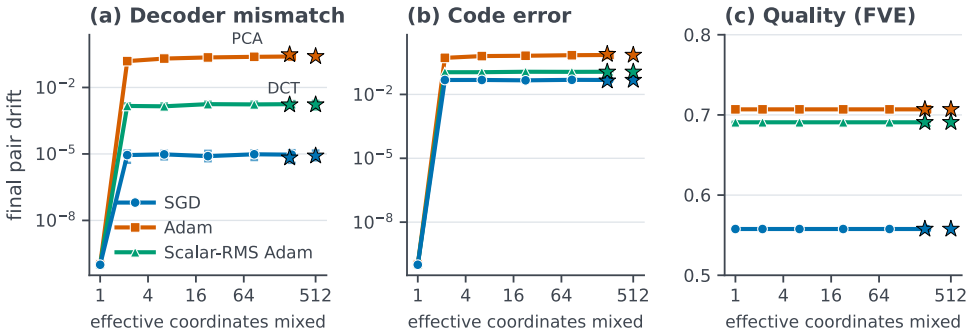


Figure 6: Endpoint results versus rotation density (Section 5, *PCA and rotation density*; layer 6, 20 epochs; means and observed ranges over 3 seeds; zeros in log panels displayed at 10^{-10}). Stars mark the deterministic PCA eigenbasis and DCT references. (a,b) Adam's mismatch rises with the number of coordinates mixed, while SGD and scalar-RMS Adam remain near their finite-precision levels. (c) Raw-side reconstruction quality is flat across densities.

Table 11: Endpoint mismatch by rotation density (layer 6, 20 epochs, mean ± s.d. over 3 seeds): final E_D by transform, ordered by measured mixing density (mean participation ratio, Section 3.1). Adam's raw-side FVE is 0.7070 in every row; transformed-side FVE is 0.6996 for PCA and 0.7070 for the dense random rotation. At $w = 1$, all optimizers have zero measured decoder mismatch (Section D).

Transform	Mixing	SGD	Adam	Scalar-RMS
block w=1	1.0	0	0	0
block w=4	2.2	$(8.9 \pm 5.1)\times10^{-6}$	0.15 ± 0.01	$(1.5 \pm 0.5)\times10^{-3}$
block w=16	6.4	$(9.6 \pm 4.1)\times10^{-6}$	0.20 ± 0.01	$(1.4 \pm 0.5)\times10^{-3}$
block w=64	22.7	$(8.0 \pm 5.1)\times10^{-6}$	0.23 ± 0.01	$(1.8 \pm 0.5)\times10^{-3}$
block w=256	86.6	$(9.7 \pm 4.6)\times10^{-6}$	0.24 ± 0.01	$(1.7 \pm 0.5)\times10^{-3}$
PCA eigenbasis	241.3	$(6.7 \pm 2.0)\times10^{-6}$	0.31 ± 0.01	$(1.7 \pm 0.3)\times10^{-3}$
block w=768 (dense)	256.8	$(9.3 \pm 5.4)\times10^{-6}$	0.25 ± 0.01	$(1.8 \pm 0.6)\times10^{-3}$
DCT-II	512.7	$(8.2 \pm 5.2)\times10^{-6}$	0.25 ± 0.01	$(1.7 \pm 0.7)\times10^{-3}$

Table 12: Dense-rotation E_D at the first crossing of each FVE level (mean over seeds; layer 6; "–" means the level is never reached in the arm's budget). The 3×10^{-2}-learning-rate SGD arm diverges and is excluded.

Optimizer	FVE ≥0.45	FVE ≥0.55	FVE ≥0.65	FVE ≥0.70	final FVE	final E_D
SGD (lr 10^{-2}, 20 ep.)	1.6×10^{-6}	7.5×10^{-6}	–	–	0.559	8.7×10^{-6}
SGD (lr 10^{-2}, 100 ep.)	2.0×10^{-6}	8.5×10^{-6}	7.1×10^{-5}	–	0.677	1.8×10^{-4}
Momentum SGD (20 ep.)	3.0×10^{-5}	1.5×10^{-4}	1.1×10^{-3}	4.7×10^{-3}	0.704	5.5×10^{-3}
Scalar-RMS Adam (20 ep.)	2.4×10^{-6}	6.2×10^{-5}	5.9×10^{-4}	–	0.689	1.7×10^{-3}
Adam (20 ep.)	3.7×10^{-2}	6.1×10^{-2}	0.11	0.23	0.707	0.25

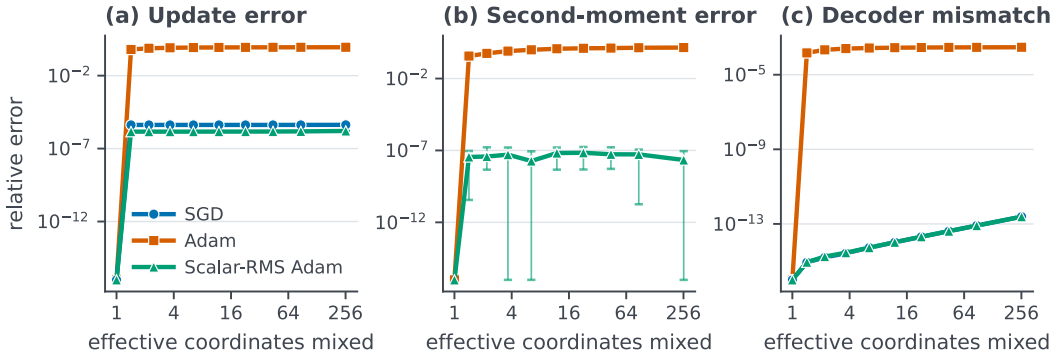


Figure 7: One-step companion to Figure 6: update and second-moment equivariance errors, and decoder mismatch after a single step, versus mixing density (5 seeds; float32; curves are means and error bars span the observed seed range; exact zeros are placed at 10^{-16} only for log display). Adam's one-step error is already $O(1)$ at minimal mixing and monotone thereafter (Spearman $r_s = 1.0$); SGD's measured decoder mismatch is zero, and scalar-RMS Adam remains at floating-point roundoff.

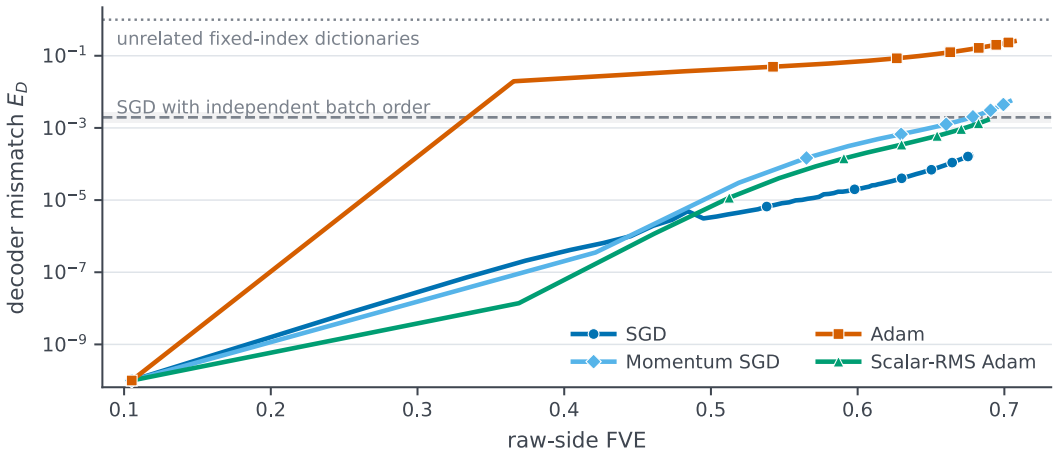


Figure 8: Decoder mismatch as a function of reconstruction quality (Section 5, *Repair at comparable reconstruction quality*; layer 6, dense rotation, epoch-mean over seeds, parametric in training time). At the shared FVE crossings, Adam has higher mismatch than the equivariant optimizers. The dashed gray line and light band show the mean and observed seed range for SGD with a scrambled minibatch order.

I LEARNING-RATE SWEEP FOR GROUP-RMS

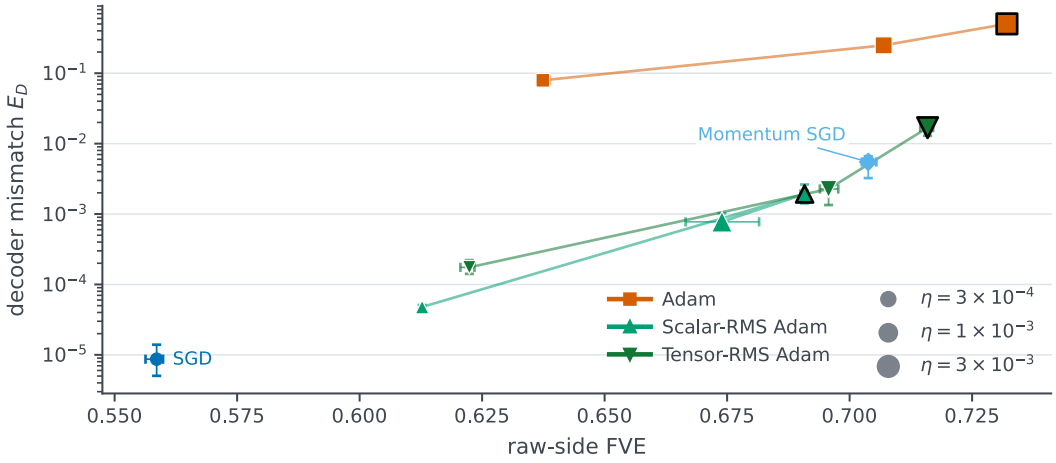


Figure 9: Reconstruction quality and decoder mismatch across learning rates (layer 6, dense rotation, 20 epochs, 3 seeds; points are means, error bars span the observed seed range, marker size encodes learning rate, and best-FVE points are outlined; SGD/momentum context comes from the main grid). At the best observed learning rates, tensor-RMS Adam reaches FVE $= 0.716$ versus Adam's 0.732, with $30\times$ lower mismatch.

Table 13: Learning-rate sweep (layer 6, 20 epochs, 3 seeds): final FVE and dense-rotation E_D per learning rate. Best-FVE rows per optimizer in bold.

Optimizer	lr	final FVE	dense E_D
Adam	3×10^{-4}	0.6374 ± 0.0012	7.9×10^{-2}
Adam	10^{-3}	0.7070 ± 0.0005	0.25
Adam	$\mathbf{3\times10^{-3}}$	$\mathbf{0.7321 \pm 0.0003}$	**0.50**
Scalar-RMS	3×10^{-4}	0.6128 ± 0.0001	4.8×10^{-5}
Scalar-RMS	$\mathbf{10^{-3}}$	$\mathbf{0.6908 \pm 0.0008}$	$\mathbf{1.9\times10^{-3}}$
Scalar-RMS	3×10^{-3}	0.6739 ± 0.0075	7.7×10^{-4}
Tensor-RMS	3×10^{-4}	0.6224 ± 0.0016	1.8×10^{-4}
Tensor-RMS	10^{-3}	0.6957 ± 0.0019	2.3×10^{-3}
Tensor-RMS	$\mathbf{3\times10^{-3}}$	$\mathbf{0.7159 \pm 0.0014}$	$\mathbf{1.7\times10^{-2}}$

Adam's quality is still increasing at the largest learning rate tested, along with its mismatch. Scalar-RMS Adam is 0.041 below Adam in FVE; tensor-RMS Adam reduces this difference to 0.016. At 3×10^{-3}, tensor-RMS Adam has float32 mismatch 0.0168, a dense-to-signed mismatch ratio of 1.03 (Adam: 6.6), and $E_D = 0$ in every seed of the matched float64 run. The dense-to-signed ratio and the matched float64 result are consistent with finite-precision drift at this learning rate.

J SYNTHETIC GROUND-TRUTH RECOVERY

Recovery. Across the six primary configurations (Table 14), Adam's pair mismatch exceeds SGD's by at least 2×10^4, while the raw and rotated models differ by at most 0.012 in mean cosine to the true dictionary.

Assignment in the synthetic control. In the canonical configuration ($m = 64$, no noise, three seeds), Adam's assignment remains unchanged at epoch 20, with $E_D^{\text{match}} = E_D = 0.041$ (Table 5). At epoch 100, $E_D = 0.218 \pm 0.132$ becomes $E_D^{\text{match}} = 0.083 \pm 0.031$: matching removes 62% of the mean mismatch, and 30% of latents change their assigned index. Raw and rotated ground-truth cosines are 0.916 and 0.910. For SGD and scalar-RMS Adam, the assignment stays at the identity, with $E_D^{\text{match}} = E_D < 10^{-3}$.

Table 14: Synthetic ground-truth recovery ($d = 32$, generation sparsity $k = 4$, dense rotation, 100 epochs, 3 seeds; mean over seeds). "Coher." is the mutual coherence $\max_{i \neq j} |d_i^\top d_j|$ of the unit ground-truth atoms; "GT" is the Hungarian-matched cosine to the true dictionary. The last row is an exploratory follow-up with 200 epochs and twice the training data.

Config (noise, m)	Coher.	GT raw	GT rot.	Adam E_D	SGD E_D
0.0, 64	0.603	0.916	0.910	0.218	1.3×10^{-6}
0.0, 256	0.697	0.889	0.891	0.256	1.2×10^{-5}
0.1, 64	0.603	0.913	0.904	0.241	1.7×10^{-6}
0.1, 256	0.697	0.895	0.888	0.247	6.5×10^{-6}
0.3, 64	0.603	0.922	0.934	0.216	1.2×10^{-6}
0.3, 256	0.697	0.800	0.803	0.199	9.5×10^{-6}
follow-up: 0.0, 32	0.523	0.903	0.941	0.293	9.8×10^{-8}

K Decoder Disagreement After Latent Matching: Full Tables

Table 15: Decoder disagreement after latent matching for Adam under a dense rotation (20 epochs, 5 seeds, independent rotations): fixed-index mismatch E_D, matched mismatch E_D^{match} (7), fraction ρ of relabeled latents and their share of E_D, index-free reconstruction disagreement E_{rec}, and mean per-latent activation correlation under π. Layer-6 rails: Table 17.

Layer	E_D	E_D^{match}	ρ (%)	share of E_D	E_{rec}	act. corr.
$\ell = 0$	0.436 ± 0.019	0.410 ± 0.016	15.9 ± 2.2	0.29	0.278	0.62
$\ell = 3$	0.315 ± 0.008	0.296 ± 0.009	10.1 ± 0.9	0.23	0.329	0.73
$\ell = 6$	0.253 ± 0.013	0.237 ± 0.010	8.9 ± 1.2	0.25	0.368	0.74
$\ell = 9$	0.232 ± 0.011	0.216 ± 0.010	9.3 ± 1.4	0.29	0.368	0.75
$\ell = 11$	0.268 ± 0.015	0.241 ± 0.011	13.2 ± 2.1	0.35	0.356	0.71

The matched analysis uses the core-grid configuration of Section C and the metrics of Section 3.2. The SGD-family embedding-stream cells use learning rate 10^{-3}, as in Table 1. The layer-6 seed-reference comparisons use three seeds.

Adam under signed permutations. Adam's assignment remains nearly the identity under signed permutations: $\rho \leq 0.1\%$ on every layer, $E_D^{\text{match}} = E_D = 0.008$–$0.031$, and $E_{\text{rec}} = 0.13$–0.16, the finite-precision level that TopK amplifies (Section 5, *Mechanism and numerical controls*). Its dense-rotation rows are in Table 15; the equivariant optimizers' dense-rotation cells are in Table 16.

Reading the rails. After 20 epochs, independent initialization gives $E_D^{\text{match}} \approx 0.5$ for optimizers reaching FVE ≈ 0.7; also varying minibatch order changes that mean little. Adam's coupled mismatch relative to its initialization reference is shown in Section E; its reconstruction disagreement is 0.82 of that reference. Adam's order-only mismatch is 0.277 under signed permutations; order and dense rotation together give 0.324. The equivariant optimizers' order references are lower at similar quality, with coupled values at the finite-precision levels of Table 7. The separate independent-training configuration in Section 6 uses $m = 2048$, $k = 32$, and 40 epochs of AdamW.

Table 16: Equivariant optimizers under a dense rotation (5 seeds; mean ± s.d.). E_D^{match} equals E_D to the precision shown in every cell; ρ is the mean relabeled fraction (its maximum over all seeds and cells is 0.5%, momentum SGD at layer 3). †Embedding stream: SGD and momentum use the 10^{-3} learning rate.

Layer	Optimizer	E_D	ρ (%)	E_{rec}	act. corr.
$\ell = 0^{\dagger}$	SGD	$(9.5 \pm 8.3)\times10^{-9}$	0	0.015	0.999
	Momentum SGD	$(2.5 \pm 0.8)\times10^{-5}$	0	0.035	0.995
	Scalar-RMS Adam	$(2.3 \pm 1.1)\times10^{-3}$	0	0.079	0.986
$\ell = 3$	SGD	$(7.6 \pm 4.7)\times10^{-5}$	0	0.093	0.993
	Momentum SGD	$(1.6 \pm 1.0)\times10^{-2}$	0.2	0.138	0.967
	Scalar-RMS Adam	$(2.6 \pm 1.8)\times10^{-3}$	0	0.097	0.984
$\ell = 6$	SGD	$(8.7 \pm 3.5)\times10^{-6}$	0	0.039	0.997
	Momentum SGD	$(5.5 \pm 1.4)\times10^{-3}$	0	0.128	0.975
	Scalar-RMS Adam	$(1.7 \pm 0.6)\times10^{-3}$	0	0.100	0.984
$\ell = 9$	SGD	$(6.2 \pm 1.8)\times10^{-6}$	0	0.034	0.997
	Momentum SGD	$(4.4 \pm 3.1)\times10^{-3}$	0	0.118	0.977
	Scalar-RMS Adam	$(1.3 \pm 0.5)\times10^{-3}$	0	0.094	0.984
$\ell = 11$	SGD	$(5.2 \pm 1.5)\times10^{-6}$	0	0.030	0.997
	Momentum SGD	$(3.6 \pm 2.0)\times10^{-3}$	0	0.106	0.975
	Scalar-RMS Adam	$(1.3 \pm 0.3)\times10^{-3}$	0	0.088	0.980

Table 17: Layer-6 comparison: matched decoder mismatch and index-free reconstruction disagreement of the coupled test (5 seeds) and of the seed-reference rails (3 seeds). The order rail is listed under both transform families for Adam; for the equivariant optimizers the two families agree to the precision shown, so only the dense values are given.

Optimizer	Rail	Transform	E_D^{match}	E_{rec}	act. corr.
Adam	coupled	dense	0.237 ± 0.010	0.368 ± 0.005	0.74
	order	signed	0.277 ± 0.005	0.379 ± 0.002	0.71
	order	dense	0.324 ± 0.003	0.398 ± 0.001	0.67
	init	dense	0.510 ± 0.005	0.446 ± 0.002	0.51
	both	dense	0.510 ± 0.006	0.450 ± 0.002	0.51
Momentum SGD	coupled	dense	$(5.5 \pm 1.4)\times10^{-3}$	0.128 ± 0.007	0.98
	order	dense	0.171 ± 0.002	0.325 ± 0.004	0.80
	init	dense	0.490 ± 0.004	0.440 ± 0.002	0.51
	both	dense	0.491 ± 0.003	0.444 ± 0.001	0.51
Scalar-RMS Adam	coupled	dense	$(1.7 \pm 0.6)\times10^{-3}$	0.100 ± 0.007	0.98
	order	dense	0.105 ± 0.005	0.291 ± 0.009	0.85
	init	dense	0.538 ± 0.001	0.458 ± 0.001	0.47
	both	dense	0.537 ± 0.001	0.460 ± 0.001	0.47
SGD	coupled	dense	$(8.7 \pm 3.5)\times10^{-6}$	0.039 ± 0.006	1.00
	order	dense	$(2.0 \pm 0.8)\times10^{-3}$	0.144 ± 0.019	0.97
	init	dense	0.767 ± 0.001	0.584 ± 0.005	0.24
	both	dense	0.766 ± 0.001	0.583 ± 0.006	0.24

L Independent Training and Reconstruction Stitching

With the configuration of Section C, the mean Hungarian-matched decoder-atom cosine is 0.370 for seeds trained on the same data (median 0.328; 5.4% above 0.7). It is $0.368/0.370$ for signed/dense raw-versus-rotated runs, whose mismatch ratios to the seed control are 1.003 and 1.000. Stitching raw seed 0 into seed 1 leaves 51%, 46%, 12%, and 5% of the bridgeable reconstruction gap under permutation, per-feature affine, TopK-re-sparsified dense, and unconstrained dense maps, respectively.