SFDA: SPARSE FEATURE DISTRIBUTION ALIGNMENT FOR POST-PRUNING RECOVERY DATA-SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured pruning is used to reduce the inference cost of large language models. However, this degrades their capabilities, which then requires recovery fine-tuning. Under a limited recovery-data budget, existing instruction-selection methods usually rank samples by difficulty, quality or semantic diversity while pruning-aware approaches may require access to both the original and pruned models. However, these criteria do not directly characterize the representation of the model, needing repair. We propose **SFDA**, a **S**parse **F**eature **D**istribution **A**lignment framework that selects recovery data directly from the *internal state* of the pruned model. **SFDA** identifies response-salient hidden coordinates using loss gradients, encodes their activations with a *TopK sparse autoencoder*, and selects a fixed-size subset whose projected sparse-feature distribution matches that of the candidate corpus. SFDA optimizes the subset jointly and requires only the pruned checkpoint. Across different pruned LLM configurations, **SFDA** achieves the highest average reasoning accuracy using only 10% of Alpaca, outperforming the strongest competing selector by 2.36-4.36 percentage points.

1 INTRODUCTION

Structured pruning removes parameter groups from a large language model (LLM) to reduce memory use and inference cost Ling et al. (2024); Hu et al. (2025); Ma et al. (2023); An et al. (2023). It can also damage capabilities unevenly, so practical pruning pipelines usually include recovery fine-tuning Ma et al. (2023); An et al. (2023). When the recovery budget is limited, the choice of training examples becomes as important as the tuning procedure itself.

Existing instruction selectors emphasize quality, uncertainty, difficulty, or coverage in an external embedding space Li et al. (2023a); Liu et al. (2024a); Li et al. (2023b); Liu et al. (2024b); Wang et al. (2024). These signals describe the data more directly than they describe the model that needs repair. Pruning changes activations, losses, and per-example updates, so examples that were useful before pruning may become redundant afterward. Conversely, semantically similar examples may reveal different failures in the pruned model. PASER He et al. (2025) captures this dependence by comparing the original and pruned models, but it requires access to both checkpoints.

We formulate limited-budget recovery as follows. Given a pruned model, a candidate corpus, and a fixed recovery budget, the objective is to retain the corpus-level signal required to recover that model. Let $\mathcal{A}_T$ denote a fixed recovery procedure and $\mathcal{R}_{\text{rec}}$ the downstream recovery risk. The ideal subset satisfies

$$\mathcal{S}^\star \in \arg\min_{\substack{\mathcal{S}\subseteq\mathcal{D}\\ |\mathcal{S}|=B}} \mathcal{R}_{\text{rec}}(\mathcal{A}_T(f_p,\mathcal{S}))\,. \tag{1}$$

Directly optimizing Equation 1 is infeasible because every candidate subset would require a separate recovery run and downstream evaluation. We instead use a *target-state coreset*: a compact subset that preserves response-relevant patterns expressed by the candidate corpus in the pruned model. Equation 1 remains the task objective, while distribution alignment provides a score that can be evaluated before recovery. We do not assume that the two objectives are globally equivalent.

Sparse Feature Distribution Alignment (SFDA) follows three stages. Response-loss gradients first identify a shared set of salient hidden coordinates in the pruned model. A TopK sparse autoencoder (SAE) then factorizes the response-position activations on those coordinates. Finally, SFDA

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.

selects exactly B examples by matching the projected sparse-code distribution of the subset to the candidate corpus. A Gaussian Bhattacharyya term compares means and covariances, while a projected Kolmogorov–Smirnov term compares shape along one fixed principal direction. Response loss determines where SFDA measures the representation; it does not rank examples.

The sparse codes make feature occurrence and co-activation easy to measure without assigning a human-readable meaning to each latent. Section 3 gives local conditions under which matching projected-code moments can reduce part of the mismatch between recovery updates. It also identifies two quantities that SFDA does not control: variation discarded by PCA and variation not predicted by the representation. We treat this link as a conditional motivation and test it with target-state, dense-representation, objective, and gradient-prediction controls.

We evaluate SFDA on pruned LLaMA and Qwen models using several pruning operators and benchmarks for language modeling, reasoning, coding, and mathematics. With a 10% Alpaca budget, SFDA obtains the highest average reasoning score in all four main model settings. We also vary the random seed, recovery budget, pruning operator, candidate corpus, SAE source model, and task domain. Perplexity and downstream accuracy are reported separately when they disagree.

Our contributions are summarized as follows:

- We identify *target-state mismatch* in post-pruning recovery-data selection. Pruning changes model representations and recovery gradients, such that selection signals defined outside the pruned state need not reflect the data distribution most useful for recovery. This motivates constructing the recovery subset directly from the state of the model being repaired.
- We formulate limited-budget post-pruning recovery as a *target-state coreset selection* problem and characterize when representation-level distribution matching can approximate recovery updates. Our analysis connects subset-gradient mismatch to the local recovery objective and makes explicit the approximation errors introduced by representation insufficiency and dimensionality reduction.
- We propose **SFDA**, which performs fixed-budget recovery-data selection in the sparse representation space of the pruned model. SFDA identifies response-salient hidden coordinates, encodes their activations with a TopK sparse autoencoder, and searches for a subset whose projected sparse-feature distribution matches that of the candidate corpus. The complete curation pipeline requires only the pruned checkpoint.
- We evaluate SFDA across multiple LLMs, pruning methods, recovery budgets, candidate corpora, and downstream tasks. With only 10% of Alpaca, SFDA achieves the highest average reasoning score across all four main model configurations, outperforming the strongest competing selector by 2.36–4.36 percentage points. Its advantage persists across multiple seeds and matched-training-budget controls.

2 Related Work

2.1 LLM Pruning and Post-Pruning Recovery

Pruning methods differ in what they remove and how they estimate importance. Structured approaches remove attention heads, MLP channels, or transformer layers Ling et al. (2024); Hu et al. (2025). LLM-Pruner Ma et al. (2023) scores coupled parameter groups, FLAP An et al. (2023) uses activation fluctuations, and Wanda Sun et al. (2023) combines weights with input activations for unstructured and $N{:}M$ sparsity. These methods choose the parameters that remain; SFDA chooses the data used to recover them.

2.2 Data Selection for Instruction Tuning

Instruction selection adapts principles from active learning and coreset construction Davis & Hwang (1992); Lewis (1995). DEITA Liu et al. (2024b) combines complexity, quality, and diversity. IFD Li et al. (2023a) compares conditional and unconditional response losses, Nuggets Li et al. (2023b) measures the effect of an example on anchor-set perplexity, and SelectIT Liu et al. (2024a) incorporates uncertainty into model-generated ratings. MONA Ma et al. (2025) represents examples through

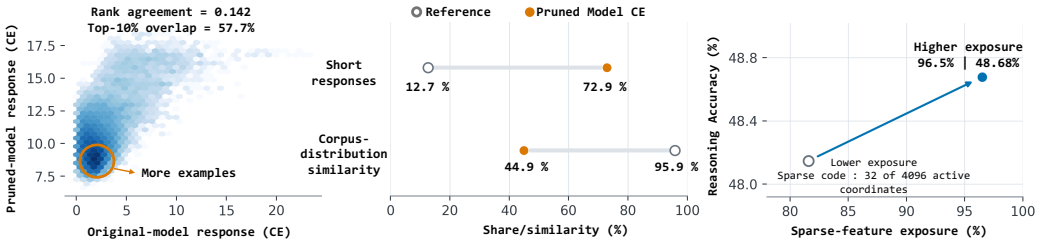

Figure 1: **Why recovery selection should be target-state and set-level.** (a) Pruning substantially changes response-loss rankings. (b) Target-state difficulty alone can produce a distributionally biased subset. (c) Broader sparse-feature exposure is associated with better recovery.

sparse monosemantic neuronal activations and selects task-relevant instruction data in that representation. Other methods emphasize representational diversity Wang et al. (2024); Yu et al. (2024), data-quality indicators Cao et al. (2023), or direct curation, as in Alpagasus Chen et al. (2023) and LIMA Zhou et al. (2024). Although several scores are model-dependent, they generally rank examples individually or measure coverage outside the post-pruning state. SFDA instead optimizes a set-level distribution match in a representation produced by the recovery target itself.

2.3 Pruning-Aware Selection and SAE Representations

PASER He et al. (2025) is the closest pruning-aware selector. It groups instructions by semantic capability, compares original- and pruned-model outputs, and uses the resulting divergence to allocate the recovery budget. SFDA shares the goal of pruning-aware recovery, but it uses only the representation of the pruned checkpoint during curation.

Sparse autoencoders provide the other foundation for our method. They produce sparse features for representation analysis Templeton et al. (2024); Karvonen et al. (2024) and have also been used for data curation. SFDA adopts response-loss-gradient saliency and SAE factorization from FineScope Bhattacharyya & Kim (2025), but solves a different problem. FineScope retrieves specialized, self-generated data; SFDA selects a fixed-size recovery set from an existing corpus. It fits the projection on that corpus and minimizes a combined Bhattacharyya and KS discrepancy. SFDA neither retrieves from domain seeds, as FineScope does, nor queries the original checkpoint, as PASER does. We use the sparse codes as coverage coordinates, not as semantic or causal capabilities.

3 Recovery-Data Requirements After Pruning

Recovery as a target-state coreset problem. Let f_p denote the pruned model and $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ the candidate recovery corpus. Given a budget $B \ll N$, we seek a subset $\mathcal{S} \subseteq \mathcal{D}$ with $|\mathcal{S}| = B$ that yields strong post-recovery performance, as formalized by Equation 1. Directly searching this objective is impractical because evaluating each candidate subset would require a separate recovery run.

The key question is therefore what information a pre-recovery surrogate should preserve. Pruning changes the model's outputs, internal representations, and per-example recovery updates. A useful subset should consequently be defined from the *target state being repaired*, rather than only from the original model or an external representation. Moreover, under a fixed budget, it should preserve the aggregate structure of the candidate corpus rather than simply retain the highest-scoring examples independently. We therefore view limited-budget recovery as a *target-state coreset* problem.

Target-state and set-level selection Figure 1 gives three diagnostics motivating this formulation. First, pruning substantially changes which examples appear difficult: the response-loss rankings of the original and pruned checkpoints have weak rank agreement (0.142), and only 57.7% of their top-10% examples overlap (Figure 1(a)). Thus, a signal measured before pruning need not describe the model that will actually be recovered.

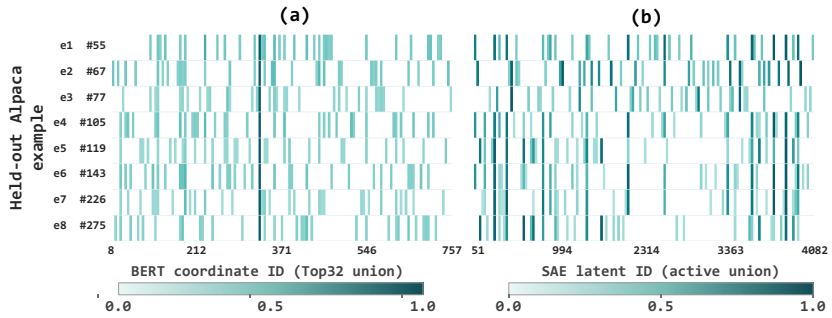

Figure 2: **Sparse-support diagnostic on matched Alpaca examples.** The same eight examples are represented using (a) the 32 largest-magnitude BERT coordinates and (b) the 32 active TopK-SAE latents. BERT is nearly dense across the full diagnostic set, whereas the SAE uses exactly 32 of $4,096$ latents per example.

Target-state difficulty alone is not enough. Selecting the top 10% of examples by pruned-model response loss strongly over-represents short responses (72.9% versus 12.7% in the reference corpus) and reduces corpus-distribution similarity from 95.9% to 44.9% (Figure 1(b)). Pointwise ranking can therefore spend much of the budget on repeatedly represented failure patterns.

Figure 1(c) gives the complementary set-level view. Under the same recovery budget, broader exposure to sparse target-state features is associated with better post-recovery reasoning. This motivates preserving coverage at the *subset level*. It does not imply that individual sparse features causally determine recovery, and SFDA does not directly maximize binary feature coverage; the implemented method instead matches continuous projected sparse-code distributions.

Sparse target-state features Set-level selection requires a representation in which coverage and redundancy can be measured explicitly. Dense representations distribute information over many simultaneously active coordinates, making it difficult to distinguish which directions repeatedly occur across examples. SFDA instead factorizes response-salient activations of the pruned model using a TopK sparse autoencoder (SAE). Each example is represented by only a small active support, making feature occurrence, activation strength, and co-activation directly measurable without requiring the latents to have human-interpretable meanings.

Figure 2 illustrates this difference on matched Alpaca examples. The BERT representation is effectively dense, activating an average of 767.9985 out of 768 coordinates, whereas the TopK SAE activates exactly 32 out of $4,096$ latents while attaining held-out reconstruction FVU of 0.0888. Sparsity therefore provides an operational coordinate system for reasoning about which patterns are represented by a limited recovery set.

Optimize the subset jointly A fixed recovery budget must allocate examples across both common and less-frequent patterns. Ranking examples independently cannot control how selected examples interact: many individually useful examples may provide nearly redundant recovery signals. Conversely, matching only marginal feature occurrence can miss differences in activation magnitude and co-activation.

SFDA therefore evaluates the selected set *jointly*. Rather than maximizing a per-example score or directly maximizing binary feature coverage, it searches for a fixed-size subset whose continuous projected sparse-code distribution approximates that of the complete candidate corpus. The method in Section 4 implements this principle by matching location, co-variation, and a complementary one-dimensional distributional statistic.

Figure 3 provides an intuitive visualization. SFDA-selected examples appear across multiple global regions and local instruction neighborhoods rather than concentrating in only a few regions. The visualization uses external instruction embeddings rather than SFDA's PCA coordinates, so it should be interpreted only as a semantic-coverage diagnostic, not as a visualization of the optimized objective.

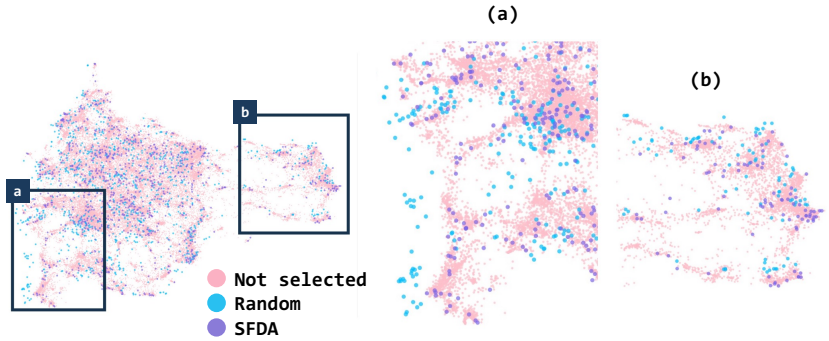

Figure 3: **Selection-space visualization for SFDA.** The overview shows a two-dimensional UMAP of Alpaca instruction embeddings; the enlarged regions illustrate local selection behavior. Pink denotes examples excluded by both selectors, blue denotes Random, and violet denotes SFDA. The UMAP is used only to visualize semantic coverage and is not SFDA's optimization space.

Recovery updates. A one-step expansion of the recovery objective shows that a useful subset should approximately reproduce the aggregate update induced by the complete candidate corpus. If variation in those local recovery updates is sufficiently represented by the target-state sparse codes, then matching appropriate code statistics can reduce part of this update mismatch.

This relationship is deliberately conditional. The SAE representation can omit gradient-relevant information, PCA can discard relevant sparse-code directions, and the remaining representation-to-gradient relation need not be globally linear. Our direct gradient-proxy diagnostic indeed finds weak global affine predictability. We therefore use this analysis only to explain *when* target-state distribution matching can serve as a recovery surrogate, rather than as a causal explanation of SFDA's empirical gains. Appendix 7.18 provides the full derivation and explicitly retains the uncontrolled representation and projection residuals.

4 Sparse Feature Distribution Alignment

We now define the representation and set-level objective. Every fitted quantity is estimated from the candidate corpus and frozen before the swap search. Figure 4 illustrates the overview of our proposed method.

Response-salient activation coordinates. Let $H_i^{(L)} \in \mathbb{R}^{T_i \times d_h}$ denote the final post-normalization hidden states of f_p, where T_i is the number of non-padding tokens. Let $\mathcal{T}_i^y$ contain the hidden-state positions used to predict response tokens. Following FineScope Bhattacharyya & Kim (2025), we measure the response-loss saliency of hidden coordinate j by

$$s_{i,j} = \sum_{t \in \mathcal{T}_i^y} \left| \frac{\partial \ell_i}{\partial H_{i,t,j}^{(L)}} \right|. \tag{2}$$

If we selected coordinates separately for each example, descriptor position j would refer to different hidden dimensions across the corpus. We instead select one coordinate set and reuse it for every example. Let $\mathcal{D}_{\text{ref}} \subseteq \mathcal{D}$ be a fixed candidate-only reference sample. We compute

$$\bar{s}_j = \frac{1}{|\mathcal{D}_{\text{ref}}|} \sum_{(x_i,y_i) \in \mathcal{D}_{\text{ref}}} s_{i,j}, \qquad \mathcal{I}_{K_h} = \text{sort}\big(\text{TopKIndices}_j(\bar{s}_j, K_h)\big). \tag{3}$$

We set $K_h = 128$, store the chosen coordinates in ascending physical-index order, and keep $\mathcal{I}_{K_h}$ fixed across the corpus. The controlled validation uses 128 reference examples for the 2,048-example pool and 512 for the near-full-corpus systems run; in both cases the reference is fixed before descriptor extraction and excluded from evaluation.

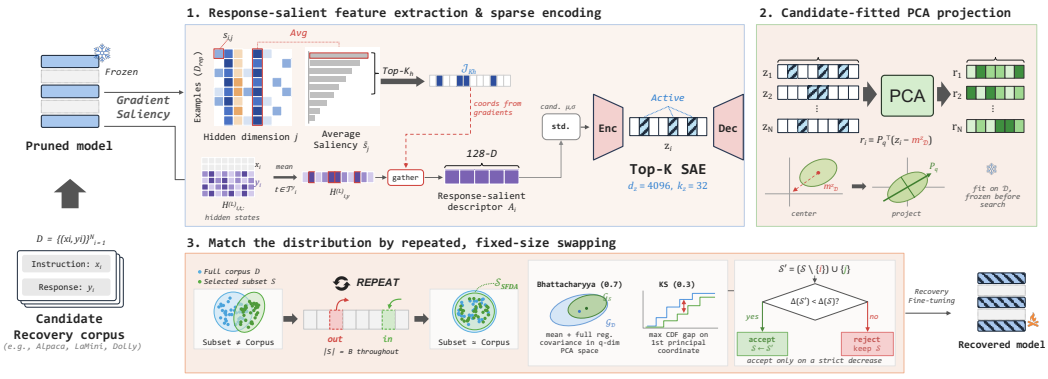

Figure 4: Overview of SFDA

We then mean-pool the final hidden states over response-prediction positions and gather those same coordinates:

$$\overline{H}^{(L)}_{i,y} = \frac{1}{|\mathcal{T}^y_i|} \sum_{t \in \mathcal{T}^y_i} H^{(L)}_{i,t,:}, \qquad A_i = \left[\overline{H}^{(L)}_{i,y}\right]_{\mathcal{I}_{K_h}} \in \mathbb{R}^{K_h}. \tag{4}$$

Response-only pooling in Equation 4 is the rule used by every reported SFDA experiment.

Standardization. Before fitting the SAE, we estimate a coordinate-wise mean μ_j and standard deviation σ_j from the candidate corpus and form

$$\widetilde{A}_{i,j} = \frac{A_{i,j} - \mu_j}{\max(\sigma_j, \epsilon_A)}. \tag{5}$$

No evaluation example contributes to these statistics.

TopK sparse autoencoder. We encode the standardized descriptor using a nonnegative TopK SAE Templeton et al. (2024); Karvonen et al. (2024):

$$u_i = \mathrm{ReLU}(W_e \widetilde{A}_i + b_e), \qquad z_i = \mathrm{TopK}_{k_z}(u_i), \tag{6}$$

where $W_e \in \mathbb{R}^{d_z \times K_h}$ and TopK_{k_z} retains the k_z largest entries and sets the remaining entries to zero. We use $d_z = 4096$ and $k_z = 32$.

The decoder reconstructs $\widehat{A}_i = W_d z_i + b_d$. We fit the SAE by minimizing

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \left\| \widetilde{A}_i - \widehat{A}_i \right\|_2^2, \qquad \|(W_d)_{:,j}\|_2 = 1 \quad \forall j, \tag{7}$$

After each update, we project every decoder column to unit norm. The TopK operator supplies the sparsity; we add no ℓ_1 penalty. Because ReLU precedes TopK, each code has at most k_z positive entries. We freeze the SAE during subset search and discard it after curation.

For the benchmark experiments, the SAE is fitted for two epochs with the `sae-lens` implementation, learning rate 5×10^{-5}, batch size four, and ten warmup updates.

Distribution-aligned subset construction. Binary exposure in Equation 20 ignores activation magnitude. SFDA instead compares continuous code distributions so that both support and magnitude affect the subset.

The distribution objective is defined directly in a candidate-fitted PCA space. Let $m^z_{\mathcal{D}} = N^{-1} \sum_{i \in \mathcal{D}} z_i$, and let $P_q \in \mathbb{R}^{d_z \times q}$ contain the q leading orthonormal eigenvectors of the population covariance of the centered candidate codes. The projected representation is

$$r_i = P_q^\top (z_i - m^z_{\mathcal{D}}) \in \mathbb{R}^q, \qquad q = 16. \tag{8}$$

We fit $m_{\mathcal{D}}^z$ and P_q once and freeze them during search. PCA is a computational approximation, not a sufficiency claim. The discarded component $d_i = (I_{d_z} - P_q P_q^\top)(z_i - m_{\mathcal{D}}^z)$ can contain recovery-relevant or rare-feature variation. Appendix 7.18.5 keeps this component explicit in the gradient-mismatch bound. For any nonempty $\mathcal{A} \subseteq \mathcal{D}$, define

$$\begin{aligned} \mu_{\mathcal{A}} &= \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} r_i, \\ \Sigma_{\mathcal{A}}^{\epsilon} &= \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} (r_i - \mu_{\mathcal{A}})(r_i - \mu_{\mathcal{A}})^\top + \epsilon_\Sigma I_q, \qquad \epsilon_\Sigma = 10^{-3}, \\ \mathcal{G}_{\mathcal{A}} &= \mathcal{N}(\mu_{\mathcal{A}}, \Sigma_{\mathcal{A}}^{\epsilon}). \end{aligned} \tag{9}$$

The Gaussian approximation uses a full, regularized $q \times q$ covariance and never forms a $d_z \times d_z$ covariance. Its link to stochastic-update dispersion depends on the conditional model in Equation 33. We test covariance matching as a component in Table 7; we do not assume the mechanism holds. To compare shape beyond these moments, we compute a KS statistic on the first frozen principal coordinate:

$$\widehat{F}_{\mathcal{A},1}(a) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathbf{1}[r_{i,1} \le a], \qquad D_{\mathrm{KS}}(\mathcal{D}, \mathcal{S}) = \sup_a \left| \widehat{F}_{\mathcal{D},1}(a) - \widehat{F}_{\mathcal{S},1}(a) \right|. \tag{10}$$

PC1 is fixed before search and maximizes candidate-code variance without using labels. This gives a stable, inexpensive one-dimensional marginal for detecting skew, tail, or multimodal mismatch. Maximum variance need not imply recovery relevance, so PC1 is a secondary heuristic and controls no other marginal. The SFDA discrepancy is

$$\Delta_{\mathrm{SFDA}}(\mathcal{S}) = w_B D_B(\mathcal{G}_{\mathcal{D}}, \mathcal{G}_{\mathcal{S}}) + w_{\mathrm{KS}} D_{\mathrm{KS}}(\mathcal{D}, \mathcal{S}), \tag{11}$$

where $w_B = 0.7$ and $w_{\mathrm{KS}} = 0.3$. The first term compares location and co-variation; the second compares one empirical marginal. This is a tractable score, not a test of equality between the full code distributions. Table 7 confirms that each term reduces its intended diagnostic, but does not show that their combination or PC1 is uniquely best for recovery.

Equations 8–11 define the objective used in the main experiments. Each representation control receives its own candidate-fitted projection. The objective control instead fixes the codes and projection and changes only the discrepancy. In Table 7, "covariance" means the full 16×16 covariance in Equation 9.

Fixed-size swap optimization. We start with B examples sampled uniformly without replacement. Each iteration draws $i \in \mathcal{S}$ and $j \in \mathcal{D} \setminus \mathcal{S}$ and proposes

$$\mathcal{S}' = (\mathcal{S} \setminus \{i\}) \cup \{j\}. \tag{12}$$

We accept the proposal only when $\Delta_{\mathrm{SFDA}}(\mathcal{S}') < \Delta_{\mathrm{SFDA}}(\mathcal{S})$. Every accepted move preserves the budget and decreases the stored objective. The search evaluates at most $100{,}000$ proposals and stops early after $10{,}000$ consecutive rejections.

5 Experimental Evaluation

We first compare SFDA with full-data training and five subset selectors. We then vary the pruning operator, random seed, source checkpoint, recovery-set size, task domain, and candidate corpus. Finally, we isolate the coordinate-selection rule and measure curation cost. The appendix reports matched controls for the representation, objective, recovery budget, and SAE capacity.

5.1 Experimental Setup

Models and pruning. We use four recovery targets: LLaMA 2 13B Touvron et al. (2023), LLaMA 3.1 8B Dubey et al. (2024), LLaMA 3.2 1B, and Qwen 3.8B Yang et al. (2025). The main experiment applies LLM-Pruner Ma et al. (2023) with pruning ratio $P_r = 0.35$. To test dependence on this choice, we also compare LLM-Pruner, FLAP An et al. (2023), and Wanda Sun et al. (2023) at $P_r = 0.15$.

	Methods	Wikitext ↓	HellaSwag ↑	BoolQ ↑	PIQA ↑	ARC-e ↑	ARC-c ↑	Average Reasoning ↑
LLaMA 3.2 1B	Pruned Model (w/o tuning)	141.32	30.17	55.99	62.57	40.07	19.88	41.74
	Alpaca-Full	27.99	**54.87**	52.11	67.24	49.83	19.69	48.75
	LaMini-Full	28.21	52.41	51.83	57.30	48.36	19.55	45.89
	GPT-J-Full	26.00	54.57	51.70	55.99	37.85	21.33	44.29
	Random Selection	97.21	50.94	53.18	59.24	46.90	19.83	46.02
	SelectIT	24.00	53.95	55.20	65.51	50.27	21.08	49.20
	IFD	24.47	54.17	50.18	65.50	50.04	22.35	48.45
	Nuggets	27.22	44.66	52.23	66.43	49.11	23.94	47.27
	MONA	24.01	45.31	53.01	66.71	50.22	24.01	47.85
	PASER	23.15	45.55	53.33	67.50	51.00	24.15	48.31
	SFDA	**22.25**	52.01	**58.00**	**69.33**	**52.21**	**26.25**	**51.56**
LLaMA 3.1 8B	Pruned Model (w/o tuning)	56.79	35.59	56.76	66.10	41.79	25.17	45.08
	Alpaca-Full	29.48	62.98	63.33	71.22	58.92	28.67	57.02
	LaMini-Full	27.33	58.09	64.82	63.92	49.25	20.82	51.38
	GPT-J-Full	35.18	57.11	**65.22**	54.07	49.18	20.37	49.19
	Random Selection	41.09	51.39	62.74	49.05	40.29	18.20	44.33
	SelectIT	26.14	62.43	63.75	71.54	59.34	30.88	57.59
	IFD	27.94	63.08	62.57	70.62	59.51	28.75	56.91
	Nuggets	24.26	64.35	63.61	71.60	60.73	30.38	58.13
	MONA	24.01	63.30	63.50	70.30	63.30	29.78	58.04
	PASER	24.85	64.50	64.10	72.50	63.33	30.15	58.92
	SFDA	**23.20**	**65.66**	64.34	**74.00**	**67.77**	**37.21**	**61.80**
LLaMA 2 13B	Pruned Model (w/o tuning)	100.87	44.95	61.67	61.11	47.18	31.31	49.24
	Alpaca-Full	31.84	55.88	51.72	68.29	67.94	27.17	54.20
	LaMini-Full	32.96	50.96	47.07	62.93	61.19	26.16	49.66
	GPT-J-Full	32.64	51.07	50.17	67.29	67.91	28.14	52.92
	Random Selection	35.17	50.17	48.11	63.27	62.69	33.20	51.49
	SelectIT	30.43	50.33	49.87	65.52	57.06	32.44	51.04
	IFD	34.21	52.34	49.23	62.71	67.31	31.75	52.67
	Nuggets	33.71	49.97	54.31	64.13	68.66	31.01	53.62
	MONA	29.35	52.78	55.89	64.67	69.69	32.40	55.09
	PASER	20.85	65.66	61.60	73.50	72.25	33.45	61.29
	SFDA	**20.39**	**68.32**	**66.75**	**75.84**	**73.98**	**43.34**	**65.65**
Qwen 3 8B	Pruned Model (w/o tuning)	143.50	39.79	53.66	55.70	35.70	18.56	40.68
	Alpaca-Full	34.54	54.35	50.56	65.24	66.32	26.67	52.63
	LaMini-Full	33.45	49.72	44.99	60.95	59.81	24.60	48.01
	GPT-J-Full	33.20	50.39	48.56	65.52	64.66	26.10	51.05
	Random Selection	36.63	48.16	46.82	61.80	61.03	31.40	49.84
	SelectIT	29.59	48.38	47.79	63.79	55.10	29.93	49.00
	IFD	34.84	50.86	46.85	60.30	65.03	29.79	50.57
	Nuggets	34.73	47.88	52.52	62.01	66.53	29.41	51.67
	MONA	28.41	50.22	57.81	69.00	68.49	39.30	56.96
	PASER	**20.22**	63.20	64.69	71.55	71.30	35.15	61.18
	SFDA	20.61	**66.35**	**64.86**	**74.35**	**71.69**	**41.00**	**63.65**

Table 1: Recovery after LLM-Pruner at $P_r = 0.35$ on four model configurations. WikiText perplexity is lower-better; the five task accuracies and their unweighted average are higher-better. Alpaca is the candidate corpus for SFDA and the other subset selectors, which use a 10% subset.

Recovery data and baselines. The default candidate pool is Alpaca Taori et al. (2023) (52K examples), from which each subset method selects 10%. We compare SFDA with uniform sampling, IFD Li et al. (2023a), SelectIT Liu et al. (2024a), Nuggets Li et al. (2023b), MONA Ma et al. (2025), and PASER He et al. (2025). Full-data controls use Alpaca, LaMini Wu et al. (2023) (2.59M examples), or GPT-J Anand et al. (2023) (710K examples). Wherever the method permits, model-dependent baselines use the same pruned checkpoint and candidate pool. PASER has different information access because it additionally queries the corresponding original model.

SAE setup. The SAE has input dimension 128, latent width $d_z = 4096$, and $k_z = 32$ active units per example. Its encoder applies a linear map, ReLU, and TopK in that order. The decoder is an untied linear layer with a bias, and its columns are projected to unit norm after every update. We minimize the mean per-example sum of squared reconstruction errors with Adam and do not use an ℓ_1 penalty. We use batch size 4, learning rate 5×10^{-5}, and 10 linear warmup updates followed by a constant learning rate. The fitted SAE is frozen before PCA and subset search.

Evaluation. We measure WikiText perplexity and accuracy on BoolQ Clark et al. (2019), HellaSwag Zellers et al. (2019), PIQA Bisk et al. (2019), ARC-Easy, and ARC-Challenge Clark et al. (2018); their unweighted mean is "average reasoning." We separately report coding and mathematics in Table 3. Randomness may enter through the saliency reference sample, SAE fit, swap search, recovery training, and evaluation. The reported aggregate deviations combine these sources and are not estimates of recovery-training variance alone.

5.2 Main Recovery Results

Table 1 reports recovery after LLM-Pruner at $P_r = 0.35$. With the same 10% Alpaca budget, SFDA achieves the highest average reasoning accuracy across all four recovery targets, outperforming the strongest competing selector by 2.36–4.36 points. It also achieves the best reported accuracy on 18 of the 20 model–task pairs.

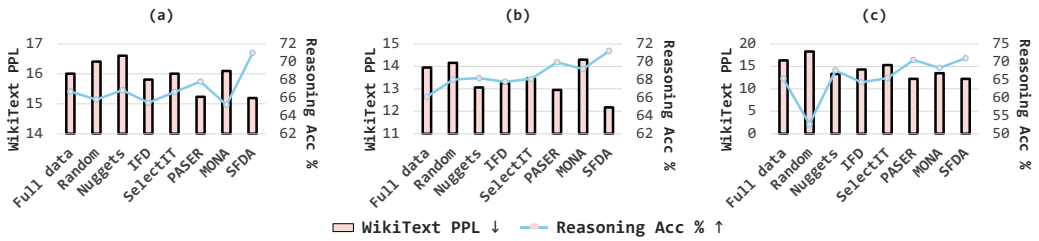

Figure 5: **Recovery across pruning schemes at $P_r = 0.15$.** (a) LLM-Pruner, (b) FLAP, and (c) Wanda. Bars give WikiText perplexity on the left axis (lower is better); connected markers give average reasoning accuracy on the right axis (higher is better). Each subset selector uses 3K examples; Full data uses the complete recovery corpus. The connecting lines distinguish the accuracy series and do not interpolate between methods.

These results support the target-state coreset view introduced in Section 3. While conventional selectors rank examples by difficulty, quality, or generic diversity, SFDA constructs its selection space directly from the pruned model and optimizes the subset jointly to preserve the candidate-corpus distribution in that state. Notably, despite using only 10% of Alpaca, SFDA exceeds Alpaca-Full in average reasoning by 2.81, 4.78, 11.45, and 11.02 points across the four targets, indicating that recovery depends strongly on data composition rather than dataset size alone.

SFDA also obtains the lowest WikiText perplexity on all three LLaMA targets. On Qwen 3 8B, PASER gives slightly lower perplexity (20.22 vs. 20.61), whereas SFDA achieves higher average reasoning (63.65 vs. 61.18), showing that language-modeling perplexity does not fully characterize downstream recovery. Since Table 1 contains single-run results, we evaluate robustness to selection and recovery randomness separately in Section 7.1.

Robustness across pruning schemes. Figure 5 evaluates whether the advantage of SFDA is specific to LLM-Pruner. Across LLM-Pruner, FLAP, and Wanda at $P_r = 0.15$, SFDA consistently achieves the highest average reasoning accuracy while maintaining the lowest or near-lowest WikiText perplexity. The separation is particularly visible under Wanda, where Random selection degrades substantially while SFDA remains competitive.

These results support the target-state motivation of our method: different pruning operators induce different model states and recovery requirements, and SFDA adapts its representation and subset selection to the checkpoint being repaired rather than relying on a fixed notion of data quality or diversity. Since an equal pruning ratio does not imply equal functional damage, we interpret the comparison within each pruning scheme rather than directly across panels.

6 Conclusion

SFDA selects recovery data from the internal state of the pruned model. It factorizes response-salient activations with a TopK SAE and searches for a fixed-size subset whose projected code distribution matches the candidate corpus without querying the original checkpoint. It achieves the highest average reasoning score in all four main model settings, and the same ordering holds over five runs for the two larger models. Matched token and optimizer budgets further show that the gain is not explained by additional training exposure. The controlled results support target-state, set-level selection while limiting the mechanistic claim: sparse codes provide useful coverage coordinates, not identified causal repair units. SFDA remains constrained by its candidate corpus and may discard rare recovery-relevant directions through PCA. Future work should strengthen the distribution objective and directly identify which target-state variation is necessary for recovery.

References

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models, 2023.

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. https://github.com/nomic-ai/gpt4all, 2023.

Chaitali Bhattacharyya and Yeseong Kim. Finescope : Precision pruning for domain-specialized large language models using sae-guided self-data cultivation, 2025. URL https://arxiv.org/abs/2505.00624.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019. URL https://arxiv.org/abs/1911.11641.

Yihan Cao, Yanbin Kang, Chi Wang, and Lichao Sun. Instruction mining: Instruction data selection for tuning large language models. *arXiv preprint arXiv:2307.06290*, 2023.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019. URL https://arxiv.org/abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

D.T. Davis and J.-N. Hwang. Attentional focus training by boundary region data selection. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pp. 676–681 vol.1, 1992. doi: 10.1109/IJCNN.1992.287109.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Bowei He, Lihao Yin, Hui-Ling Zhen, Xiaokun Zhang, Mingxuan Yuan, and Chen Ma. Paser: Post-training data selection for efficient pruned large language model recovery. *arXiv preprint arXiv:2502.12594*, 2025.

Hanyu Hu, Pengxiang Zhao, Ping Li, Yi Zheng, Zhefeng Wang, and Xiaoming Yuan. Fasp: Fast and accurate structured pruning of large language models. *arXiv preprint arXiv:2501.09412*, 2025.

Adam Karvonen, Can Rager, Samuel Marks, and Neel Nanda. Evaluating sparse autoencoders on targeted concept erasure tasks. *arXiv preprint arXiv:2411.18895*, 2024.

David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm Sigir Forum*, volume 29, pp. 13–19. ACM New York, NY, USA, 1995.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. *arXiv preprint arXiv:2308.12032*, 2023a.

Yunshui Li, Binyuan Hui, Xiaobo Xia, Jiaxi Yang, Min Yang, Lei Zhang, Shuzheng Si, Ling-Hao Chen, Junhao Liu, Tongliang Liu, et al. One-shot learning as instruction data prospector for large language models. *arXiv preprint arXiv:2312.10302*, 2023b.

Gui Ling, Ziyang Wang, and Qingwen Liu. Slimgpt: Layer-wise structured pruning for large language models. *Advances in Neural Information Processing Systems*, 37:107112–107137, 2024.

Liangxin Liu, Xuebo Liu, Derek F. Wong, Dongfang Li, Ziyi Wang, Baotian Hu, and Min Zhang. Selectit: Selective instruction tuning for llms via uncertainty-aware self-reflection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 97800–97825. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/b130a5691815f550977e331f8bec08ae-Paper-Conference.pdf.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=BTKAeLqLMw.

Da Ma, Gonghu Shang, Zhi Chen, Libo Qin, Yijie Luo, Hongshen Xu, Lei Pan, Shuai Fan, Kai Yu, and Lu Chen. Task-specific data selection for instruction tuning via monosemantic neuronal activations. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4607. URL https://papers.nips.cc/paper_files/paper/2025/hash/c971ffd786876d5927caf482b06bcba5-Abstract-Conference.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

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/index.html.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. Diversity measurement and subset selection for instruction tuning datasets. *arXiv preprint arXiv:2402.02318*, 2024.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. *arXiv preprint arXiv:2304.14402*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Simon Yu, Liangyu Chen, Sara Ahmadian, and Marzieh Fadaee. Diversify and conquer: Diversity-centric data selection with iterative refinement. *arXiv preprint arXiv:2409.11378*, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019. URL https://arxiv.org/abs/1905.07830.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

7 APPENDIX

APPENDIX INDEX

7.1 Multi-seed Experiments . . . 13
7.2 Effect of SAE Source Model and Recovery-Set Size . . . 13
7.3 Coding and Mathematical Tasks . . . 13
7.4 Comparison with LIMA and ALPAGASUS . . . 13
7.5 Candidate-Corpus Comparison . . . 14
7.6 Effect of TopK Selection Method: Magnitude vs. Gradient Magnitude . . . 15
7.7 Sparse Target-State Coresets . . . 16
7.8 Recovery Selection as Local Gradient Approximation . . . 16
7.9 Mean Updates and Stochastic Update Dispersion . . . 17
7.10 Why Measure the Pruned Target State? . . . 17
7.11 Sparse Features as a Coverage Coordinate System . . . 17
7.12 A Local Bridge from Sparse Codes to Recovery Gradients . . . 18
7.13 From Code Statistics to Recovery Statistics . . . 18
7.14 Effect of the PCA Projection . . . 19
7.15 What the SFDA Discrepancy Controls . . . 19
7.16 Budget, Rare Features, and Set-Level Coverage . . . 20
7.17 Scope of the Theoretical Argument . . . 20
7.18 Theory of Sparse Target-State Coresets . . . 21
7.18.1 Recovery Selection as Local Gradient Approximation . . . 21
7.18.2 Why Target-State, Response-Salient Features? . . . 21
7.18.3 Why Use a Sparse Autoencoder? . . . 22
7.18.4 From Code Statistics to Recovery Statistics . . . 23
7.18.5 What the SFDA Discrepancy Controls . . . 23
7.18.6 Budget, Coverage, and the Scope of the Guarantee . . . 24
7.19 Controlled Validation of the SFDA Design . . . 25
7.19.1 Representation control . . . 25
7.19.2 Objective control . . . 25
7.19.3 Controlled recovery budgets . . . 26
7.19.4 Gradient-proxy diagnostic . . . 29
7.19.5 SAE capacity . . . 29
7.19.6 Descriptor and projection dimensions . . . 30
7.19.7 Measured systems profile . . . 30

7.1 MULTI-SEED EXPERIMENTS

Table 2 reports marginal means and sample standard deviations across five runs for the two larger LLaMA models. SFDA obtains 63.30 ± 0.51 average reasoning on LLaMA 3.1 8B and 66.87 ± 0.60 on LLaMA 2 13B. The corresponding margins over PASER are 3.35 and 4.59 points; the gains over the untuned pruned checkpoints are 18.22 and 17.63 points. SFDA also exceeds Alpaca-Full by 5.10 and 11.29 points.

Perplexity follows the same ordering. SFDA reaches 23.10 ± 0.27 on the 8B model and 21.43 ± 0.21 on the 13B model. The closest alternatives are MONA at 23.35 ± 0.40 and PASER at 21.93 ± 0.09, respectively. These values are marginal aggregates, so the comparison uses differences of means rather than paired inference.

Method	LLaMA 3.1 8B		LLaMA 2 13B	
	Wikitext ↓	Avg. Reasoning ↑	Wikitext ↓	Avg. Reasoning ↑
Pruned Model (w/o tuning)	56.79	45.08	100.87	49.24
Alpaca-Full	29.75 ± 0.09	58.20 ± 0.42	33.05 ± 0.12	55.58 ± 0.41
LaMini-Full	27.35 ± 0.10	52.60 ± 0.22	33.98 ± 0.26	50.99 ± 0.45
GPT-J-Full	35.39 ± 0.11	50.60 ± 0.28	34.08 ± 0.25	54.08 ± 0.46
Random Selection	42.40 ± 0.22	45.80 ± 0.37	36.46 ± 0.28	52.72 ± 0.62
SelectIT	26.63 ± 0.29	58.61 ± 0.32	31.67 ± 0.09	52.37 ± 0.51
IFD	28.01 ± 0.15	57.97 ± 0.56	35.53 ± 0.28	54.16 ± 0.31
Nuggets	24.35 ± 0.23	59.42 ± 0.36	34.90 ± 0.09	54.95 ± 0.40
MONA	23.35 ± 0.40	57.42 ± 0.50	28.35 ± 0.50	56.42 ± 0.30
PASER	25.12 ± 0.09	59.95 ± 0.28	21.93 ± 0.09	62.28 ± 0.54
SFDA	$\mathbf{23.10 \pm 0.27}$	$\mathbf{63.30 \pm 0.51}$	$\mathbf{21.43 \pm 0.21}$	$\mathbf{66.87 \pm 0.60}$

Table 2: Five-run recovery results for two LLaMA models after LLM-Pruner at $P_r = 0.35$. Cells report marginal mean $\pm$ sample standard deviation, except the untuned pruned-model row, which gives a single reference result. These aggregates are reported separately from the single-run results in Table 1. WikiText perplexity is lower-better; average reasoning accuracy is the unweighted mean of the five tasks in Table 1 and is higher-better. Paired differences and confidence intervals are not inferable from aggregate means and standard deviations.

7.2 EFFECT OF SAE SOURCE MODEL AND RECOVERY-SET SIZE

Experimental variants. Both panels of Figure 6 recover the LLM-Pruner $P_r = 0.35$ LLaMA 3.2 1B checkpoint. In panel (a), a pruned LLaMA 3.1 8B model runs the full curation pipeline and transfers only the selected identifiers to the 1B target. In panel (b), the pruned 1B model selects its own data. Recovery starts only after selection is complete in both cases.

Effect of source checkpoint and budget. Larger recovery sets generally reduce perplexity and improve reasoning, though neither curve is monotonic. The target-native 1B model performs better at small budgets and near 36K examples, while the 8B source is better around 26K. At the largest budget, their perplexities are similar and the 1B source retains a small reasoning advantage. We keep target-native curation as the default because the same checkpoint performs selection and recovery.

7.3 CODING AND MATHEMATICAL TASKS

Table 3 evaluates whether the main-task advantage extends to code generation and mathematical reasoning. On LLaMA 3.1 8B, SFDA obtains the best MBPP, Minerva, and GSM8K scores: 37.21, 17.45, and 45.01, respectively. These exceed the strongest alternative by 0.88, 1.56, and 3.68 points. HumanEval is the exception, where SelectIT reaches 51.09 and SFDA reaches 49.22.

The 13B results show the same mixed pattern. SFDA leads HumanEval (55.44) and Minerva (20.70), whereas IFD leads MBPP (39.77 versus 38.00). On GSM8K, SFDA is the strongest subset selector at 42.78, but it is 0.10 below full-data tuning. SFDA's advantage extends beyond the five main reasoning tasks, but not to every domain or metric.

7.4 COMPARISON WITH LIMA AND ALPAGASUS

Table 4 compares SFDA with two curated instruction sets at their native scales. At 1K examples, SFDA obtains both the lowest WikiText perplexity (32.18) and the highest average reasoning score

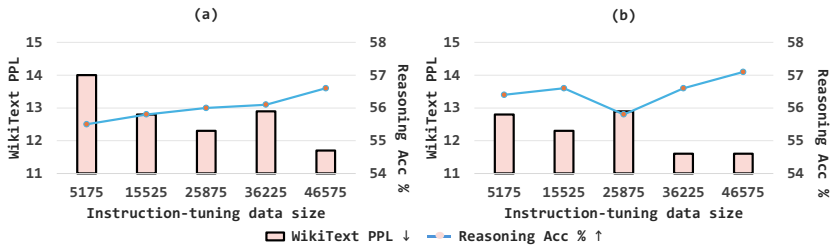

Figure 6: **Descriptor/SAE source checkpoint and recovery-set size for a pruned 1B target.** Both panels recover the LLM-Pruner $P_r = 0.35$ LLaMA 3.2 1B checkpoint. (a) Transfer variant: the LLM-Pruner $P_r = 0.35$ LLaMA 3.1 8B checkpoint generates the saliency coordinates, response-only descriptors, SAE codes, PCA statistics, and selected example identifiers; only the selected identifiers are passed to the 1B target. (b) Canonical target-native variant: that complete extraction-to-selection path and recovery training both use the pruned 1B checkpoint. The horizontal axis gives the number of instruction-tuning examples. Bars show WikiText perplexity on the left axis (lower is better), and connected markers show average reasoning accuracy on the right axis (higher is better). The lines connect discrete training-set sizes only.

Model	Method	HE	MBPP	Min.	GSM8K
8B	Pruned	32.61	30.06	4.29	1.00
	Full	50.19	36.33	13.42	35.29
	Random	37.22	24.00	12.79	36.99
	SelectIT	51.09	36.12	15.89	40.67
	IFD	46.13	31.21	14.92	37.99
	Nuggets	47.12	31.76	14.99	39.12
	MONA	47.03	30.24	15.01	38.61
	PASER	49.00	35.32	15.50	41.33
	SFDA	**49.22**	**37.21**	**17.45**	**45.01**
13B	Pruned	33.11	22.57	6.12	1.10
	Full	53.58	39.12	17.01	42.88
	Random	35.17	29.44	17.22	41.85
	SelectIT	54.24	38.25	18.95	42.44
	IFD	54.16	39.77	17.02	35.21
	Nuggets	52.19	35.17	18.73	41.00
	MONA	51.70	34.67	18.01	39.01
	PASER	54.01	37.92	20.59	42.06
	SFDA	**55.44**	**38.00**	**20.70**	**42.78**

Table 3: Coding and math accuracy (%, higher is better) after LLM-Pruner at $P_r = 0.35$. HE: HumanEval; Min.: Minerva. MONA results are included in both model blocks.

(63.89). Its reasoning average is 0.47 points above SelectIT, the strongest non-SFDA alternative, and its perplexity is 1.37 below PASER. SFDA also leads four of the five individual tasks in this block, indicating that the average is not driven by one benchmark.

At 9K examples, SFDA again achieves the highest reasoning average (65.05), although its 0.08-point margin over Nuggets indicates effectively comparable performance. PASER obtains slightly lower perplexity (33.50 versus 33.57), whereas SFDA leads BoolQ and PIQA. The two metrics give different rankings.

The separation is largest at 1K examples. Because the two blocks use different reference datasets and independently selected examples, they are not a scaling curve. They compare two useful budgets but do not isolate the effect of budget size.

7.5 Candidate-Corpus Comparison

Table 5 fixes the selected set at 1,000 examples and varies the candidate corpus. LaMini gives the strongest result, with 30.22 perplexity and 65.11 average reasoning. Relative to Alpaca, perplexity is 1.96 lower and reasoning is 1.22 points higher. Dolly is weakest on both measures, at 36.19 and 59.78.

SFDA can select only patterns present in its candidate pool, so corpus choice matters. This comparison changes corpus size, data quality, response style, and task composition together.

	Dataset	Data size	Wikitext ↓	HellaSwag ↑	BoolQ ↑	PIQA ↑	ARC-e ↑	ARC-c ↑	Average Reasoning ↑
LLaMA 3.1-8B — P_r = 0.35	**LIMA**	1K	43.67	65.42	61.07	71.50	72.76	41.93	62.54
	Random	1K	43.44	65.12	62.44	71.98	73.10	40.49	62.63
	IFD	1K	44.03	63.64	62.22	72.36	73.57	42.67	62.89
	Nuggets	1K	42.44	66.32	61.96	72.57	72.14	42.50	63.09
	SelectIT	1K	41.09	65.71	62.79	71.87	74.39	42.33	63.42
	MONA	1K	35.01	66.71	58.90	68.91	70.35	39.08	60.79
	PASER	1K	33.55	66.05	60.33	71.87	73.33	42.10	62.74
	SFDA	**1K**	**32.18**	**67.11**	**61.92**	**72.69**	**74.50**	**43.27**	**63.89**
	Alpagasus	9K	36.22	65.00	62.87	72.63	76.69	43.45	64.13
	Random	9K	36.97	67.41	63.21	72.25	75.72	43.10	64.34
	IFD	9K	40.11	65.86	62.41	72.43	74.50	43.19	63.68
	Nuggets	9K	38.48	68.76	63.51	71.98	77.40	43.19	64.97
	SelectIT	9K	39.77	65.37	62.07	73.00	75.87	40.11	63.28
	MONA	9K	33.58	65.30	62.10	70.35	74.40	41.10	62.65
	PASER	9K	33.50	66.50	63.91	72.91	77.05	42.30	64.53
	SFDA	**9K**	**33.57**	**67.66**	**64.90**	**74.33**	**76.11**	**42.26**	**65.05**

Table 4: Matched-budget recovery on LLaMA 3.1 8B after LLM-Pruner at $P_r = 0.35$. The 1K block compares the original LIMA subset with 1K examples selected by each method; the 9K block does the same for Alpagasus. WikiText perplexity is lower-better; the five task accuracies and their unweighted average are higher-better. MONA is included at both budgets; its average is computed from the five reported task accuracies.

Candidate corpus	WikiText ↓	Reasoning ↑
SFDA–Alpaca	32.18	63.89
SFDA–LaMini	30.22	65.11
SFDA–Dolly	36.19	59.78

Table 5: One-thousand-example SFDA subsets from three candidate corpora. Lower WikiText perplexity and higher average reasoning are better.

7.6 Effect of TopK Selection Method: Magnitude vs. Gradient Magnitude

Figure 7 varies only the coordinate-selection rule applied before sparse factorization. Gradient magnitude outperforms activation magnitude on all five reasoning tasks, with displayed differences of approximately two to five points. The variants share the factorization, distribution objective, subset size, and recovery procedure, which isolates the coordinate rule. Response-loss gradients find coordinates that affect the current prediction; activation magnitude finds coordinates that are strongly expressed. The consistent ordering favors gradient saliency in this experiment. Figure 7 reports point estimates. Appendix 7.19.3 reports the seed-crossed analysis.

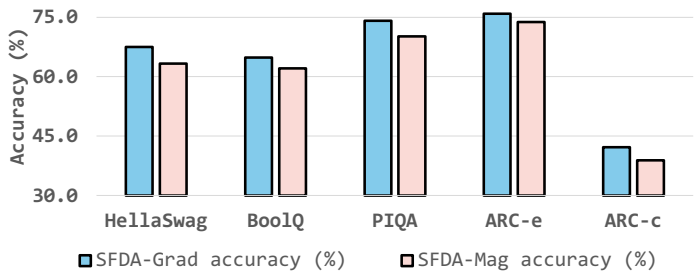

Figure 7: **TopK coordinate rule.** Gradient-selected (blue) and magnitude-selected (pink) SFDA variants on five reasoning tasks. Bars give accuracy (%, higher is better).

Compute breakdown. Table 10 reports the measured SFDA profile on 51,970 eligible Alpaca examples. End-to-end curation takes 768.5 seconds on one RTX 5090; response-descriptor extraction accounts for 683.1 seconds and is the dominant stage. All times come from the same hardware and implementation.

7.7 Sparse Target-State Coresets

The post-recovery objective in Equation 1 is the quantity of ultimate interest. SFDA's distribution-matching objective is only a pre-recovery surrogate. This appendix describes the conditions under which that surrogate can approximate the local recovery behavior of the complete candidate corpus.

The argument has three steps: (i) a recovery subset should reproduce the aggregate update of the candidate corpus; (ii) a target-state representation can act as a proxy for variation in those updates only when it retains recovery-relevant information; and (iii) SFDA controls only the part of that representation retained by its PCA projection and distribution discrepancy. The resulting statements are local sufficient conditions, not claims of global equivalence or causal sparse features.

7.8 Recovery Selection as Local Gradient Approximation

Let

$$L_{\mathcal{D}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell_i(\theta)$$

denote the candidate-corpus recovery loss and let $L_{\mathcal{S}}(\theta)$ be the corresponding uniformly weighted loss on a selected subset $\mathcal{S}$. At the pruned parameters θ_p, define

$$g_{\mathcal{D}} = \nabla L_{\mathcal{D}}(\theta_p), \qquad g_{\mathcal{S}} = \nabla L_{\mathcal{S}}(\theta_p), \qquad H_{\mathcal{D}} = \nabla^2 L_{\mathcal{D}}(\theta_p). \tag{13}$$

Consider one recovery step using the subset gradient, $\theta_p - \eta g_{\mathcal{S}}$. A second-order expansion of the full-corpus loss around θ_p gives

$$\begin{aligned} L_{\mathcal{D}}(\theta_p - \eta g_{\mathcal{S}}) = L_{\mathcal{D}}(\theta_p) - \eta\, g_{\mathcal{D}}^{\top} g_{\mathcal{S}} \\ + \frac{\eta^2}{2} g_{\mathcal{S}}^{\top} H_{\mathcal{D}} g_{\mathcal{S}} + \mathcal{O}\big(\eta^3 \|g_{\mathcal{S}}\|_2^3\big)\,. \end{aligned} \tag{14}$$

The first-order decrease is large when the subset update $g_{\mathcal{S}}$ agrees with the candidate-corpus update $g_{\mathcal{D}}$. This provides the first reason for set-level rather than pointwise selection: the relevant quantity is the *aggregate* update induced by the selected examples.

The curvature term does not change this interpretation into a global guarantee. Neural-network Hessians can be indefinite, and multi-step recovery can leave the neighborhood in which the expansion is accurate. We therefore state the comparison more explicitly.

Let

$$\delta_{\mathcal{S}} = g_{\mathcal{S}} - g_{\mathcal{D}}.$$

Assume that $L_{\mathcal{D}}$ has Hessian $H_{\mathcal{D}}$ at θ_p and that its Hessian is ρ-Lipschitz in a neighborhood containing the two one-step updates.

Proposition 1 (one-step recovery gap). For a step size η that keeps both updates inside this neighborhood,

$$\begin{aligned} &|L_{\mathcal{D}}(\theta_p - \eta g_{\mathcal{S}}) - L_{\mathcal{D}}(\theta_p - \eta g_{\mathcal{D}})| \\ &\le \eta \|g_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2 + \eta^2 \|H_{\mathcal{D}}\|_2 \|g_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2 \\ &\quad + \frac{\eta^2}{2}\|H_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2^2 + \frac{\rho\eta^3}{6}\left(\|g_{\mathcal{S}}\|_2^3 + \|g_{\mathcal{D}}\|_2^3\right). \end{aligned} \tag{15}$$

Reasoning. Apply the second-order Taylor expansion to both update directions and subtract. Substituting $g_{\mathcal{S}} = g_{\mathcal{D}} + \delta_{\mathcal{S}}$ leaves terms involving $g_{\mathcal{D}}^{\top}\delta_{\mathcal{S}}$, $g_{\mathcal{D}}^{\top} H_{\mathcal{D}} \delta_{\mathcal{S}}$, and $\delta_{\mathcal{S}}^{\top} H_{\mathcal{D}} \delta_{\mathcal{S}}$. Cauchy–Schwarz, the spectral-norm quadratic-form bound, and the Lipschitz-Hessian remainder yield Equation 41.

Thus, when the subset mean gradient is close to the corpus mean gradient, their one-step recovery losses are also close up to curvature and higher-order terms. The proposition does not claim that the full-corpus step is globally optimal or that this relation holds throughout a multi-epoch trajectory.

7.9 Mean Updates and Stochastic Update Dispersion

Matching only a deterministic mean update is not the entire story because recovery uses stochastic mini-batches. Let $g_i = \nabla_\theta \ell_i(\theta_p)$ and, for any nonempty set $\mathcal{A} \subseteq \mathcal{D}$, define

$$g_{\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} g_i.$$

For a uniformly sampled without-replacement mini-batch $\mathcal{B}_m \subseteq \mathcal{A}$ of size m, define

$$\widehat{g}_{\mathcal{A},m} = \frac{1}{m} \sum_{i \in \mathcal{B}_m} g_i$$

and let

$$G_{\mathcal{A}} = \mathrm{Cov}_{i \in \mathcal{A}}(g_i),$$

where the population covariance uses normalization $|\mathcal{A}|^{-1}$. Then

$$\mathbb{E}[\widehat{g}_{\mathcal{A},m}] = g_{\mathcal{A}}, \qquad \mathrm{Cov}(\widehat{g}_{\mathcal{A},m}) = \frac{|\mathcal{A}| - m}{m(|\mathcal{A}| - 1)} G_{\mathcal{A}}. \tag{16}$$

Hence, reproducing $g_{\mathcal{D}}$ approximates the expected stochastic update, whereas reproducing $G_{\mathcal{D}}$ approximates its dispersion. SFDA never observes the full gradient covariance during selection. Its sparse-feature statistics are therefore only a representation-level proxy for these quantities, under the assumptions developed below.

7.10 Why Measure the Pruned Target State?

The gradients relevant to recovery are evaluated at θ_p. Pruning changes both activations and the Jacobians through which activation changes affect the loss, so a representation obtained from the original model or an external encoder has no general guarantee of preserving $g_{\mathcal{D}}$.

SFDA instead constructs its representation from the pruned checkpoint. Response-loss gradients are used to identify where the representation should be measured. If h_i denotes a hidden descriptor and Δh is a small perturbation, then

$$\ell_i(h_i + \Delta h) - \ell_i(h_i) = \nabla_{h_i} \ell_i(h_i)^\top \Delta h + \mathcal{O}(\|\Delta h\|_2^2). \tag{17}$$

The derivative magnitude is therefore a local sensitivity signal for the supervised response loss. SFDA averages this signal over a candidate-only reference set and uses one shared coordinate set for every example. Importantly, this procedure determines *where to measure variation*; it does not use the scalar loss as an example utility score and does not make gradient saliency a causal attribution.

7.11 Sparse Features as a Coverage Coordinate System

Let h_i denote the standardized response-salient descriptor supplied to the TopK SAE. The SAE produces a sparse code $z_i \in \mathbb{R}^{d_z}$ and reconstruction

$$\widehat{h}_i = b_D + D z_i, \qquad e_i = h_i - \widehat{h}_i. \tag{18}$$

Because the encoder retains at most k_z active coordinates,

$$\widehat{h}_i = b_D + \sum_{j \in \mathrm{supp}(z_i)} z_{i,j} D_{:,j}, \qquad |\,\mathrm{supp}(z_i)| \le k_z. \tag{19}$$

The sparse support makes feature occurrence and co-activation observable. For a subset $\mathcal{S}$, define the count and exposure of latent j as

$$c_j(\mathcal{S}) = \sum_{i \in \mathcal{S}} \mathbf{1}[z_{i,j} > 0], \qquad e_j(\mathcal{S}) = \frac{c_j(\mathcal{S})}{|\mathcal{S}|}. \tag{20}$$

This does not require latents to correspond to uniquely identifiable semantic concepts. Latents may be permuted, split, merged, or redundant while providing a useful operational coordinate system. The role of sparsity in SFDA is to make support and co-activation explicit, not to assert monosemanticity.

7.12 A Local Bridge from Sparse Codes to Recovery Gradients

We next state the condition needed to connect sparse-code statistics to recovery updates.

Suppose the part of the per-example recovery gradient predictable from the descriptor can be represented as

$$g_i = q(h_i) + \xi_i, \tag{21}$$

where ξ_i denotes information about g_i not captured by the descriptor.

Assume that q is L_q-Lipschitz and has a β_q-Lipschitz Jacobian in a neighborhood containing h_i, $\widehat{h}_i$, and the decoder anchor b_D. Using $\widehat{h}_i = b_D + Dz_i$, a first-order expansion of q around b_D gives

$$g_i = b + Mz_i + \varepsilon_i, \qquad b = q(b_D), \qquad M = J_q(b_D)D, \tag{22}$$

with residual bounded by

$$\|\varepsilon_i\|_2 \le \|\xi_i\|_2 + L_q\|e_i\|_2 + \frac{\beta_q}{2}\|Dz_i\|_2^2. \tag{23}$$

Reasoning. Insert and subtract $q(\widehat{h}_i)$:

$$q(h_i) = q(\widehat{h}_i) + \big(q(h_i) - q(\widehat{h}_i)\big).$$

The Lipschitz condition bounds the second term by the SAE reconstruction error. Expanding $q(b_D+Dz_i)$ around b_D gives the affine component $q(b_D)+J_q(b_D)Dz_i$, while the Lipschitz-Jacobian assumption bounds the nonlinear remainder. Combining that remainder with ξ_i yields Equation 23.

This derivation explains the assumptions behind the affine bridge. Good SAE reconstruction controls one contribution to the residual, but it cannot guarantee gradient prediction: the descriptor itself may omit useful information, and the descriptor-to-gradient map may be substantially nonlinear.

7.13 From Code Statistics to Recovery Statistics

For any set $\mathcal{A}$, define

$$\mu_{\mathcal{A}}^z = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} z_i, \tag{24}$$

$$\Sigma_{\mathcal{A}}^z = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} (z_i - \mu_{\mathcal{A}}^z)(z_i - \mu_{\mathcal{A}}^z)^\top, \tag{25}$$

$$\bar{\varepsilon}_{\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \varepsilon_i, \tag{26}$$

$$\Omega_{\mathcal{A}} = \mathrm{Cov}_{i\in\mathcal{A}}(\varepsilon_i), \tag{27}$$

$$C_{z\varepsilon,\mathcal{A}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} (z_i - \mu_{\mathcal{A}}^z)(\varepsilon_i - \bar{\varepsilon}_{\mathcal{A}})^\top. \tag{28}$$

Averaging Equation 22 gives

$$g_{\mathcal{A}} = b + M\mu_{\mathcal{A}}^z + \bar{\varepsilon}_{\mathcal{A}}.$$

Therefore,

$$\|g_{\mathcal{S}} - g_{\mathcal{D}}\|_2 \le \|M\|_2\|\mu_{\mathcal{S}}^z - \mu_{\mathcal{D}}^z\|_2 + \|\bar{\varepsilon}_{\mathcal{S}} - \bar{\varepsilon}_{\mathcal{D}}\|_2. \tag{29}$$

Thus, matching code means controls the average recovery update only when the residual means are also sufficiently similar.

Centering Equation 22 yields

$$G_{\mathcal{A}} = M\Sigma_{\mathcal{A}}^z M^\top + \Omega_{\mathcal{A}} + MC_{z\varepsilon,\mathcal{A}} + C_{z\varepsilon,\mathcal{A}}^\top M^\top. \tag{30}$$

Consequently,

$$\begin{aligned}\|G_{\mathcal{S}} - G_{\mathcal{D}}\|_F \le{}& \|M\|_2^2\|\Sigma_{\mathcal{S}}^z - \Sigma_{\mathcal{D}}^z\|_F \\ &+ \|\Omega_{\mathcal{S}} - \Omega_{\mathcal{D}}\|_F \\ &+ 2\|M\|_2\|C_{z\varepsilon,\mathcal{S}} - C_{z\varepsilon,\mathcal{D}}\|_F.\end{aligned} \tag{31}$$

Code covariance can therefore approximate stochastic-update dispersion only when residual covariance and code–residual dependence remain sufficiently stable.

7.14 Effect of the PCA Projection

SFDA does not optimize the full code z_i directly. Let $m_{\mathcal{D}}^z$ and the candidate-fitted orthonormal projector P_q be those defined by the method. The retained coordinates are

$$r_i = P_q^\top (z_i - m_{\mathcal{D}}^z),$$

and the discarded component is

$$d_i = (I_{d_z} - P_q P_q^\top)(z_i - m_{\mathcal{D}}^z).$$

Hence,

$$z_i = m_{\mathcal{D}}^z + P_q r_i + d_i.$$

Substituting this decomposition into Equation 22 gives

$$g_i \approx M_q r_i + b_q + \xi_i', \qquad M_q = MP_q, \qquad b_q = Mm_{\mathcal{D}}^z + b, \qquad \xi_i' = Md_i + \varepsilon_i. \tag{32}$$

Let $\mu_r = \mathbb{E}[r_i]$, $\mu_\xi = \mathbb{E}[\xi_i']$, $\Sigma_r = \mathrm{Cov}(r_i)$, $\Sigma_\xi = \mathrm{Cov}(\xi_i')$, and $\Sigma_{r\xi} = \mathrm{Cov}(r_i, \xi_i')$. Then

$$\begin{aligned} \mathbb{E}[g_i] &\approx M_q \mu_r + b_q + \mu_\xi, \\ \mathrm{Cov}(g_i) &\approx M_q \Sigma_r M_q^\top + M_q \Sigma_{r\xi} + \Sigma_{r\xi}^\top M_q^\top + \Sigma_\xi. \end{aligned} \tag{33}$$

SFDA directly targets differences in the retained projected statistics, particularly $\Delta\mu_r$ and $\Delta\Sigma_r$. It does *not* directly control variation in d_i, the representation residual ε_i, or their dependence on r_i. PCA is therefore a computational approximation rather than a sufficiency claim.

7.15 What the SFDA Discrepancy Controls

We now connect the implemented Bhattacharyya term to the retained mean mismatch.

Let $\Sigma_{\mathcal{D}}^{r,\epsilon}$ and $\Sigma_{\mathcal{S}}^{r,\epsilon}$ denote the regularized projected covariances and define

$$\bar{\Sigma} = \frac{1}{2}\left(\Sigma_{\mathcal{D}}^{r,\epsilon} + \Sigma_{\mathcal{S}}^{r,\epsilon}\right), \qquad d_\mu = P_q^\top \left(\mu_{\mathcal{S}}^z - \mu_{\mathcal{D}}^z\right). \tag{34}$$

The Gaussian Bhattacharyya distance is

$$D_B = \frac{1}{8} d_\mu^\top \bar{\Sigma}^{-1} d_\mu + \frac{1}{2} \log \frac{\det \bar{\Sigma}}{\sqrt{\det \Sigma_{\mathcal{D}}^{r,\epsilon} \det \Sigma_{\mathcal{S}}^{r,\epsilon}}}. \tag{35}$$

The log-determinant term is nonnegative, which implies

$$\|d_\mu\|_2 \le \sqrt{8 \lambda_{\max}(\bar{\Sigma}) D_B}. \tag{36}$$

Since

$$\Delta_{\mathrm{SFDA}}(\mathcal{S}) = w_B D_B + w_{\mathrm{KS}} D_{\mathrm{KS}}$$

and both discrepancy terms are nonnegative,

$$D_B \le \frac{\Delta_{\mathrm{SFDA}}(\mathcal{S})}{w_B}.$$

Let $P_\perp$ span the orthogonal complement of P_q. Combining the projected-mean bound with Equation 29 yields

$$\begin{aligned} \|g_{\mathcal{S}} - g_{\mathcal{D}}\|_2 \le \|MP_q\|_2 &\sqrt{\frac{8 \lambda_{\max}(\bar{\Sigma}) \Delta_{\mathrm{SFDA}}(\mathcal{S})}{w_B}} \\ &+ \|MP_\perp\|_2 \left\|P_\perp^\top (\mu_{\mathcal{S}}^z - \mu_{\mathcal{D}}^z)\right\|_2 \\ &+ \|\bar{\varepsilon}_{\mathcal{S}} - \bar{\varepsilon}_{\mathcal{D}}\|_2 . \end{aligned} \tag{37}$$

Equation 37 shows exactly what the implemented objective can and cannot guarantee. A small SFDA discrepancy constrains the mean mismatch in the *retained PCA subspace*, subject to covariance conditioning. Two quantities remain uncontrolled: (i) code variation discarded by PCA and (ii) representation-to-gradient residual variation. With $q = 16$, SFDA therefore assumes that the retained directions contain enough recovery-relevant variation for the approximation to be useful.

The covariance part of the Bhattacharyya distance similarly penalizes differences between the regularized projected covariances. However, equal first and second moments do not identify a general non-Gaussian sparse distribution: distinct mixtures or support patterns may share identical moments.

The projected KS term provides a complementary check along the fixed first principal direction. For the first candidate-fitted principal coordinate,

$$D_{\mathrm{KS}} = \sup_a \left| \widehat{F}_{\mathcal{D},1}(a) - \widehat{F}_{\mathcal{S},1}(a) \right|. \tag{38}$$

It can detect marginal skew, tail, or multimodal differences that are not determined by the Gaussian moment summary. It does not constrain every sparse latent, every projected marginal, or the full multivariate distribution. PC1 is used because it is fixed before selection and captures maximum candidate-code variance, not because it is known to be a uniquely recovery-optimal direction.

7.16 Budget, Rare Features, and Set-Level Coverage

A fixed budget creates a simple coverage problem. Suppose latent j is active in n_j of the N candidate examples. A uniformly sampled subset of size B omits that latent with probability

$$\Pr[c_j(\mathcal{S}) = 0] = \frac{\binom{N-n_j}{B}}{\binom{N}{B}}. \tag{39}$$

The omission probability is largest for low-frequency latents and becomes more severe as B decreases. This explains why a small recovery set can easily lose less-common patterns.

Marginal exposure alone is nevertheless insufficient. Two subsets can contain the same features with similar frequencies while exhibiting very different activation magnitudes or co-activation structure. SFDA therefore operates on continuous projected codes rather than directly optimizing Equation 20.

This also clarifies the interpretation of the coverage diagnostics in the main paper. Broader sparse-feature exposure is evidence that a subset spans more of the candidate representation, but SFDA does not guarantee retention of every rare latent. In particular, PCA may discard a rare but recovery-relevant low-variance direction.

7.17 Scope of the Theoretical Argument

The complete surrogate chain can now be summarized as

small projected code discrepancy $\implies$ small retained-subspace update mismatch $\implies$ small local one-step recovery gap,

but only when the uncontrolled terms above remain small.

More precisely, the interpretation requires the following conditions:

1. The pruned-state, response-salient descriptor must retain information relevant to local recovery gradients.
2. The SAE must reconstruct that descriptor sufficiently well, and the descriptor-to-gradient relationship must be locally smooth enough that the affine approximation in Equation 22 has a small residual.
3. The candidate-fitted PCA subspace must retain the code variation relevant to recovery, while the discarded-subspace mismatch remains small.
4. Differences in representation residuals, their covariances, and their dependence on the retained code must remain small between $\mathcal{S}$ and $\mathcal{D}$.
5. The recovery trajectory must remain sufficiently local for the one-step analysis in Equation 41 to remain informative.

These conditions intentionally limit the theoretical claim. The analysis does not establish that individual SAE latents are causal repair units, that sparse codes globally determine recovery gradients, that SFDA finds the globally optimal subset, or that matching its discrepancy is equivalent to minimizing downstream recovery risk.

Indeed, the direct gradient-proxy control finds only weak global affine predictability. We therefore use the theory to specify the conditions under which target-state distribution matching can approximate local recovery updates, while judging the effectiveness of SFDA itself through the empirical recovery experiments and controlled ablations.

7.18 Theory of Sparse Target-State Coresets

Code-distribution matching is a surrogate, not the final recovery objective. Its justification has three local conditions. The subset gradient must be close to the corpus gradient, the SAE code must retain gradient-relevant variation, and the implemented discrepancy must control the relevant code mismatch. None of these conditions requires identifiable SAE concepts or makes code matching equivalent to downstream risk minimization.

7.18.1 Recovery Selection as Local Gradient Approximation

Let $g_i = \nabla_\theta \ell_i(\theta_p)$ be the recovery gradient of example i at the pruned parameters θ_p. For any nonempty $\mathcal{A} \subseteq \mathcal{D}$, define

$$g_{\mathcal{A}} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} g_i. \tag{40}$$

Locally, the desired coreset reproduces $g_{\mathcal{D}}$ using B uniformly weighted examples. It need not reproduce every g_i. The mean is the quantity that enters the first-order term in Equation 14.

Let $\delta_{\mathcal{S}} = g_{\mathcal{S}} - g_{\mathcal{D}}$. Assume that $L_{\mathcal{D}}$ has Hessian $H_{\mathcal{D}}$ at θ_p and that its Hessian is ρ-Lipschitz in a neighborhood containing both one-step updates. A Taylor expansion around θ_p gives the following bound.

Proposition 1 (One-step recovery gap)**.** *For a step size η that keeps both updates in the stated neighborhood,*

$$\begin{aligned} &|L_{\mathcal{D}}(\theta_p - \eta g_{\mathcal{S}}) - L_{\mathcal{D}}(\theta_p - \eta g_{\mathcal{D}})| \\ &\quad \le \eta \|g_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2 + \eta^2 \|H_{\mathcal{D}}\|_2 \|g_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2 + \frac{\eta^2}{2} \|H_{\mathcal{D}}\|_2 \|\delta_{\mathcal{S}}\|_2^2 \\ &\qquad + \frac{\rho \eta^3}{6} \left(\|g_{\mathcal{S}}\|_2^3 + \|g_{\mathcal{D}}\|_2^3 \right). \end{aligned} \tag{41}$$

Proof. Apply the second-order Taylor expansion to the two update directions and subtract. Substituting $g_{\mathcal{S}} = g_{\mathcal{D}} + \delta_{\mathcal{S}}$ leaves the terms $-\eta g_{\mathcal{D}}^\top \delta_{\mathcal{S}}$, $\eta^2 g_{\mathcal{D}}^\top H_{\mathcal{D}} \delta_{\mathcal{S}}$, and $\eta^2 \delta_{\mathcal{S}}^\top H_{\mathcal{D}} \delta_{\mathcal{S}}/2$. Cauchy–Schwarz, the spectral-norm quadratic-form bound, and the Lipschitz-Hessian remainder bound yield Equation 41. □

The proposition has a narrow interpretation. If the subset mean gradient is close to the corpus mean, their one-step losses are close up to curvature and third-order terms. It neither makes the full-corpus step globally optimal nor describes a multi-epoch trajectory after the parameters leave the local neighborhood.

7.18.2 Why Target-State, Response-Salient Features?

Why build the representation at the pruned checkpoint? Pruning changes the function, its activations, and the Jacobian that maps activation changes to parameter updates. The relevant gradients in Equation 40 are evaluated at θ_p, not at the unpruned parameters and not in an external encoder. A model-agnostic representation $\phi(x_i, y_i)$ could still be sufficient in a particular problem, but semantic proximity alone supplies no general bound on $\|g_{\mathcal{S}} - g_{\mathcal{D}}\|_2$. SFDA instead builds its descriptor from the same pruned state at which the local recovery gradients are defined.

Response saliency narrows the descriptor to coordinates locally coupled to the supervised response. If h_i denotes a hidden descriptor and Δh is a small activation perturbation, then

$$\ell_i(h_i + \Delta h) - \ell_i(h_i) = \nabla_{h_i}\ell_i(h_i)^\top \Delta h + \mathcal{O}(\|\Delta h\|_2^2). \tag{42}$$

The derivative magnitude gives a first-order sensitivity measure. We average it over a candidate-only reference sample and reuse the same coordinates for every example. This tells SFDA where to measure variation; it does not turn scalar loss into a utility score or saliency into a causal attribution.

7.18.3 Why Use a Sparse Autoencoder?

The SAE has two jobs. It must reconstruct the response-salient descriptor, and it must express that descriptor with a small set of active additive directions. Reconstruction measures fidelity; TopK supplies the support structure. Neither property alone is enough.

A local derivation of the code-to-gradient bridge. Let h_i denote the standardized response-salient descriptor used by the SAE, and let

$$\widehat{h}_i = b_D + Dz_i, \qquad e_i = h_i - \widehat{h}_i \tag{43}$$

be its reconstruction and error. The next proposition gives conditions under which the affine bridge in Equation 22 follows from reconstruction and a local linearization.

Proposition 2 (SAE-induced local gradient model)**.** *Suppose the component of g_i predictable from the descriptor can be written as*

$$g_i = q(h_i) + \xi_i, \tag{44}$$

where ξ_i is descriptor-insufficiency error. Assume that q is L_q-Lipschitz and has a β_q-Lipschitz Jacobian on a neighborhood containing h_i, $\widehat{h}_i$, and the decoder anchor b_D. Then

$$g_i = b + Mz_i + \varepsilon_i, \qquad b = q(b_D), \qquad M = J_q(b_D)D, \tag{45}$$

with residual bounded by

$$\|\varepsilon_i\|_2 \le \|\xi_i\|_2 + L_q\|e_i\|_2 + \frac{\beta_q}{2}\|Dz_i\|_2^2. \tag{46}$$

Proof. Insert and subtract $q(\widehat{h}_i)$. Lipschitz continuity bounds $\|q(h_i) - q(\widehat{h}_i)\|_2$ by $L_q\|e_i\|_2$. A first-order expansion of $q(b_D + Dz_i)$ around b_D, together with the Lipschitz-Jacobian remainder, gives $q(b_D) + J_q(b_D)Dz_i$ plus a term bounded by $\beta_q\|Dz_i\|_2^2/2$. Combining these terms with ξ_i proves the claim. □

The decoder turns sparse coordinates into additive activation directions, and the local Jacobian maps those directions into additive gradient directions. Reconstruction error controls one part of the residual:

$$\mathbb{E}\|q(h_i) - q(\widehat{h}_i)\|_2 \le L_q\sqrt{\mathbb{E}\|e_i\|_2^2}, \tag{47}$$

After normalization by descriptor variance, held-out FVU measures one aspect of representation fidelity. It does not guarantee recovery because the descriptor may omit useful information or relate nonlinearly to the gradient.

What sparsity contributes. For a TopK code with at most k_z active coordinates,

$$\widehat{h}_i = b_D + \sum_{j \in \mathrm{supp}(z_i)} z_{i,j} D_{:,j}, \qquad |\,\mathrm{supp}(z_i)| \le k_z. \tag{48}$$

Each example exposes at most k_z decoder directions, and each outer product $z_i z_i^\top$ has at most k_z^2 nonzero entries before aggregation. This makes support and co-activation explicit. A dense autoencoder can reconstruct the descriptor but usually lacks this support-level notion of coverage. The matched dense-autoencoder control tests the difference; the equation alone does not imply that sparsity improves recovery.

TopK fixes an exact support budget per example. If u_i is the pre-TopK vector and $\gamma_i = u_{i,(k_z)} - u_{i,(k_z+1)} > 0$ is the boundary gap between its sorted entries, any perturbation satisfying $\|\Delta u_i\|_\infty <$

$\gamma_i/2$ preserves the selected support. The support is stable when this margin is large and unstable near ties, which motivates the sensitivity analysis for k_z.

Sparsity does not make the factorization identifiable. Latents may be permuted, redundant, split, or merged while retaining similar reconstruction quality. Unit-norm decoder columns constrain scale but do not establish monosemanticity. SFDA needs a stable operational coordinate system, not a unique semantic interpretation of every latent.

7.18.4 From Code Statistics to Recovery Statistics

For a set $\mathcal{A}$, define the empirical code mean, covariance, residual mean, residual covariance, and code–residual cross-covariance as

$$\begin{aligned} \mu^z_{\mathcal{A}} &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} z_i, & \Sigma^z_{\mathcal{A}} &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} (z_i-\mu^z_{\mathcal{A}})(z_i-\mu^z_{\mathcal{A}})^\top, \\ \bar{\varepsilon}_{\mathcal{A}} &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \varepsilon_i, & \Omega_{\mathcal{A}} &= \mathrm{Cov}_{\mathcal{A}}(\varepsilon_i), \\ C_{z\varepsilon,\mathcal{A}} &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} (z_i-\mu^z_{\mathcal{A}})(\varepsilon_i-\bar{\varepsilon}_{\mathcal{A}})^\top. \end{aligned} \tag{49}$$

Proposition 3 (Transfer of code-moment error)**.** *Under the affine decomposition $g_i = b + Mz_i + \varepsilon_i$, the subset-gradient error obeys*

$$\|g_{\mathcal{S}} - g_{\mathcal{D}}\|_2 \le \|M\|_2\|\mu^z_{\mathcal{S}} - \mu^z_{\mathcal{D}}\|_2 + \|\bar{\varepsilon}_{\mathcal{S}} - \bar{\varepsilon}_{\mathcal{D}}\|_2. \tag{50}$$

Moreover, if $G_{\mathcal{A}} = \mathrm{Cov}_{\mathcal{A}}(g_i)$, then

$$\begin{aligned} \|G_{\mathcal{S}} - G_{\mathcal{D}}\|_F \le \|M\|_2^2\|\Sigma^z_{\mathcal{S}} - \Sigma^z_{\mathcal{D}}\|_F + \|\Omega_{\mathcal{S}} - \Omega_{\mathcal{D}}\|_F \\ + 2\|M\|_2\|C_{z\varepsilon,\mathcal{S}} - C_{z\varepsilon,\mathcal{D}}\|_F. \end{aligned} \tag{51}$$

Proof. Averaging $g_i = b + Mz_i + \varepsilon_i$ over each set gives $g_{\mathcal{A}} = b + M\mu^z_{\mathcal{A}} + \bar{\varepsilon}_{\mathcal{A}}$, from which Equation 50 follows by the triangle inequality. Centering the same decomposition gives

$$G_{\mathcal{A}} = M\Sigma^z_{\mathcal{A}}M^\top + \Omega_{\mathcal{A}} + MC_{z\varepsilon,\mathcal{A}} + C^\top_{z\varepsilon,\mathcal{A}}M^\top. \tag{52}$$

Subtracting the identities for $\mathcal{S}$ and $\mathcal{D}$ and applying spectral/Frobenius norm inequalities yields Equation 51. □

The bounds make the missing assumptions clear. Code-mean matching controls the average update only if the residual means also match. Code covariance controls update dispersion only if residual covariance and code and residual dependence remain stable. Table 9 finds weak global affine predictability for the controlled checkpoint. It does not reconstruct the selected-subset residual statistics, so residual stability remains an untested condition.

Combining Propositions 3 and 1 gives the main conditional guarantee. Define

$$q_{\mathcal{S}} = \|M\|_2\|\mu^z_{\mathcal{S}} - \mu^z_{\mathcal{D}}\|_2 + \|\bar{\varepsilon}_{\mathcal{S}} - \bar{\varepsilon}_{\mathcal{D}}\|_2, \tag{53}$$

Then Equation 41 holds with $\|\delta_{\mathcal{S}}\|_2$ replaced by $q_{\mathcal{S}}$. The link from code means to recovery passes through M and the residual mean; it does not treat each latent as a causal repair unit.

7.18.5 What the SFDA Discrepancy Controls

What does the implemented objective control? Its Gaussian term operates in a $q = 16$ projection fitted only on the candidate corpus. Let $P_q \in \mathbb{R}^{d_z\times q}$ have orthonormal columns, let $\mu^z_{\mathcal{D}}$ be the candidate-code mean, and define $r_i = P_q^\top(z_i - \mu^z_{\mathcal{D}})$. Denote the regularized projected covariances by $\Sigma^{r,\epsilon}_{\mathcal{D}}$ and $\Sigma^{r,\epsilon}_{\mathcal{S}}$, and define

$$\overline{\Sigma} = \frac{1}{2}\left(\Sigma^{r,\epsilon}_{\mathcal{D}} + \Sigma^{r,\epsilon}_{\mathcal{S}}\right), \qquad d_\mu = P_q^\top(\mu^z_{\mathcal{S}} - \mu^z_{\mathcal{D}}). \tag{54}$$

The Gaussian Bhattacharyya distance is

$$D_B = \frac{1}{8} d_\mu^\top \overline{\Sigma}^{-1} d_\mu + \frac{1}{2} \log \frac{\det \overline{\Sigma}}{\sqrt{\det \Sigma_{\mathcal{D}}^{r,\epsilon} \det \Sigma_{\mathcal{S}}^{r,\epsilon}}}. \tag{55}$$

The log-determinant term is nonnegative by concavity of $\log \det$, so

$$\|d_\mu\|_2 \le \sqrt{8 \lambda_{\max}(\overline{\Sigma}) D_B}. \tag{56}$$

Because both terms in Equation 11 are nonnegative and $w_B > 0$, $D_B \le \Delta_{\mathrm{SFDA}}/w_B$. If $P_\perp$ spans the orthogonal complement of P_q, Proposition 3 implies

$$\begin{aligned} \|g_{\mathcal{S}} - g_{\mathcal{D}}\|_2 \le \|M P_q\|_2 \sqrt{\frac{8 \lambda_{\max}(\overline{\Sigma}) \Delta_{\mathrm{SFDA}}(\mathcal{S})}{w_B}} \\ + \|M P_\perp\|_2 \|P_\perp^\top (\mu_{\mathcal{S}}^z - \mu_{\mathcal{D}}^z)\|_2 + \|\overline{\varepsilon}_{\mathcal{S}} - \overline{\varepsilon}_{\mathcal{D}}\|_2. \end{aligned} \tag{57}$$

Equation 57 contains two uncontrolled terms. Subject to covariance conditioning, a small SFDA discrepancy bounds the mean mismatch in the retained subspace, but not the discarded subspace or the affine residual. With $q = 16$, SFDA assumes that the retained directions contain the important recovery variation.

The covariance part of D_B is zero exactly when the two regularized covariances match. Under bounded eigenvalues, it behaves locally as a quadratic penalty on their whitened difference. Equal first and second moments still do not identify a non-Gaussian sparse distribution because different mixtures and support patterns can share them.

The projected KS term checks one part of this ambiguity. For the fixed candidate-fitted direction v_1, let

$$D_{\mathrm{KS}} = \sup_t \left| \widehat{F}_{\mathcal{D},v_1}(t) - \widehat{F}_{\mathcal{S},v_1}(t) \right|. \tag{58}$$

It bounds empirical mass mismatch for threshold sets along v_1 and can reveal skew, multimodality, or tail differences missed by the Gaussian summary on that direction. It does not cover every latent marginal or the full multivariate distribution.

We choose v_1 for computational and statistical reasons, not because it is known to be recovery-optimal. It is fixed before search, uses no labels, and maximizes candidate-code variance among unit directions. This gives a relatively stable one-dimensional CDF when B is small. Because variance need not track recovery relevance, PC1 KS remains a secondary diagnostic. Table 7 does not show that PC1 is uniquely best.

7.18.6 Budget, Coverage, and the Scope of the Guarantee

The budget $|\mathcal{S}| = B$ fixes the empirical gradient normalization and prevents the trivial solution of keeping the full corpus. Sparse support makes the allocation visible. If latent j occurs with probability p_j, an independent sample of size B omits it with probability $(1 - p_j)^B$; Equation 39 gives the corresponding finite-corpus expression. Rare latents disappear first as the budget shrinks. SFDA reduces a projected set-level discrepancy, but the projection can still miss a rare latent. Exposure and co-activation remain diagnostics rather than guarantees.

The resulting guarantee depends on four conditions:

1. The pruned-state, response-salient descriptor must retain information predictive of local recovery gradients.
2. The SAE must reconstruct that descriptor accurately enough, and the descriptor-to-gradient map must be sufficiently smooth for an affine approximation to have small residuals.
3. The retained projection and the SFDA discrepancy must preserve the code statistics that matter through M, while discarded-subspace and residual mismatches remain small.
4. The recovery trajectory must remain local enough that the one-step analysis continues to be informative.

When these conditions hold, the chain

$$\begin{aligned} \text{small code discrepancy} &\Longrightarrow \text{small local gradient mismatch} \\ &\Longrightarrow \text{small one-step recovery gap} \end{aligned} \tag{59}$$

is quantified by Equations 57 and 41. Outside these conditions, SFDA must be judged empirically. The analysis does not establish causal or monosemantic latents, global subset optimality, or gains on every task. We test the main assumptions with target-state and dense controls, an objective ablation, and a direct gradient probe.

7.19 Controlled Validation of the SFDA Design

This appendix reports source-backed controls for the representation, selection objective, recovery budget, gradient proxy, SAE capacity, and systems cost. All controlled runs use the repaired LLaMA 3 8B instruction checkpoint at $P_r = 0.35$ and Alpaca as the candidate corpus. The globally shared coordinate set has $K_h = 128$.

7.19.1 Representation control

Table 6 holds the coordinate set, selected-set size, distribution objective, swap budget, and recovery schedule fixed while changing only the representation supplied to PCA and subset search. The TopK SAE achieves the lowest reconstruction fraction of variance unexplained (FVU), 0.262 ± 0.006, compared with 0.300 ± 0.007 for the dense autoencoder. Changing the representation substantially alters subset composition: relative to the standard SFDA subset, the unpruned dense, pruned dense, and dense autoencoder variants attain Jaccard similarities of 0.24 ± 0.04, 0.26 ± 0.02, and 0.29 ± 0.03, respectively. The standard TopK SAE configuration has Jaccard 1.00 by definition because it provides the reference SFDA subset. Despite these differences in representation and subset composition, held-out recovery NLL remains comparable across variants: 1.759 ± 0.090 for TopK SAE, 1.755 ± 0.085 for pruned dense descriptors, and 1.762 ± 0.070 for the dense autoencoder. Thus, this control shows that the sparse representation provides a compact reconstruction space and materially changes which examples are selected, but does not by itself produce a measurable improvement in downstream recovery NLL.

Representation	FVU ↓	NLL ↓	NLL reduction ↑	Search (s) ↓	Jaccard ↑
Unpruned dense	N/A	1.757 ± 0.079	0.348 ± 0.020	1.03 ± 0.01	0.24 ± 0.04
Pruned dense	N/A	1.755 ± 0.085	0.350 ± 0.009	1.05 ± 0.00	0.26 ± 0.02
Pruned dense AE	0.300 ± 0.007	1.762 ± 0.070	0.343 ± 0.020	0.97 ± 0.01	0.29 ± 0.03
Pruned TopK SAE	$\mathbf{0.262 \pm 0.006}$	1.759 ± 0.090	0.346 ± 0.011	$\mathbf{0.98 \pm 0.02}$	1.00

Table 6: Representation control. Cells report mean $\pm$ sample standard deviation over three selection seeds and three recovery seeds where applicable. NLL is held-out response-token negative log-likelihood after recovery; NLL reduction is measured from the corresponding unrecovered checkpoint. Jaccard measures overlap with the subset selected by the standard SFDA configuration.

7.19.2 Objective control

Table 7 separates the distribution terms while holding the TopK codes and optimizer fixed. Each single-term objective most strongly reduces its intended diagnostic: mean matching lowers the mean gap to 0.010 ± 0.002, covariance matching lowers the covariance gap to 0.205 ± 0.021, and KS matching lowers the KS statistic to 0.013 ± 0.001. The combined SFDA objective reduces all three gaps relative to random selection. Its recovery NLL, 1.759 ± 0.090, lies within the cross-seed variation of every optimized alternative. The objective behaves as intended in representation space, but this control does not distinguish the alternatives by recovery NLL. It also does not show that covariance matching, PC1 KS, or their combination is necessary for the gain. We use the combined objective as the fixed benchmark specification, not as evidence of component synergy.

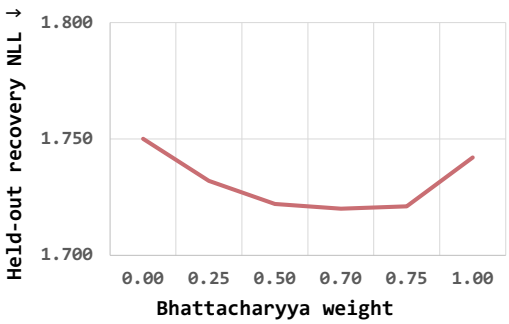

Figure 8: **Sensitivity to the Bhattacharyya–KS weighting.** Held-out recovery NLL as w_B varies with $w_{\mathrm{KS}} = 1 - w_B$. The curve connects the discrete tested weights. Intermediate mixtures have lower observed NLL than either single-term endpoint, and the default $w_B = 0.7$ lies in the broad low-NLL region.

Objective	Mean gap ↓	Cov. gap ↓	Rare exposure ↑	KS ↓	Search (s) ↓	NLL ↓
Random	0.261 ± 0.047	1.122 ± 0.066	0.545 ± 0.012	0.068 ± 0.019	0.00	1.761 ± 0.085
Mean	$\mathbf{0.010 \pm 0.002}$	0.811 ± 0.139	0.554 ± 0.011	0.051 ± 0.007	1.04 ± 0.03	1.760 ± 0.083
Covariance	0.087 ± 0.020	$\mathbf{0.205 \pm 0.021}$	0.557 ± 0.008	0.046 ± 0.003	1.03 ± 0.01	$\mathbf{1.753 \pm 0.072}$
Bhattacharyya	0.052 ± 0.011	0.219 ± 0.042	0.556 ± 0.007	0.038 ± 0.002	1.02 ± 0.03	1.766 ± 0.089
KS	0.182 ± 0.027	0.943 ± 0.150	0.546 ± 0.013	$\mathbf{0.013 \pm 0.001}$	0.98 ± 0.03	1.758 ± 0.079
SFDA	0.059 ± 0.013	0.304 ± 0.051	$\mathbf{0.557 \pm 0.005}$	0.017 ± 0.000	$\mathbf{0.98 \pm 0.02}$	1.759 ± 0.090

Table 7: Objective control on the same frozen TopK codes. Diagnostic gaps are measured between the selected subset and the complete 2,048-example validation pool. Rare exposure is the fraction of low-frequency active features represented by the subset.

Figure 8 examines sensitivity within the combined-objective family by varying the Bhattacharyya weight w_B and setting $w_{\mathrm{KS}} = 1 - w_B$. The two single-term endpoints have higher observed held-out recovery NLL than the intermediate mixtures, while the curve is comparatively flat from $w_B = 0.5$ to $w_B = 0.75$. The default $(w_B, w_{\mathrm{KS}}) = (0.7, 0.3)$ lies near the lowest observed value in this broad region.

This sweep supports using both distribution terms and indicates that the default is not sensitive to a small change in their relative weighting. It does not establish 0.7 as a unique optimum: the neighboring settings are close, and the plot does not report paired uncertainty for their differences. We therefore treat $(0.7, 0.3)$ as a fixed operating point selected from a stable region, not as evidence that this exact coefficient pair is necessary. This interpretation is consistent with Table 7, which does not distinguish the objective variants by recovery NLL.

7.19.3 Controlled recovery budgets

Table 8 crosses three selection seeds with three recovery seeds. Under a one-epoch schedule, SFDA improves held-out response NLL over random selection by 0.016 ± 0.017 and wins seven of nine paired runs. With an equal 10,000-response-token budget, the advantage is 0.009 ± 0.014, again with seven wins. The same win count under both schedules rules out unequal epoch length as the source of the ordering, although the mean NLL gap is small.

Recovery budget	Random NLL ↓	SFDA NLL ↓	Paired advantage ↑	Wins
One epoch	1.777 ± 0.086	1.760 ± 0.079	0.016 ± 0.017	$7/9$
10K response tokens	1.771 ± 0.078	1.762 ± 0.080	0.009 ± 0.014	$7/9$

Table 8: Seed-crossed recovery control. Means and sample standard deviations aggregate the same nine selection-seed/recovery-seed pairs for each schedule. Paired advantage is Random NLL minus SFDA NLL.

The seed-crossed NLL results test whether the advantage survives changes in selection and recovery seeds. Figure 9 provides a complementary task-level comparison against Full, Random, and PASER. It evaluates intermediate recovery checkpoints after the same number of consumed response tokens or the same number of optimizer updates.

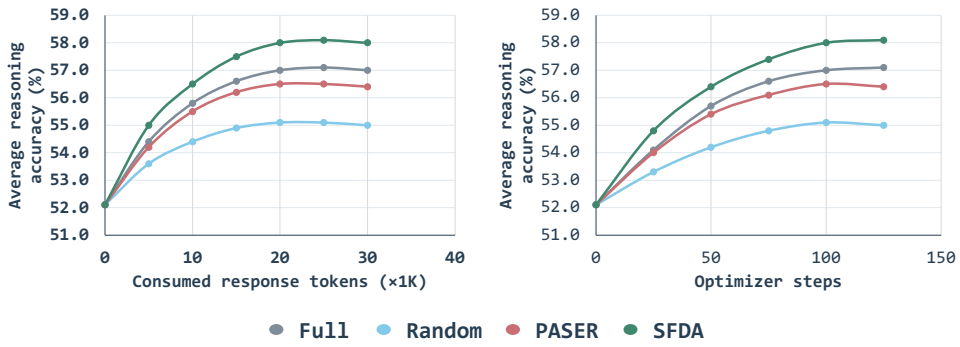

Figure 9: **Recovery under matched optimization budgets.** (a) Average reasoning accuracy as a function of consumed response tokens. (b) Average reasoning accuracy as a function of optimizer updates. SFDA remains ahead of Full, Random, and PASER under both controls. The comparison isolates data composition from differences in training exposure.

The two panels give the same ordering throughout most of recovery. At the final token-matched checkpoint, SFDA reaches 58.0% average reasoning accuracy, compared with 57.0% for Full. The separation also appears before the final checkpoint and remains when progress is indexed by optimizer updates. SFDA therefore does more than reproduce the behavior of full-corpus training with fewer examples. Within the matched budgets, its selected subset produces greater recovery per unit of training. This result supports the role of SFDA as an active subset optimizer, while distribution matching serves as the criterion used to construct that subset.

Figure 10 examines whether the examples selected by SFDA induce updates that are directionally consistent with the recovery signal present in the full candidate corpus. Higher cosine similarity indicates that an example's gradient agrees more strongly with the mean corpus gradient. The SFDA-selected examples show substantially greater alignment, with a mean cosine of 0.315, compared with 0.109 for rejected examples, 0.120 for Random, and 0.240 for PASER. This suggests that SFDA preferentially retains examples whose local recovery gradients better reflect the aggregate update direction of the candidate pool.

Figure 11 provides the complementary view of internal redundancy. A recovery subset can be well aligned with the corpus gradient while still wasting budget on many examples that induce nearly identical updates. We therefore measure the mean pairwise cosine between example gradients within each subset. SFDA obtains a mean pairwise cosine of 0.22, compared with 0.34 for PASER and 0.41 for Random, indicating that its selected examples cover a broader range of update directions.

Taken together, the two figures show that SFDA does not improve gradient alignment simply by repeatedly selecting examples from one dominant direction. Instead, the selected set is both more closely oriented toward the full-corpus recovery update and less internally redundant. This is the behavior expected from a set-level selection rule: the method favors examples that contribute to

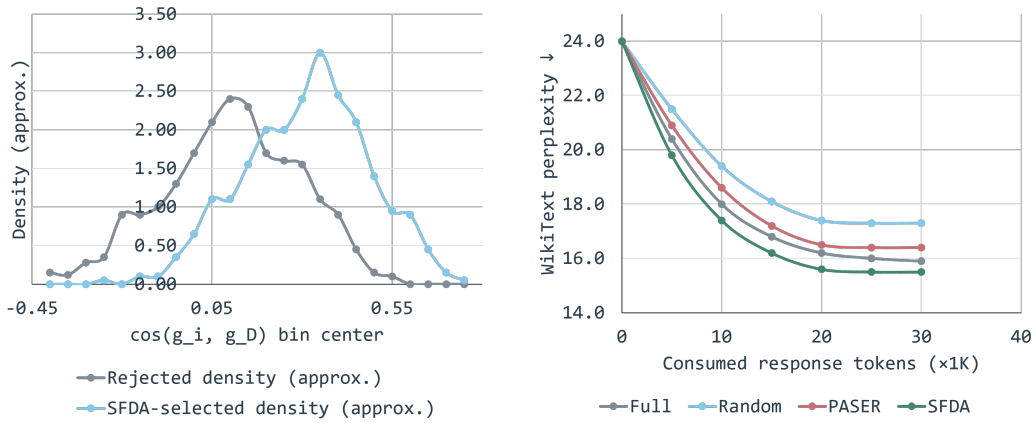

Figure 10: **Gradient alignment with the full-corpus update.** Distribution of $\cos(g_i, g_{\mathcal{D}})$ for SFDA-selected and rejected examples. The curves are smoothed summaries of the measured per-example cosine similarities.

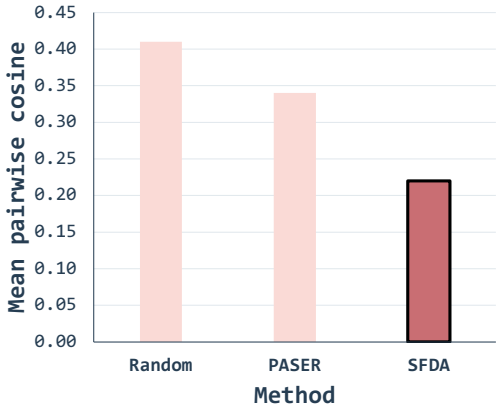

Figure 11: **Gradient redundancy within the selected subset.** Mean pairwise gradient cosine for Random, PASER, and SFDA subsets. Lower values indicate fewer repeated gradient directions within the selected set.

the desired recovery direction while avoiding unnecessary duplication within the limited selection budget.

These measurements provide an empirical connection between SFDA's selected subset and the gradient structure of recovery. They should nevertheless be interpreted as diagnostic evidence rather than a causal result. In particular, they do not establish that individual sparse features directly cause recovery, nor that the complete recovery gradient is globally determined by the projected sparse code. The next diagnostic evaluates that stronger relationship separately.

7.19.4 GRADIENT-PROXY DIAGNOSTIC

The local analysis in Section 3 uses an affine relation between descriptors and recovery gradients as a sufficient condition. We test that condition directly with a ridge probe. For 256 examples, the target is the 32,768-dimensional gradient of the last-layer query-projection LoRA-B matrix. The train/validation/test split is 70/15/15 for recovery seeds 11, 23, and 47. Ridge strength is selected from $\{1, 10, 100, 1000\}$ on validation data and refitted on train plus validation data. A fixed 64-dimensional Gaussian projection (seed 181) supplies the covariance diagnostic.

Table 9 shows weak linear predictability for all three representations. Pruned dense descriptors give the highest test R^2, but it is only 0.055 ± 0.002. The autoencoder representations are near zero or negative, and every cosine similarity is below 0.23. This rules out a strong global affine-gradient explanation for the checkpoint. The moment analysis remains a local design motivation, not an empirical account of the main gains. The probe also does not estimate the subset and corpus residual means, covariances, or code and residual cross-covariances required by Equation 33. Residual stability is still an untested assumption.

Representation	Mean-gradient error ↓	Projected-cov. error ↓	Test R^2 ↑	Cosine ↑	Normalized MSE ↓
Pruned dense	$\mathbf{0.719 \pm 0.033}$	0.983 ± 0.008	$\mathbf{0.055 \pm 0.002}$	$\mathbf{0.228 \pm 0.017}$	$\mathbf{0.945 \pm 0.002}$
Pruned dense AE	0.819 ± 0.051	$\mathbf{0.901 \pm 0.049}$	0.007 ± 0.038	0.171 ± 0.015	0.993 ± 0.038
Pruned TopK SAE	0.728 ± 0.071	0.960 ± 0.032	-0.050 ± 0.096	0.156 ± 0.024	1.050 ± 0.096

Table 9: Gradient-proxy diagnostic. Errors are relative Frobenius norms; normalized MSE divides test MSE by the variance around the training-gradient mean. Cells report mean ± sample standard deviation across the three recovery splits.

7.19.5 SAE CAPACITY

Figure 12 varies one capacity parameter at a time under the same selection and recovery protocol. For widths from 1,024 to 8,192, held-out NLL ranges from about 1.710 to 1.716. Across $k_z \in \{8, 16, 32, 64\}$, it ranges from 1.709 to 1.718. Recovery is insensitive to these capacity choices at this scale. The default $(d_z, k_z) = (4096, 32)$ is an interior operating point rather than a tuned extremum.

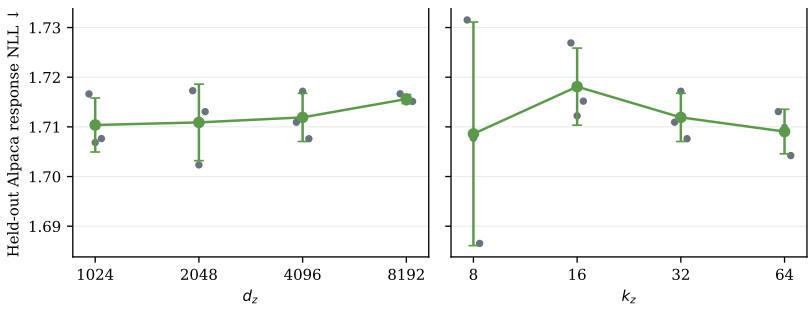

Figure 12: SAE-capacity control. Left: held-out response NLL as SAE width varies with $k_z = 32$. Right: held-out response NLL as TopK sparsity varies with width 4,096. Lower is better.

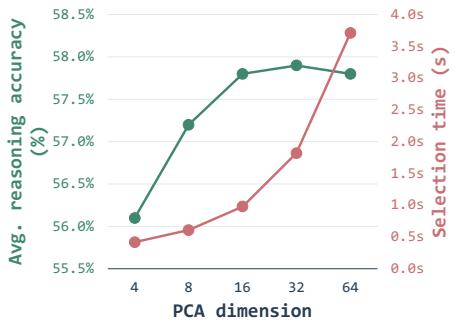

Figure 14: **Sensitivity to PCA dimension q.** Average reasoning accuracy changes only modestly from $q = 16$ to $q = 64$, whereas selection time increases with the projected dimension. The default is $q = 16$.

7.19.6 Descriptor and projection dimensions

Figure 13 first examines the response-salient descriptor width K_h. Increasing K_h from 32 to 128 improves average reasoning accuracy, after which the curve becomes approximately flat. At the same time, reconstruction FVU continues to decrease as more hidden coordinates are retained, while selection time increases. The default $K_h = 128$ therefore lies near the observed accuracy–efficiency knee rather than representing a uniquely optimal setting.

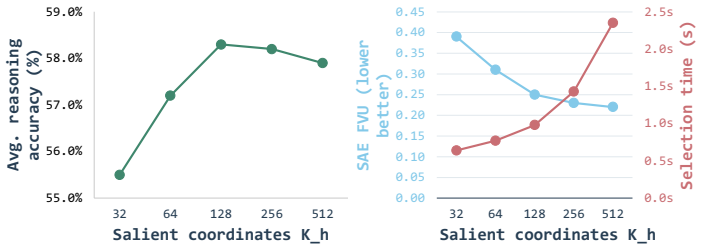

Figure 13: **Sensitivity to response-salient descriptor width K_h.** The green curve reports average reasoning accuracy, the blue curve reports SAE reconstruction FVU (lower is better), and the red curve reports selection time. Increasing K_h improves accuracy up to approximately $K_h = 128$, after which the performance gain becomes small while computational cost continues to increase. The default used elsewhere in the paper is $K_h = 128$.

Figure 14 examines the dimensionality of the candidate-fitted PCA representation used during subset search. Reasoning accuracy improves across the lower-dimensional settings and then varies only slightly from $q = 16$ to $q = 64$. In contrast, selection time continues to grow as the projected dimension increases. We therefore retain $q = 16$, which lies within this relatively stable performance region while keeping the full-covariance search inexpensive.

Together, these sweeps indicate that SFDA is not highly sensitive to either descriptor width or projection dimension once a moderate representation size is reached. We use the sweeps primarily to characterize this sensitivity and the associated computational trade-off. Because paired uncertainty estimates are not available for these auxiliary experiments, small differences between nearby settings should not be interpreted as statistically significant.

7.19.7 Measured systems profile

The near-full-corpus profile in Table 10 uses 51,970 eligible Alpaca examples, a 512-example saliency reference, a 5,197-item subset, and 100,000 swap proposals on one RTX 5090. End-to-end curation takes 768.5 seconds, with response-descriptor extraction accounting for 683.1 seconds (88.9%). Swap search takes 56.8 seconds, while SAE fitting and PCA together take 2.3 seconds.

Peak GPU allocation is 12.57 GiB and peak resident CPU memory is 5.09 GiB. Descriptor extraction is the dominant cost. All measurements use the same hardware and implementation.

Stage	Time (s)
Saliency coordinates	7.3
Response descriptors	683.1
TopK SAE fit	1.1
PCA and statistics	1.2
Fixed-size swap search	56.8
End-to-end curation	768.5

Table 10: Measured SFDA systems profile on one RTX 5090. End-to-end time includes model loading and tokenization in addition to the listed stages.