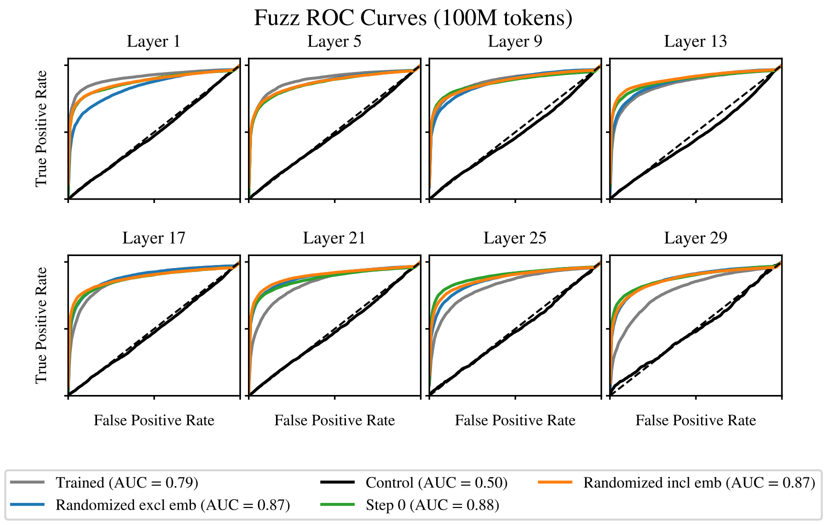

Figure 1: 'Fuzzing' ROC curve vs. layer for Pythia-6.9b (100 latents sampled per SAE). The trained model (gray line) and randomized variants (colored) overlap, whereas the control (black) is near chance (dotted). This suggests aggregate AUROC alone is insufficient to attribute latents to learned computation. See Figure 2 for other metrics/model sizes and Appendix E for multiple random seeds.

a given token does not have a consistent embedding vector. For this variant, we expect auto-interpretability to perform at the level of chance.

For our primary experiments, we trained SAEs on 100M tokens from the RedPajama dataset (Weber et al., 2024) using an activation buffer size of 10M tokens (see Appendix C for a subset of experiments that demonstrate similar results with SAEs trained on one billion tokens). For models with fewer than 410M parameters, we trained an SAE at every layer; for Pythia-1b, we trained SAEs at every second layer; and for Pythia-6.9b, we trained SAEs at every fourth layer.

Unless otherwise stated, we trained k-sparse autoencoders (also known as TopK SAEs; Makhzani and Frey 2014; Gao et al. 2024), with an expansion factor of $R = 64$ and sparsity $k = 32$. We confirm that our results are robust with respect to these hyperparameters by training SAEs on Pythia-160m with expansion factors equal to powers of 2 between 16 and 128, and sparsities of 16 and 32 (Figure 18). The training implementation is based on Belrose et al. (2025); our evaluations are based on Caden et al. (2025) and Karvonen et al. (2024a).

Auto-interpretability Feature explanations that identify a concept can be input to a classifier that predicts whether the concept appears in the text inputs. Such a classifier may be evaluted by traditional metrics, like the area under the receiver operating characteristic (ROC) curve (AUROC). Paulo et al. (2024) proposed 'fuzzing' and 'detection' classification tasks to evaluate feature explanations. For 'fuzzing' scoring, both positive and negative examples of tokens (i.e., with non-zero and zero activation values, respectively) for a given latent are delimited with special characters, and a language model is prompted to identify which examples have been correctly delimited for the latent given its explanation. For 'detection', a language model is asked to identify which examples contain activating tokens for each feature. Bills et al. (2023) originally proposed 'simulation' scoring, based on the correlation between predicted and observed activations, but this method is expensive to compute.

Except where noted, we report 'fuzzing' scores as a measure of auto-interpretability, because this measure has been demonstrated to correlate with simulation scoring (Paulo et al., 2024). We include similar AUROC curves for the 'detection' scoring method in Appendix B. For each trained SAE (i.e., underlying model, variant, and layer), we randomly sampled 100 features to obtain auto-interpretability scores. The implementation is based on Paulo et al. (2024). We use the `Meta-Llama-3.1-70B-Instruct-AWQ-INT4` model to generate explanations and make predictions (larger than the 8B models used by Choi et al. (2024) and open-source, unlike Bills et al. 2023).

We found that the auto-interpretability scores were far more similar between the trained and randomized models than with the control (Figures 1 and 2). The similarity between the ROC scores for trained and randomized transformers demonstrates that 'fuzzing' auto-interpretability alone, applied to SAE latent explanations, may not meaningfully distinguish between these underlying models.

Evaluation We considered standard SAE evaluation metrics alongside the auto-interpretability AUROC for Pythia models with between 70M and 6.9B parameters. As above, we broadly found that the randomly initialized and re-randomized models (Figure 2; blue, green, orange lines) were more similar to the trained model (Figure 2; gray lines) than to our control (Figure 2; black lines).

Notably, the cosine similarity between the original activation and the SAE reconstruction and explained variance are often far lower for the random control than the other models, and its reconstruction errors tend to increase across layers while the remaining variants decrease. For the random control, this can perhaps be explained by the fact that a Gaussian is the highest entropy distribution with fixed mean and variance (Jaynes, 2003); we speculate that Gaussian vectors are the 'least structured', in some sense, and thus hardest for SAEs to reconstruct. As Gaussian-distributed activations are propagated through successive layers, we would expect the activations to become less Gaussian and perhaps more 'sparse', i.e., easier to reconstruct (Section 4).

Interestingly, the randomized variants (blue and orange lines) are more similar to the trained model than the variant at initialization (green line). This is especially evident if we look at the L^1 norm values in larger models. We speculate that this pattern arises because parameter norms may differ greatly between a trained model and its state at initialization. In contrast, our randomization procedure was specifically designed to preserve parameter norms with respect to the trained model. The scale of parameters at different layers may be important, e.g., to control the growth of activations as they progress through the residual stream (Liu et al., 2020). In the AUROC plots, we find that for all but the control variant, AUROC increases with model size. We speculate that features become more specific as SAE size increases: in smaller SAEs, each latent must explain more of the input, making classification tasks easier for larger SAEs.

Figure 2 (row five) shows the cross-entropy (CE) loss score, or loss recovered, against model layer. This is the increase in the loss when the original model activations are replaced by their SAE reconstructions, divided by the increase when the activations are replaced by zeros ('ablated'). The results show that the 'trained' variant SAEs perform similarly to others from the literature (e.g., Kissane et al., 2024; Rajamanoharan et al., 2024a; Mudide et al., 2024). Importantly, the CE loss score only makes sense for the trained variant: for any of the randomized variants, the loss is very poor, regardless of whether the original or reconstructed activations are used.

Latent explanation complexity Despite sometimes similar auto-interpretability scores and evaluation metrics, we had expected that SAEs applied to trained vs. randomized transformers would discover qualitatively different features. In particular, we expected SAEs trained on the randomized variants to learn relatively simple features based on characteristics of the input text, but not more complex, abstract features as with trained transformers (Templeton et al., 2024). For qualitative examples, we provide a random sample of features and the corresponding maximally activating dataset examples for each variant of Pythia-6.9b in Appendix J, and more detailed information in Appendix L.

Anecdotally, we have observed that a significant proportion of SAE latents have non-zero activations only on a single token or a small number of distinct tokens within a text dataset (e.g., Lin and Bloom, 2024; Dooms and Wilhelm, 2024). Hence, a simple measure of the complexity of an SAE latent given a set of maximally activating examples is the degree to which the latent activates on a single token ID or multiple distinct IDs. Specifically, we quantify the number of token IDs in terms of the entropy of the observed distribution of latent activations over tokens: the greater the entropy, the more 'spread out' the latent activations, and the less token-specific the latent. We take this distribution to be the total latent activation per token across the set of maximally activating examples used to generate explanations for auto-interpretability. We show the relationship between entropy and 'fuzzing' AUROC score for individual latents in Appendix H.

We include the entropy of the observed distributions of latent activations over token IDs in the last row of Figure 2. The negative control variant displays a consistently high entropy, which is to be expected given that the embedding for a given token ID is sampled i.i.d. from a Gaussian on each

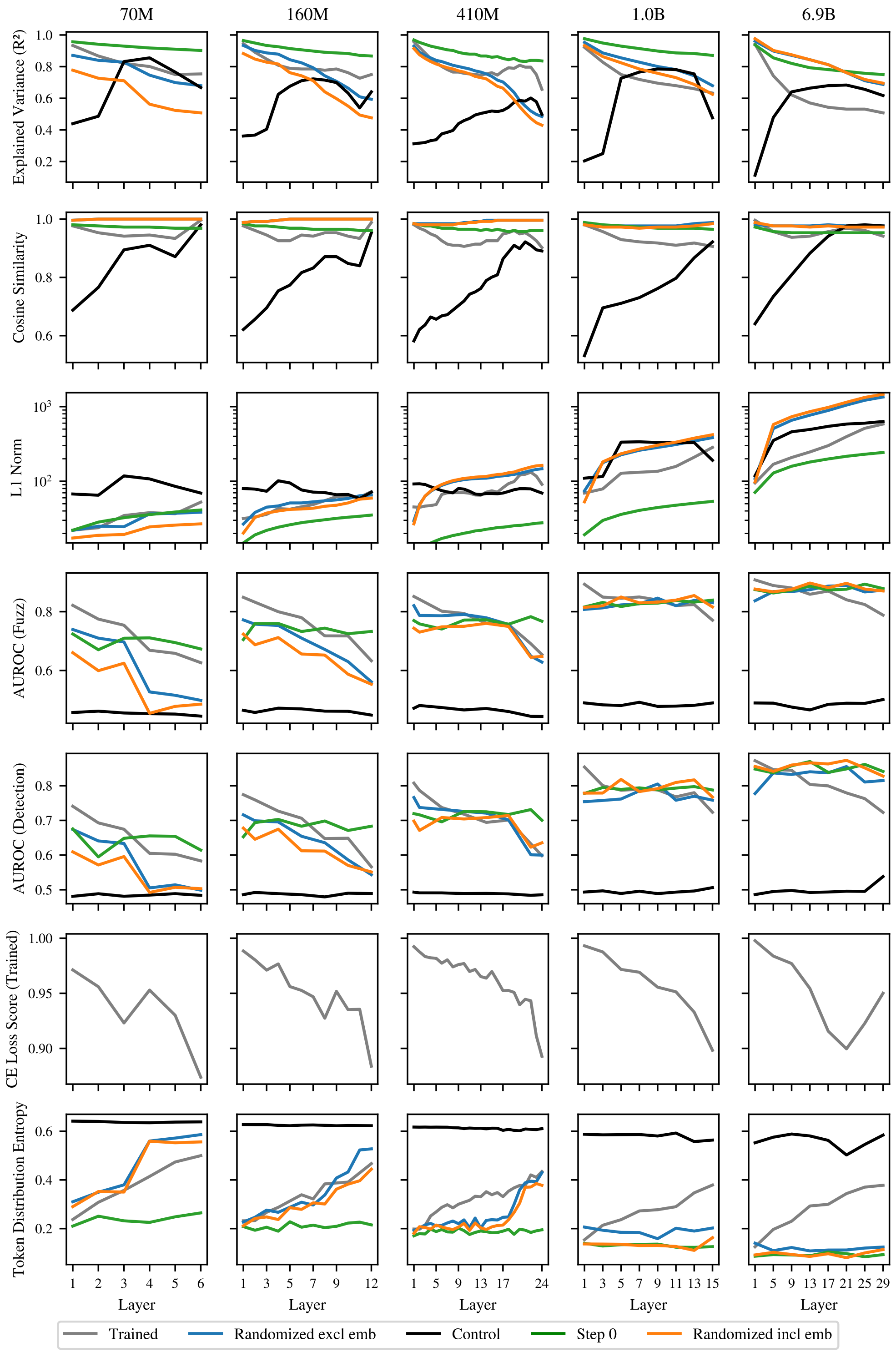

Figure 2: Comparison of sparse autoencoder performance across Pythia models (70M to 6.9B parameters). The different SAE variants show remarkably similar trends across model scales, with larger models exhibiting more consistent behavior across layers. All variants save for control achieve comparable performance despite fundamentally different initialization approaches.