
Towards Interpretability Without Sacrifice: Faithful Dense Layer Decomposition with Mixture of Decoders

James Oldfield^{m,q*} Shawn Im^m Sharon Li^m Mihalis A. Nicolaou^{c,i}
Ioannis Patras^q Grigoris G Chrysos^m

^m University of Wisconsin–Madison ^q Queen Mary University of London ^c University of Cyprus
ⁱ The Cyprus Institute

Abstract

Multilayer perceptrons (MLPs) are an integral part of large language models, yet their dense representations render them difficult to understand, edit, and steer. Recent methods learn interpretable approximations via neuron-level sparsity, yet fail to faithfully reconstruct the original mapping—significantly increasing model’s next-token cross-entropy loss. In this paper, we advocate for moving to *layer*-level sparsity to overcome the accuracy trade-off in sparse layer approximation. Under this paradigm, we introduce Mixture of Decoders (MxDs). MxDs generalize MLPs and Gated Linear Units, expanding pre-trained dense layers into tens of thousands of specialized sublayers. Through a flexible form of tensor factorization, each sparsely activating MxD sublayer implements a linear transformation with full-rank weights—preserving the original decoders’ expressive capacity even under heavy sparsity. Experimentally, we show that MxDs significantly outperform state-of-the-art methods (e.g., Transcoders) on the sparsity-accuracy frontier in language models with up to 3B parameters. Further evaluations on sparse probing and feature steering demonstrate that MxDs learn similarly specialized features of natural language—opening up a promising new avenue for designing interpretable yet faithful decompositions. Our code is included at: <https://github.com/james-oldfield/MxD/>.

1 Introduction

One strategy for addressing concerns about large language models’ (LLMs) [1, 2, 3] behavior is via a bottom-up approach to understanding and controlling the network internals—developing models of how and where human-interpretable features are represented in LLMs and how they affect the output [4, 5, 6]. Such a mechanistic understanding has proved helpful for a number of issues relating to safety and transparency, from controlling refusal of harmful requests [7] to detecting generation of unsafe code [6] and latent model knowledge [8].

However, developing models of LLMs’ internals faces challenges due to the dense nature of their representations [9, 10]. Indeed, many studies have found that individual neurons in MLP layers encode *multiple* distinct concepts. Rather than human-interpretable features being neatly aligned with individual neurons, they are often distributed across many [11, 12]. As a result, it is not straightforward to cleanly isolate specific concepts of interest in the models’ latent token representations.

Traditionally, imposing constraints on model form has offered a way to instill more predictable properties or structure. Indeed, there is a rich history of success with constraints in machine learning: from parts-based representations through non-negativity [13, 14], to structure through low-rankness or assumptions on geometry [15, 16]. With the particular issues posed by dense representations

*Corresponding author: jamesalexanderoldfield@gmail.com. Work done whilst at UW-Madison.

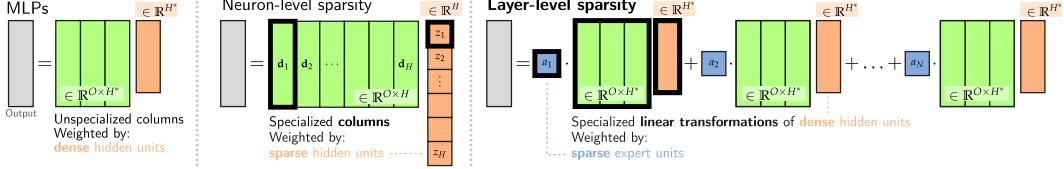


Figure 1: **Units of specialization for sparse layer variants:** *Neuron-level sparsity* of existing sparse MLPs [27, 26] (center) vs *layer-level sparsity* (right), which the proposed Mixture of Decoders (MxD) layer enables at scale. For GPT2, the dimensions are: $O = 768$, $H^* = O \cdot 4$, $H \approx N \approx O \cdot 32$.

in LLMs, *specialization through sparsity* has re-emerged as a dominating strategy for learning more interpretable representations. With prior work showing that sparser models both aid human explanation [17] and achieve higher scores on LLM-based auto-interpretability metrics [18, 19], sparsity is often used as a proxy for interpretability [20, 21]. To this end, many recent works—such as sparse autoencoders [22, 23, 6]—take inspiration from traditional sparse dictionary learning methodologies [24, 25], re-writing pre-trained LLMs’ activations as sparse, non-negative linear combinations of atoms in a learned overcomplete basis. However, as argued in [26], such approaches do not learn the functional mechanisms of LLMs’ layers, and their inherent post-hoc nature demands additional parameters and computation on top of the base models.

One alternative approach is to directly replace layers with more interpretable equivalents [28], such as with wide MLPs with sparsity constraints. Transcoders [27, 29, 30, 26] (TCs) are a recent example of this, training new MLPs to mimic the functional behavior of MLPs with *sparse* hidden units, which have recently been shown to also learn more interpretable features [26]. Thus, instead of relying on external post-hoc analysis, sparse MLP layers offer a way to distill specialized features directly into the model’s forward pass itself.

Both of the above methods for learning specialized features fall into the same category of what one may call ‘neuron-level sparsity’. Dictionary learning methods restrict the number of non-zero elements used from a learned dictionary, whilst sparse MLPs [27] limit the number of active rows used from a learned ‘decoder’ matrix. At its core, whilst this constraint is useful for interpretability, it is too restrictive—often heavily trading off accuracy for sparsity, poorly reconstructing the original model components [31, 28]. We argue that preserving the base models’ performance is a crucial component of sparse MLP layer approximations for the following two key reasons:

1. **Model faithfulness:** sparse layers that poorly approximate the original layers risk missing critical intricacies of the base models’ behavior or latent features [32]. Conversely, an accurate reconstruction (yielding similar downstream next-token loss) is some evidence that the combination of newly learned subcomputations faithfully emulates the base model.
2. **Practical adoption:** sparse layers that closely preserve base models’ performance are capable of *replacing* the existing MLPs, directly integrating specialized computation into the native forward pass. Otherwise, downstream use of the sparse layers’ features must run on top of the base models’ computation. This introduces additional inference-time cost to every forward pass, and restricts any analysis to post-hoc settings.

In this paper, we advocate for moving from *neuron-level* to *layer-level* sparsity (as illustrated in Figure 1) to address this. We propose the *Mixture of Decoders (MxD)* layer to overcome the sparsity-accuracy trade-off through scalable, resource-efficient conditional computation. Rather than individual vectors, MxDs learn interpretable sublayers as atomic units of specialization. This faithfully mirrors the functional form of dense layer we wish to approximate, and allows MxDs to readily generalize to modern MLP variants (i.e., the Gated Linear Unit [33]).

At a technical level, MxDs are constructed via a flexible tensor factorization [34] with the Hadamard product [35]. Through their parameter efficiency, MxDs scale the number of specialized layers far beyond what is feasible with classic sparse mixture of experts (MoEs) [36], and recover prior adapter-based MoEs [37, 38] as a special case. Crucially, we prove that the proposed tensor factorization in MxDs leads to each ‘expert’ sublayer implementing a linear transformation with full-rank weights—allowing faithful reconstruction even under heavy sparsity. Empirically, we demonstrate that MxDs significantly outperform alternative sparse MLP layers such as Transcoders [27] and Skip Transcoders

[26] on the sparsity-accuracy frontier. In addition to their faithfulness, MxDs remain competitive with the SOTA on interpretability metrics. **Our contributions can be summarized as follows:**

- We propose *Mixture of Decoders*, an instance of a flexible class of parameter-efficient MoE through Hadamard product-factorized weight tensors.
- We prove that each specialized MxD expert’s weights inherit up to the same rank as the original MLP’s decoder, providing faithful approximation even in very sparse models.
- Across 108 sparse layers in 4 LLMs (with up to 3B parameters) MxDs (i) pareto-dominate existing techniques on the sparsity-accuracy frontier yet (ii) remain competitive on 34 sparse probing and steering tasks, validating the interpretability of the learned experts.

2 Methodology

We first recall the technical details of language models’ MLP layers and existing approaches to sparse approximations in Section 2.1. We then introduce the proposed MxD in Section 2.2, outlining the attractive rank properties it inherits in Section 2.3 and factorized implementation in Section 2.4. We conclude with extensions to modern MLP layers in Section 2.5.

2.1 Preliminaries

Let $\mathbf{x} \in \mathbb{R}^I$ be the pre-MLP latent representation of a specific token at a given layer. Omitting bias terms throughout for brevity, the GPT2-style MLP layer produces the output vector $\mathbf{y} \in \mathbb{R}^O$ as:

$$\text{MLP}(\mathbf{x}) = \mathbf{D}^{*\top} \mathbf{z}^* \in \mathbb{R}^O, \quad \text{with } \mathbf{z}^* := \phi(\mathbf{E}^{*\top} \mathbf{x}) \in \mathbb{R}^{H^*}, \quad (1)$$

where $\mathbf{E}^* \in \mathbb{R}^{I \times H^*}$, $\mathbf{D}^* \in \mathbb{R}^{H^* \times O}$ are the learnable ‘encoder’ and ‘decoder’ parameters respectively, and $\phi(\cdot)$ is an activation function, often a GELU [39]. We use * to denote the weights/dimensions of the pre-trained base LLM.

Sparse approximations One approach to learning interpretable features in MLPs is to train new, wider MLPs with *sparse* hidden units to reconstruct the original layer’s outputs [27, 26, 30, 29], reminiscent of dictionary learning techniques [25]. In general, sparse MLPs share the model form:

$$\text{SMLP}(\mathbf{x}) = \mathbf{D}^\top \mathbf{z} = \sum_{h=1}^H z_h \mathbf{d}_h \in \mathbb{R}^O, \quad \text{with } \mathbf{z} := \mathcal{S}(\mathbf{E}^\top \mathbf{x}) \in \mathbb{R}^H, \quad (2)$$

where $\mathcal{S}(\cdot)$ is a sparsity-inducing function (such as the top- K [23] activation used in this paper). Here, the dimensionality of sparse MLPs’ learnable weights $\mathbf{E} \in \mathbb{R}^{I \times H}$, $\mathbf{D} \in \mathbb{R}^{H \times O}$ are set as $H \gg H^*$ such that the hidden layer is significantly larger than that of the original MLP. The original post-MLP output vectors are approximated as a K -sparse, non-negative linear combination of the rows \mathbf{d}_n of a newly learned decoder matrix. Whilst this model form has been shown to learn interpretable, specialized features z_h in language models [27, 26], their poor reconstruction is of questionable faithfulness and limits their use as a layer replacement in practice.

2.2 Mixture of Decoders

We now detail the proposed Mixture of Decoders (MxD) layer, which overcomes the sparsity-accuracy trade-off by treating sparsely activating linear layers as the atomic unit of specialization. We approximate the original MLP with a conditional combination of N linear transformations:

$$\text{MxD}(\mathbf{x}) = \sum_{n=1}^N a_n (\mathbf{W}_n^\top \mathbf{z}) \in \mathbb{R}^O, \quad (3)$$

where $\mathbf{a} := \mathcal{S}(\mathbf{G}^\top \mathbf{x}) \in \mathbb{R}^N$ are *sparse* ‘expert coefficients’ from learnable gating matrix $\mathbf{G} \in \mathbb{R}^{I \times N}$, and $\mathbf{z} := \phi(\mathbf{E}^\top \mathbf{x}) \in \mathbb{R}^H$ is the *dense* output from an encoder. Here, $\mathbf{W} \in \mathbb{R}^{N \times H \times O}$ is a third-order tensor of parameters collating all N experts’ decoder weights $\mathbf{W}(n, :, :) = \mathbf{W}_n \in \mathbb{R}^{H \times O}$. In MxDs, we use a large N to scale the feature specialization, and set $H := H^*$ to match the original MLP’s smaller hidden dimension.

With the gate routing each token to just its top- K experts, each $\mathbf{W}_n \in \mathbb{R}^{H \times O}$ receives a gradient signal from only a specific set of semantically similar tokens. This implicit clustering naturally leads experts to specialize in feature-specific subcomputations, while collectively covering the layer's full functionality. MxDs in Equation (3) also directly inherit the MLP layers' original functional form, avoiding the need to impose sparsity and non-negativity constraints on the hidden units $\mathbf{z} \in \mathbb{R}^H$. However, MxD decoders naively require a prohibitive NHO parameters—preventing N from scaling to tens of thousands of specialized components. To achieve parameter-efficiency whilst retaining layer capacity for faithful layer approximation, we parameterize MxDs' third-order weight tensor $\mathbf{W} \in \mathbb{R}^{N \times H \times O}$ specifically to yield full-rank expert weights, defined elementwise as:

$$\mathbf{W}(n, h, :) = \mathbf{c}_n * \mathbf{d}_h \in \mathbb{R}^O, \quad \forall n \in \{1, \dots, N\}, h \in \{1, \dots, H\}, \quad (4)$$

where $*$ is the *Hadamard product* [34, 35], and $\mathbf{c}_n, \mathbf{d}_h \in \mathbb{R}^O$ are the rows of learnable weights $\mathbf{C} \in \mathbb{R}^{N \times O}, \mathbf{D} \in \mathbb{R}^{H \times O}$. Intuitively, \mathbf{D} implements a base transformation modulated by the N specialized units in \mathbf{C} . Additional technical motivation for this parameterization with tensor methods can be found in Appendix A.3. This brings MxDs' parameter count down significantly to $O \cdot (N + H)$ from NHO in Equation (3) with N full decoders. One can then vary N to parameter-match sparse MLP layers. We next detail how this design (i) retains expressivity in each unit for faithful layer approximation under sparsity in Section 2.3 and (ii) yields a simple forward pass in Section 2.4.

2.3 MxDs are rank-preserving

In the original LLM, the linear transformation from the hidden units to the output is constrained by the rank of the original MLP's decoder matrix $\mathbf{D}^* \in \mathbb{R}^{H^* \times O}$. Under only mild technical conditions, *every* expert's weight matrix in MxDs inherits the rank of $\mathbf{D} \in \mathbb{R}^{H \times O}$, thus allowing it to match that of the original MLP's decoder, despite its parameter-efficiency:

Lemma 1 (Decoder rank preservation). *We can materialize linear expert n 's weight matrix as $\mathbf{W}(n, :, :) = \mathbf{W}_n = \mathbf{D} \operatorname{diag}(\mathbf{c}_n) \in \mathbb{R}^{H \times O}$. Assuming $\operatorname{diag}(\mathbf{c}_n) \in \mathbb{R}^{O \times O}$ is a diagonal matrix with no zeros along its diagonal (and thus invertible), we then have*

$$\operatorname{rank}(\mathbf{W}_n) = \operatorname{rank}(\mathbf{D} \operatorname{diag}(\mathbf{c}_n)) = \operatorname{rank}(\mathbf{D}).$$

The proof is found in Appendix A.1, which first derives the matrix-valued expression for each expert from Equation (4) and then applies a standard rank equality. At a sparsity level of K , each MxD output vector is a weighted sum of K -many linear transformations (each with potentially full-rank weights) of the dense hidden units \mathbf{z} . As a result, MxDs retain layer capacity even under high sparsity. Sparse MLPs' hidden units have only K non-zero elements in contrast—each output in Equation (2) is therefore confined to a K -dimensional subspace of \mathbb{R}^O , potentially limiting the capacity of sparse MLPs to faithfully approximate the original mapping in the small K regime desirable for interpretability (mirroring speculations by [26]). Further, whilst alternative soft linear MoEs achieve scalability through low-rankness [40], Lemma 1 states that no such rank constraints are present in MxDs. For approximating existing MLP layers where low-rank assumptions may not hold, MxDs are consequently a more suitable class of conditional layer.

2.4 Factorized forward pass

MxDs compute a linear combination of N linear transformations of the dense vector. With the proposed Hadamard-factorized weights, this yields a simple implementation.

Lemma 2 (Hadamard-factorized MoE forward pass). *Let $\mathbf{z} \in \mathbb{R}^H$ and $\mathbf{a} \in \mathbb{R}^N$ denote the MLP hidden units and expert coefficients respectively. Further, denote the decoder matrices as $\mathbf{C} \in \mathbb{R}^{N \times O}, \mathbf{D} \in \mathbb{R}^{H \times O}$ parameterizing $\mathbf{W} \in \mathbb{R}^{N \times H \times O}$. MxD's forward pass can be re-written as:*

$$\text{MxD}(\mathbf{x}) = \sum_{n=1}^N a_n (\mathbf{W}_n^\top \mathbf{z}) = (\mathbf{C}^\top \mathbf{a}) * (\mathbf{D}^\top \mathbf{z}). \quad (5)$$

The proof is found in Appendix A.2. We include a notebook at <https://github.com/james-oldfield/MxD/blob/main/form-equivalence.ipynb> showing the equivalence in PyTorch. Further, please see Appendix A.5 for a discussion of how the Hadamard factorization relates

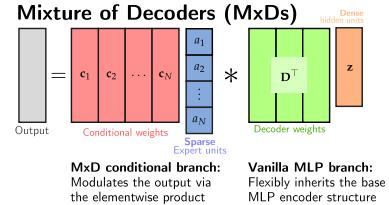


Figure 2: **Mixture of Decoders** extends the base MLP/GLU layers with a conditional 'expert' branch, modulating the MLP's outputs.

Table 1: **Model formulations of related work:** $\mathbf{x} \in \mathbb{R}^I$, $\mathbf{y} \in \mathbb{R}^O$ are the pre- and post-MLP representations respectively, \mathbf{z} are the hidden units, and \mathbf{a} is the vector of the ‘expert coefficients’ for MxD. Model-specific encoders/decoders \mathbf{E} , \mathbf{D} map between the hidden units and output.

| | MLPs [3] | SAEs [22] | Transcoders [27] | Skip Transcoders [26] | MxDs (Ours) |
|------------------|--|--|--|--|--|
| Model form | $\mathbf{y} = \mathbf{D}^* \mathbf{z}^*$ | $\mathbf{y} \approx \mathbf{D}^T \mathbf{z}$ | $\mathbf{y} \approx \mathbf{D}^T \mathbf{z}$ | $\mathbf{y} \approx \mathbf{D}^T \mathbf{z} + \mathbf{S}^T \mathbf{x}$ | $\mathbf{y} \approx \sum_n a_n (\mathbf{W}_n^T \mathbf{z})$ |
| Sparse component | None | $\mathbf{z} = \mathcal{S}(\mathbf{E}^T \mathbf{y}) \in \mathbb{R}^H$ | $\mathbf{z} = \mathcal{S}(\mathbf{E}^T \mathbf{x}) \in \mathbb{R}^H$ | $\mathbf{z} = \mathcal{S}(\mathbf{E}^T \mathbf{x}) \in \mathbb{R}^H$ | $\mathbf{a} = \mathcal{S}(\mathbf{G}^T \mathbf{x}) \in \mathbb{R}^N$ |

to prior parameter-efficient MoEs with element-wise scaling [37], and Appendix B.6 for performance/computational cost comparisons.

2.5 Extending MxDs to GLUs

In contrast to methods imposing neuron-level sparsity [22, 27, 26], MxDs do not make assumptions about the base layer’s encoder architecture or activation function. As a result, MxDs readily generalize to alternative architectures such as the Gated Linear Units (GLUs) [33] used in recent LLMs [1, 2]. Recall that GLUs’ hidden units are computed as $\mathbf{z}_{\text{GLU}} = \psi(\mathbf{E}_{\text{GLU}}^T \mathbf{x}) * (\mathbf{E}^T \mathbf{x}) \in \mathbb{R}^H$, with additional GLU parameters $\mathbf{E}_{\text{GLU}} \in \mathbb{R}^{I \times H}$ and GLU activation function ψ (e.g., Swish [1]). By substituting in the GLU hidden representations, MxDs straightforwardly extend the GLU model form too:

$$\text{MxD}_{\text{GLU}}(\mathbf{x}) = \sum_{n=1}^N a_n \mathbf{W}_n^T \left(\underbrace{\psi(\mathbf{E}_{\text{GLU}}^T \mathbf{x}) * (\mathbf{E}^T \mathbf{x})}_{\text{GLU hidden units}} \right) = (\mathbf{C}^T \mathbf{a}) * \mathbf{D}^T (\psi(\mathbf{E}_{\text{GLU}}^T \mathbf{x}) * (\mathbf{E}^T \mathbf{x}))$$

where $\mathbf{a} := \mathcal{S}(\mathbf{G}^T \mathbf{x}) \in \mathbb{R}^N$ are the expert units, and $\mathbf{W}_n = \mathbf{D} \text{diag}(\mathbf{c}_n) \in \mathbb{R}^{H \times O}$ as before. For a technical discussion of GLUs and their relationship to MxDs, we refer readers to Appendix A.4—through the theoretical results developed in this paper, we show that GLU encoders themselves can be viewed as a mixture of rank-1 linear experts (in contrast to the rank-preserving MxDs).

3 Experiments

The experimental section in the main paper is split into two parts. Section 3.1 first demonstrates how MxDs perform significantly better on the accuracy-sparsity frontier as sparse MLP layer approximations on 4 LLMs. We then demonstrate in Section 3.2 that MxD’s features retain the same levels of specialization through sparse probing and steering evaluations. Thorough ablation studies, experiments with matrix rank, and comparisons to low rank MoEs are presented in Appendix B.

3.1 Sparse approximations of MLPs in LLMs

In this section, we perform experiments approximating LLMs’ existing feed-forward layers with sparse MLPs, establishing that MxDs better navigate the sparsity-accuracy frontier, more faithfully approximating the base models’ MLPs than the SOTA baseline methods.

Implementation details We train on 4 base models: GPT2-124M [3], Pythia-410m, Pythia-1.4b [41], and Llama-3.2-3B [1] with up to 80k experts/features. We train all sparse layers on a total of 480M tokens of OpenWebText [42], with learning rate $1e-4$ and a context length of 128, initializing the output bias as the empirical mean of the training tokens, and \mathbf{D} in MxDs as the zero-matrix (following [26]). We vary N in MxD layers to parameter-match Transcoders in all experiments, with parameter counts and dimensions shown in Table 2. For Llama3.2-3B, we use the Swish-GLU variant of MxD and GELU-MLP MxDs for the other three models, matching the architectures of their base encoders. Through ablation studies in Appendix B.8 we show that MxDs using the GELU/GLU variants are much more accurate layer approximators than the ReLU variants. Full experimental details are included in Appendix D. Whilst we do not have the computational resources to show similarly thorough experiments on even larger models, we expect MxDs to scale just as well as sparse MLPs to models with tens of billions of parameters or more.

Table 2: Sparse layer parameters/dimensions: H denotes the size of the layers’ hidden units and N is the expert count. MxDs perform almost as many linear transformations as the baselines have features.

| Model | GPT2-124M | | | Pythia-410M | | | Pythia-1.4B | | | Llama-3.2-3B | | |
|-----------------------|--------------|-------------|---------------|--------------|-------------|---------------|---------------|-------------|---------------|--------------|-------------|---------------|
| | Params | H | N | Params | H | N | Params | H | N | Params | H | N |
| Transcoders [27] | 37.7M | 24,576 | — | 67.1M | 32,768 | — | 268.5M | 65,536 | — | 604M | 98,304 | — |
| Skip Transcoders [26] | 38.4M | 24,576 | — | 68.2M | 32,768 | — | 272.7M | 65,536 | — | 614M | 98,304 | — |
| MxDs | 37.7M | 3072 | 21,490 | 67.1M | 4096 | 28,658 | 268.4M | 8192 | 57,330 | 604M | 8202 | 86,015 |

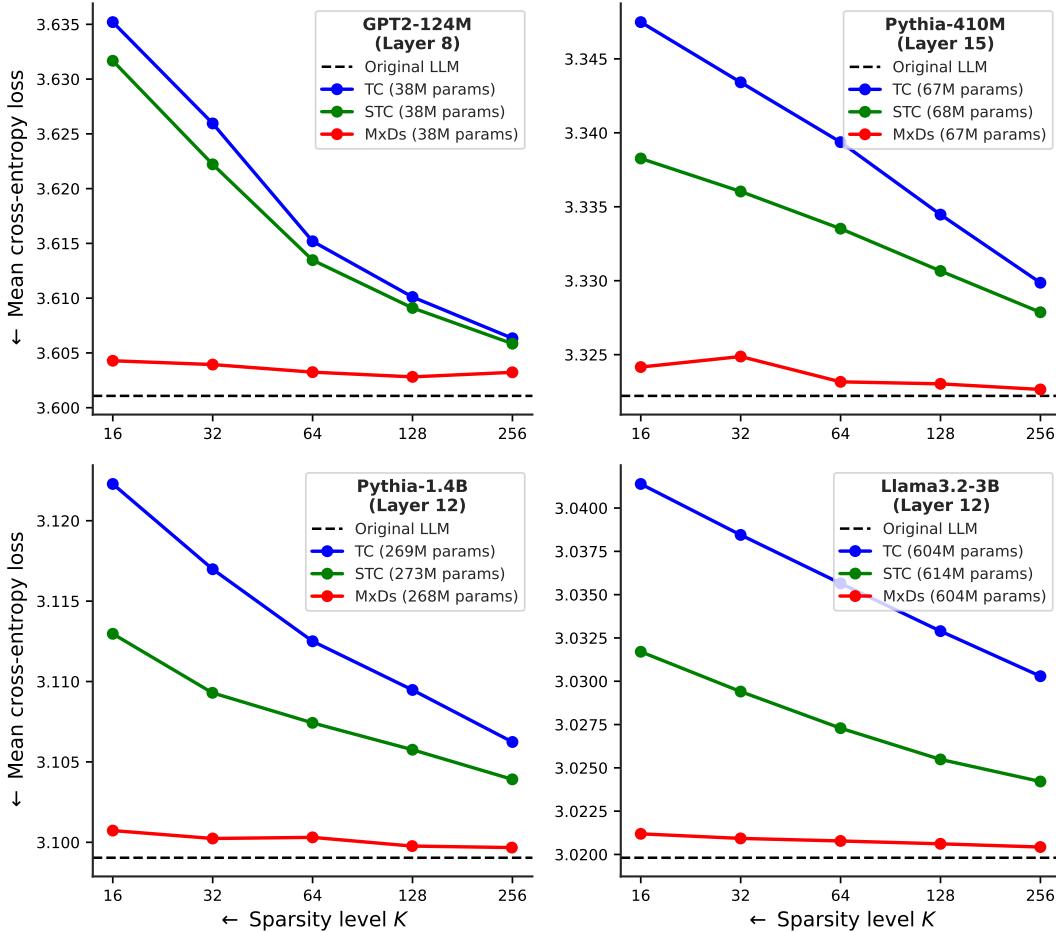


Figure 3: Model cross-entropy loss preserved when replacing MLPs with Transcoders [27], Skip Transcoders [26], and MxDs, as a function of the number of active units K (hidden neurons/experts). We highlight that MxDs have consistently lower loss at all levels of sparsity.

Objective function Given the frozen weights of the MLP, we train sparse layers to minimize the normalized reconstruction loss between its output and that of the original MLP layer with objectives of the form $\mathcal{L} = \mathbb{E}_{\mathbf{x}} \left[\frac{\|\text{MLP}(\mathbf{x}) - f(\mathbf{x})\|_2^2}{\|\text{MLP}(\mathbf{x})\|_2} \right]$, where $f(\cdot)$ denotes the various learnable sparse MLP layers. This follows the protocol of past work [27, 26], where the new sparse layers’ parameters alone are trained directly on the output of the MLP. To compare with recent work [26], we adopt the TopK activation function [23] for sparsity-inducing function $\mathcal{S}(\cdot)$, removing the need for an additional sparsity penalty. Please see Appendix A.6 for details on the TopK activation function.

3.1.1 Results: sparsity vs faithfulness

We train an exhaustive set of 60 sparse MLP approximations across 4 diverse LLMs with up to 3B parameters. We show in Figure 3 the resulting downstream base model cross-entropy loss when using the trained sparse layers in place of the original MLPs. As can be seen, not only do the proposed

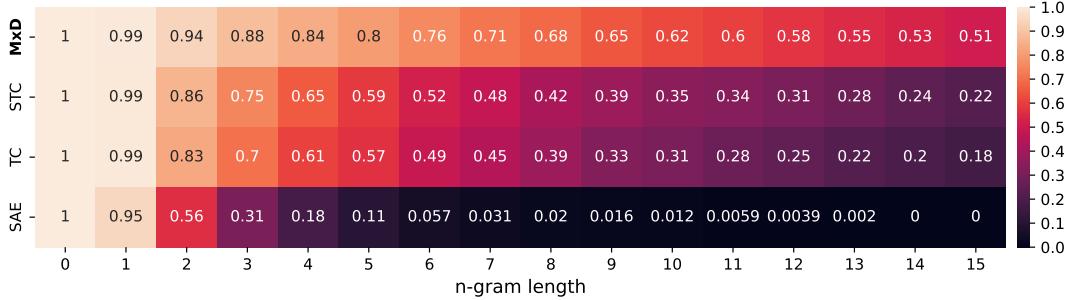
MxD layers outperform Transcoders [27] notably, but **model performance is similarly preserved at all sparsity levels in MxD layers**. Please also see Figure 9 for results with normalized MSE, where MxDs’ reconstruction errors are up to an order of magnitude smaller. Full results on additional layers are included in Appendix B.3 for 48 more trained sparse layers.

The recent ‘Skip Transcoders’ (STCs) [26], introduce an additional IO parameters with a skip connection $\mathbf{S} \in \mathbb{R}^{I \times O}$ mapping the input directly to the output with $y \approx \mathbf{D}^\top \mathbf{z} + \mathbf{S}^\top \mathbf{x}$. STC layers thus have considerably more parameters (e.g., STCs on 11ama3.2-3B have 10M more parameters than MxDs). Despite the smaller parameter counts, we find MxDs consistently outperform STCs on the sparsity-accuracy frontier, attesting to the benefits of MxDs’ model form.

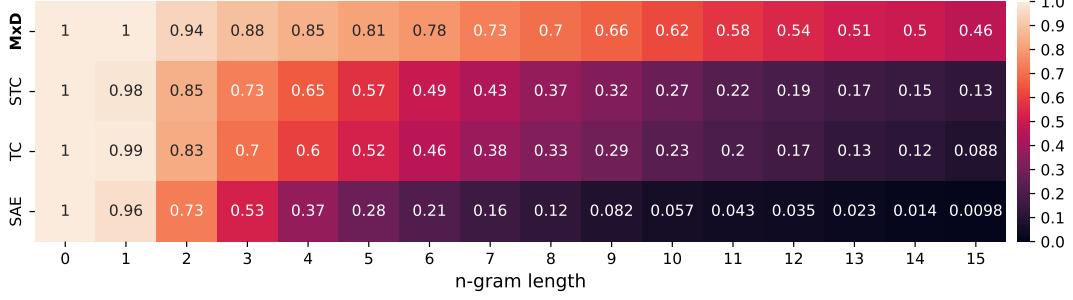
3.1.2 Faithfulness in output space

Next, we perform experiments comparing the faithfulness of sparse layers as their computation propagates to the model output space. We sample 16 future tokens with the base model and then measure how similar the same generations are when the target MLP layer is replaced with the sparse layers. We use 512 text snippets from OpenWebText, and take the first 4 words of each as the initial prompts, generating 16 future tokens after each prompt. We plot in Figures 4a and 4b the percentage of the samples’ continuations that are identical in the original LLM and hooked LLMs up to n future tokens ahead. We note that this is a rather punishing task—any small deviations quickly compound as n grows. Despite this, we see that the MxDs match the future token generations far better than the baselines, exhibiting more faithfulness in model output space (as well as in latent space).

Please see qualitative examples of the first 8 prompts and the subsequent ‘diffs’ (using Python 3’s `difflib`) of the generated tokens in the Appendix in Figures 7 and 8.



(a) Pythia-410m.



(b) GPT2-124m.

Figure 4: Proportion of 512 generated samples that contain n predicted future words identical to the original model’s output when replacing the base LLM’s MLP layer with the sparse layers.

3.2 Feature evaluations

The accurate reconstruction of MxD models in Section 3.1 provides some evidence that MxDs are faithfully emulating the original MLP layers’ functional mapping. However, for interpretability, we care equally about the extent to which the learned features correspond to specialized, human-interpretable concepts. We confirm that MxD’s features compete with the baselines quantitatively in

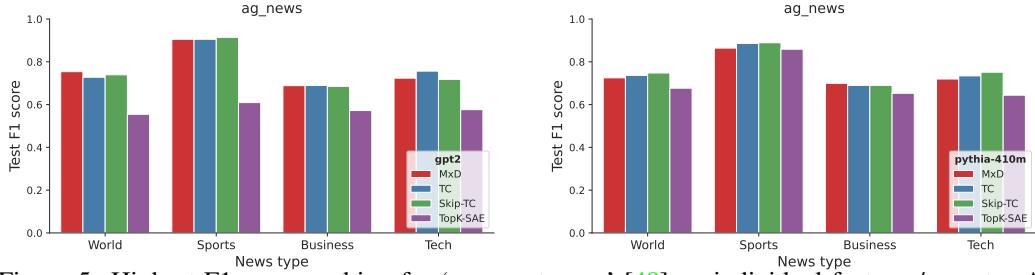


Figure 5: Highest F1 score probing for ‘news category’ [48] on individual features/experts. As expected, the MxDs remain competitive with the Transcoder baselines, outperforming TopK-SAEs.

two ways: through probing for known concepts in Section 3.2.1 and by steering the model using the learned features Section 3.2.2. For all experiments in this section, we use the $K = 32$ models.

Shared experts and specialization Interestingly, we find MxDs naturally learn a ‘shared’ expert performing a common base transformation—the remaining $K - 1$ active experts are thus free to dedicate their capacity to modelling features unique to individual tokens. This emergent shared/private processing complements recent trends to use shared experts *by design* in MoEs [43, 44, 45, 46, 47] with [43] arguing this facilitates greater specialization. Furthermore, one may view the skip connection in STCs [26] as performing an analogous role to the shared expert. With MxDs, however, *all* units have the same high capacity to accurately learn separate subcomputation regardless of the frequency or rarity of features.

We also observe that our trained MxDs exhibit very few ‘dead’ experts, as shown in Appendix C.1, with many experts contributing actively. Furthermore, initial ablations in Appendix C.2 show that one can train MxDs without shared experts if desired, at small performance cost. Please see qualitative results of activated tokens for particular experts in Appendix E.

3.2.1 Sparse probing with individual features/experts

One challenge is that the sparse layers learn features in an unsupervised manner. As pointed out in [23], we therefore do not know which high-level features we ought to expect the model to learn (or even whether they exist in the OpenWebText training data). Nonetheless, we can reasonably expect a useful unsupervised model to learn at least a handful of commonly occurring concepts and linguistic themes. We accordingly focus our evaluation on the relative abilities of the sparse models to learn features well-predicting a variety of binary features used in the literature.

Concretely, to quantify the extent to which sparse layer features reliably fire in response to common high-level, interpretable concepts of natural language, we adopt the experimental settings of [49, 23, 19], training binary probes on the individual units of specialization (sparse hidden units z_n for TCs/SAEs and expert units a_n for MxDs—all pre-activation). For probing of sample-level concepts, we mean-pool activations across all non-padding tokens [19]. We train separate probes on 100 features with the largest mean difference between positive and negative activations, as per [49].

We perform experiments on all 24 binary probing tasks in the SAEbench suite [19]. Four of which are shown in Figure 5, plotting the best F1 score (on a held-out set) for news topic classification in a 1-vs-all setting [48]. As can be seen, there exist individual MxD expert units that are predictive of various categories of news articles, competitive with the baselines. We refer readers to Appendix B.7 for additional experiments on 20 more sample-level probing tasks, 10 token-level probing tasks, and experimental details.

3.2.2 Feature steering

Specific features might reliably fire in response to interpretable patterns of the input, yet not contribute to the generation process. Here, we aim to test this functional role of features by steering the LLMs. We note that these experiments do not aim to establish TCs/MxDs as competitive with the SOTA for controllable LLM generation. Rather, we aim to validate that the learned features contribute mechanistically to the LLM’s forward pass in a predictable way.

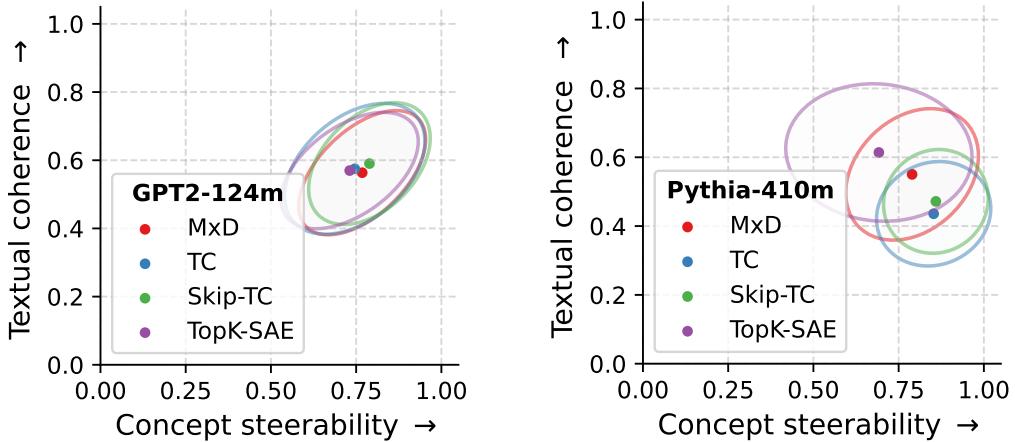


Figure 6: Mean score along dimensions of ‘textual coherence’ and ‘steerability’ of text generated by steering with the first 100 features of the sparse layers. Each sample is scored by 2 LLM judges.

Mechanisms for steering Let $\lambda \in \mathbb{R}$ be a hyperparameter controlling the desired ‘strength’ of the model edit. For TCs, we hook the forward pass at the relevant layer to increase the presence of target feature n with $\hat{\mathbf{y}} = \mathbf{y} + \lambda \mathbf{d}_n$. In contrast, MxDs can be steered with $\hat{\mathbf{y}} = \mathbf{y} + \lambda \cdot (\mathbf{W}_n^\top \mathbf{z})$. Intuitively, increasing the weight of an expert’s contribution in the forward pass modulates the token representation in the direction of the learned specialization.

Results We perform steering with the first 100 neurons/experts individually, using $\lambda := 100$ for all experiments. We generate a collection of 10 synthetic outputs for each neuron, each string consisting of 32 generated tokens to the prompt “Let’s talk about ”. We then ask two LLMs² to rate the collection of text along two dimensions separately: (1) the extent to which a shared concept, theme, or linguistic pattern is present throughout the generated collection of text, and (2) the grammatical fluency of the text (please see Appendix D.1 for the full prompt). As can be seen from the mean scores over the 100 neurons shown in Figure 6, MxDs are competitive with the baselines, exhibiting a similar trade-off between textual coherence and presence of concept as we expect.

4 Related work

Sparse decompositions Learning sparse [50, 25], non-negative [51] features of a data signal has found many applications in computer vision [15, 52, 53, 54] and natural language processing [55, 56, 57], motivated by the pursuit of interpretable, parts-based representations [13, 14]. In transformer-based language models [3], similar variants have been proposed for post-hoc analysis; sparse autoencoders (SAEs) are a popular method that rewrites latent features as non-negative combinations of atoms in a learned overcomplete dictionary, imposing either soft sparsity penalties [6, 22, 31] or thresholding activations directly [23, 58, 59]. Recent work aims to sparsify the existing layers of pretrained LLMs, either by learning new MLPs with sparse hidden units [29] (for circuit analysis [27] or more interpretable yet faithful computation [26, 60]), or by decomposing model parameters directly using attribution [61] and/or masking [62]. Despite the surge of interest in SAEs, many works are emerging drawing attention to their limitations—underperforming baselines for probing [63], unlearning [64], and steering [65], in addition to other pathologies [66, 32, 67, 68].

Conditional computation One natural alternative to static fully connected layers is conditional computation [69, 70]. Tracing back to the early work of [71, 72], single dense layers are replaced with specialized subunits—conditional on the input—as a form of layer-level sparsity. The Mixture of Experts (MoE) architecture [36, 73, 74] is a prominent example of conditional computation, breaking the link between parameter count and FLOPs. Consequently, MoEs have seen rapid adoption in SOTA models in recent years—scaling to very large parameter counts [75, 76, 77, 78, 79]. For

²We use `gemini-2.0-flash` and `llama-4-scout-17b-16e-instruct` as two independent LLM judges.

parameter-efficient instruction tuning [37] introduces conditional (IA)³ adapters [38], modulating the MLP hidden dimension with the Hadamard product. Our proposed formulation with factorized weight tensors yields ‘MoVs’ [37] as a less scalable special case (see Appendix A.5). In contrast, MxDs model the decoder output space directly for reconstruction, and also provide significantly more specialized units than [37], making MxDs more suitable for our goal of interpretability.

Whilst the primary focus of MoEs has been on their impressive capabilities, the literature has observed that individual experts often specialize in particular semantic patterns of the input data, despite not being trained to do so [80, 81, 43, 82, 83]. For example, many works find that data that are in some sense similar are routed to the same experts—specializing to object shapes [84], texture [85], image category [86], or semantic patterns in natural language [36]. In the context of large language models, this emergent property of specialization in MoEs has been a primary focus of recent work: from encouraging monosemantic experts [87] or sparsity amongst experts’ weights [88] to efficiently scaling the expert count for fine-grained specialization [40]. In contrast to these works exploring pre-training, we explore an efficient design of MoE to replace existing LLMs’ dense layers.

Interpretability by design Whilst MxDs follow the paradigm of specialization-via-sparsity, a number of promising interpretable mechanisms have been recently proposed that do not rely on sparsity constraints. For example, recent work explores bilinear layers [89], combinatorial structure [90], and/or tensor networks [91] as alternative ways of designing interpretable architectures. Notably, [89] show how the Hadamard product between two linear transformations of the same input vector allows direct interpretation (such as analyzing input-output feature interactions). Whilst MxDs yield a similar Hadamard product forward pass, it involves two different input vectors, and establishes a theoretical equivalence to mixture of experts and conditional computation when sparsity is present in one of the operands (i.e., the expert coefficients).

5 Conclusion

In this paper, we showed the benefits of decomposing dense layers’ computations as a mixture of interpretable sublayers. We proposed the Mixture of Decoders (MxD) layer to achieve this at scale, proving that MxDs’ linear experts preserve the matrix rank properties of the original decoders. Experimentally, we showed MxDs significantly outperform on the sparsity-accuracy frontier when trained to replace dense MLP layers. Quantitative results on sparse probing and feature steering demonstrated MxDs nonetheless learn specialized latent features similarly to existing interpretability techniques. Crucially, MxDs reexamine the dominating neuron-level sparsity paradigm of popular techniques, providing evidence that specialization doesn’t have to come with such a high cost to model performance. We believe MxDs (and specialization at the layer-level more generally) are an important step towards sparsity without sacrifice, and hope future work continues to build interpretable mechanisms that better preserve model capabilities. We are excited about future work exploring MxDs (and mixtures of linear transformations more generally) in alternative settings, such as for cross-layer features or transformations [92].

Limitations Our experiments show MxDs outperform on the sparsity-accuracy frontier on 4 diverse LLMs. Whilst we fully anticipate this trend to scale just as well as with sparse MLPs in even larger models, our experiments only provide direct evidence for LLMs with up to 3B parameters, given our limited resources. Furthermore, whilst the TopK activation can greatly reduce the decoders’ FLOPs, the large encoders in sparse MLPs and the gating function in MxDs remain an additional inference-time cost. Future work could explore hierarchical structures [87, 36] and/or efficient retrieval [93] for further reductions in FLOPs. Secondly, MoEs are prone to issues of expert imbalance [73], or collapse [94]. Through random weight initialization alone, we find MxDs in our experiments learn diverse experts without collapsing to the base decoder. However, we expect standard MoE load-balancing [95] or diversity losses [96] to be useful for MxDs should one need more explicit ways of encouraging expert diversity, or when training new interpretable MxD architectures end-to-end. With regards to feature evaluations, our steering experiments rely on LLMs as judges. Whilst this is commonplace in recent popular steering benchmarks [65], there is mixed evidence emerging about the reliability of all base models [97, 98]. Our experiments attempt to circumvent issues by reporting scores across two capable SOTA models, but we caution inferring too much into the absolute values of reported scores (rather than the relative performance between sparse layers).

Acknowledgments

James Oldfield is grateful to Demian Till for reviewing the draft and providing valuable feedback and suggestions, and would also like to thank Markos Georgopoulos, Benjamin Hayum, and Wisconsin AI Safety Initiative’s Safety Scholars for insightful discussions throughout the project. We are also grateful to the open-source [Zulip](#) platform for facilitating research discussion. **Sharon Li** is supported in part by the AFOSR Young Investigator Program under award number FA9550-23-1-0184, National Science Foundation under awards IIS-2237037 and IIS-2331669, Office of Naval Research under grant number N00014-23-1-2643, Schmidt Sciences Foundation, Open Philanthropy, Alfred P. Sloan Fellowship, and gifts from Google and Amazon. **Shawn Im** is also supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 2137424. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support was also provided by the Graduate School and the Office of the Vice Chancellor for Research at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation. **Mihalis Nicolaou** is supported in part by the TensorICE project (EXCELLENCE/0524/0407), implemented under the social cohesion programme “THALIA 2021-2027”, co-funded by the European Union through the Research and Innovation Foundation.

References

- [1] 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.
- [2] Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.
- [3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [4] Zhengxuan Wu, Atticus Geiger, Thomas Icard, Christopher Potts, and Noah Goodman. Interpretability at scale: Identifying causal mechanisms in Alpaca. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, volume 36, pages 78205–78226. Curran Associates, Inc., 2023.
- [5] Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 160–187. PMLR, 01–03 Apr 2024.
- [6] Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet*. Anthropic, 2024.
- [7] Andy Ardit, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.
- [8] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *Int. Conf. Learn. Represent. (ICLR)*, 2023.
- [9] David E. Rumelhart and James L. McClelland. *A General Framework for Parallel Distributed Processing*, pages 45–76. 1987.
- [10] Geoffrey E Hinton. Distributed representations. 1984.
- [11] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. <https://distill.pub/2020/circuits/zoom-in>.
- [12] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition, 2022.

- [13] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [14] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [15] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.
- [16] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 4690–4699, 2019.
- [17] Vikram V. Ramaswamy, Sunnie S. Y. Kim, Ruth C. Fong, and Olga Russakovsky. Overlooked factors in concept-based explanations: Dataset choice, concept learnability, and human capability. *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 10932–10941, 2022.
- [18] Caden Juang, Gonçalo Paulo, Jacob Drori, and Nora Belrose. Open source automated interpretability for sparse autoencoder features. <https://blog.eleuther.ai/autointerp/>, July 2024. EleutherAI Blog.
- [19] Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Calum McDougall, Yeu-Tong Lau, Eoin Farrell, Arthur Conmy, Kola Ayonrinde, Demian Till, Matthew Wearden, Samuel Marks, and Neel Nanda. SAEbench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *Int. Conf. Mach. Learn. (ICML)*, 2025.
- [20] Zachary Chase Lipton. The mythos of model interpretability. *Communications of the ACM*, 61:36 – 43, 2016.
- [21] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna M. Wallach. Manipulating and measuring model interpretability. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2018.
- [22] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *Int. Conf. Learn. Represent. (ICLR)*, 2023.
- [23] Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *Int. Conf. Learn. Represent. (ICLR)*, 2025.
- [24] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.
- [25] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. doi: 10.1109/TSP.2006.881199.
- [26] Gonçalo Paulo, Stepan Shabalin, and Nora Belrose. Transcoders beat sparse autoencoders for interpretability. *arXiv preprint arXiv:2501.18823*, 2025.
- [27] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2024.
- [28] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025.
- [29] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. <https://transformer-circuits.pub/2023/monosemantic-features/index.html>.
- [30] Samuel Marks, Adam Karvonen, and Aaron Mueller. dictionary_learning. https://github.com/saprmarks/dictionary_learning, 2024.

- [31] Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *arXiv preprint arXiv:2407.14435*, 2024.
- [32] Joshua Engels, Logan Riggs, and Max Tegmark. Decomposing the dark matter of sparse autoencoders. *arXiv preprint arXiv:2410.14670*, 2024.
- [33] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.
- [34] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [35] Grigoris G Chrysos, Yongtao Wu, Razvan Pascanu, Philip Torr, and Volkan Cevher. Hadamard product in deep learning: Introduction, advances and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2025. doi: 10.1109/TPAMI.2025.3560423.
- [36] Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Int. Conf. Learn. Represent. (ICLR)*, 2017.
- [37] Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermis, Aycı̄ Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *Int. Conf. Learn. Represent. (ICLR)*, 2024.
- [38] Haokun Liu, Derek Tam, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.
- [39] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.
- [40] James Oldfield, Markos Georgopoulos, Grigoris Chrysos, Christos Tzelepis, Yannis Panagakis, Mihalis Nicolaou, Jiankang Deng, and Ioannis Patras. Multilinear mixture of experts: Scalable expert specialization through factorization. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2024.
- [41] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Afrah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *Int. Conf. Mach. Learn. (ICML)*, pages 2397–2430. PMLR, 2023.
- [42] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. OpenWebText corpus. <http://Skylion007.github.io/OpenWebTextCorpus>, 2019.
- [43] Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models, 2024.
- [44] Meta AI. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, 2025. URL <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>. Accessed: 2025-04-06.
- [45] Qwen Team. Qwen1.5-MoE: Matching 7b model performance with 1/3 activated parameters", February 2024. URL <https://qwenlm.github.io/blog/qwen-moe/>.
- [46] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Kegin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.
- [47] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei

Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report, 2025.

- [48] Antonio Gulli. Ag corpus of news articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html, 2005.
- [49] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.
- [50] Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 366–373, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.
- [51] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.
- [52] Edo Collins, Radhakrishna Achanta, and Sabine Süsstrunk. *Deep Feature Factorization for Concept Discovery*, page 352–368. Springer International Publishing, 2018. ISBN 9783030012649. doi: 10.1007/978-3-030-01264-9_21.
- [53] James Oldfield, Christos Tzelipis, Yannis Panagakis, Mihalis Nicolaou, and Ioannis Patras. PandA: Unsupervised learning of parts and appearances in the feature maps of GANs. In *Int. Conf. Learn. Represent. (ICLR)*, 2023.
- [54] Yue Song, Thomas Anderson Keller, Yisong Yue, Pietro Perona, and Max Welling. Unsupervised representation learning from sparse transformation analysis. *arXiv preprint arXiv:2410.05564*, 2024.
- [55] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’03, page 267–273, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136463. doi: 10.1145/860435.860485.
- [56] Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative matrix factorization for interactive topic modeling and document clustering. *Partitional clustering algorithms*, pages 215–243, 2015.
- [57] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495, 2018. doi: 10.1162/tacl_a_00034.
- [58] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.
- [59] Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK sparse autoencoders. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024.
- [60] Lucy Farnik, Tim Lawson, Conor Houghton, and Laurence Aitchison. Jacobian sparse autoencoders: Sparsify computations, not just activations, 2025.
- [61] Dan Braun, Lucius Bushnaq, Stefan Heimersheim, Jake Mendel, and Lee Sharkey. Interpretability in parameter space: Minimizing mechanistic description length with attribution-based parameter decomposition. *arXiv preprint arXiv:2501.14926*, 2025.
- [62] Lucius Bushnaq, Dan Braun, and Lee Sharkey. Stochastic parameter decomposition. *arXiv preprint arXiv:2506.20790*, 2025.
- [63] Subhash Kantamneni, Joshua Engels, Senthooran Rajamanoharan, Max Tegmark, and Neel Nanda. Are sparse autoencoders useful? A case study in sparse probing, 2025.
- [64] Eoin Farrell, Yeu-Tong Lau, and Arthur Conmy. Applying sparse autoencoders to unlearn knowledge in language models, 2024.
- [65] Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Int. Conf. Mach. Learn. (ICML)*, 2025.

[66] David Chanin, James Wilken-Smith, Tomáš Dulka, Hardik Bhatnagar, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders, 2024.

[67] Patrick Leask, Bart Bussmann, Michael Pearce, Joseph Bloom, Curt Tigges, Noura Al Moubayed, Lee Sharkey, and Neel Nanda. Sparse autoencoders do not find canonical units of analysis, 2025.

[68] Gonçalo Paulo and Nora Belrose. Sparse autoencoders trained on the same data learn different features, 2025.

[69] Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, 44(11):7436–7456, 2021.

[70] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. In *Int. Conf. Mach. Learn. Worksh. (ICMLW)*, 2015.

[71] Robert A Jacobs, Michael I Jordan, and Andrew G Barto. Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive science*, 15(2):219–250, 1991.

[72] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[73] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[74] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. ST-MoE: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

[75] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. In *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[76] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *Int. Conf. Mach. Learn. (ICML)*, pages 5547–5569. PMLR, 2022.

[77] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[78] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruijie Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song,

Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025.

[79] Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *Int. Conf. Learn. Represent. (ICLR)*, 2024.

[80] Aya Abdelsalam Ismail, Sercan O Arik, Jinsung Yoon, Ankur Taly, Soheil Feizi, and Tomas Pfister. Interpretable mixture of experts. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[81] Marmik Chaudhari, Idhant Gulati, Nishkal Hundia, Pranav Karra, and Shivam Raval. MoE lens - an expert is all you need. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*, 2025.

[82] Huy Nguyen, Xing Han, Carl Harris, Suchi Saria, and Nhat Ho. On expert estimation in hierarchical mixture of experts: Beyond softmax gating functions, 2025.

[83] Stefan Nielsen, Rachel Teo, Laziz Abdullaev, and Tan Minh Nguyen. Tight clusters make specialized experts. In *Int. Conf. Learn. Represent. (ICLR)*, 2025.

[84] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. CondConv: Conditionally parameterized convolutions for efficient inference. *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 32, 2019.

[85] Basil Mustafa, Carlos Riquelme Ruiz, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. Multimodal contrastive learning with LIMoE: the language-image mixture of experts. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.

[86] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 34:8583–8595, 2021.

[87] Jungwoo Park, Ahn Young Jin, Kee-Eung Kim, and Jaewoo Kang. Monet: Mixture of monosemantic experts for transformers. In *Int. Conf. Learn. Represent. (ICLR)*, 2025.

[88] Xingyi Yang, Constantin Venhoff, Ashkan Khakzar, Christian Schroeder de Witt, Puneet K. Dokania, Adel Bibi, and Philip Torr. Mixture of experts made intrinsically interpretable, 2025.

[89] Michael T Pearce, Thomas Dooms, Alice Rigg, Jose Oramas, and Lee Sharkey. Bilinear MLPs enable weight-based mechanistic interpretability. In *Int. Conf. Learn. Represent. (ICLR)*, 2025. URL <https://openreview.net/forum?id=I0kPk1UKS>.

[90] Micah Adler, Dan Alistarh, and Nir Shavit. Towards combinatorial interpretability of neural computation. *arXiv preprint arXiv:2504.08842*, 2025.

[91] Thomas Dooms, Ward Gauderis, Geraint A. Wiggins, and Jose Oramas. Compositionality unlocks deep interpretable models. In *AAAI'25 workshop on CoLoRAI - Connecting Low-Rank Representations in AI*, 2025. URL <https://openreview.net/forum?id=bXAt5iZ691>.

[92] Jack Lindsey, Brian Chen, Adam Pearce, Sasha Hydrie, and Thomas Conerly. Sparse mixtures of linear transforms. <https://transformer-circuits.pub/2025/bulk-update/index.html>, 2025. Preliminary research update, Transformer Circuits.

[93] Xu Owen He. Mixture of a million experts, 2024.

[94] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022.

[95] Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL <https://arxiv.org/abs/2408.15664>.

[96] Boan Liu, Liang Ding, Li Shen, Keqin Peng, Yu Cao, Dazhao Cheng, and Dacheng Tao. Diversifying the mixture-of-experts representation for language models with orthogonal optimizer, 2024. URL <https://arxiv.org/abs/2310.09762>.

[97] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Adv. Neural Inform. Process. Syst. (NeurIPS)*, volume 36, pages 46595–46623, 2023.

[98] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-252-7. doi: 10.18653/v1/2025.acl-short.20.

[99] James E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer, New York, 2nd edition, 2007.

[100] Nicholas D Sidiropoulos and Rasmus Bro. On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3): 229–239, 2000.

[101] Donghyun Lee, Jaeyong Lee, Genghan Zhang, Mo Tiwari, and Azalia Mirhoseini. CATS: Context-aware thresholding for sparsity in large language models. In *First Conference on Language Modeling*, 2024.

[102] Frank Lauren Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6:164–189, 1927.

[103] J. Douglas Carroll and Jih Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart-Young” decomposition. *Psychometrika*, 35: 283–319, 1970.

[104] fvcore: Flop counter for PyTorch models. <https://github.com/facebookresearch/fvcore>.

[105] CodeParrot. Github code dataset. <https://huggingface.co/datasets/codeparrot/github-code>, 2022.

[106] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

[107] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand, September 13-15 2005.

[108] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* ’19, page 120–128, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255. doi: 10.1145/3287560.3287572. URL <https://doi.org/10.1145/3287560.3287572>.

[109] Anthony Duong Joseph Bloom, Curt Tigges and David Chanin. Saelens. <https://github.com/jbloomAus/SAELens>, 2024.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The quantitative results in Section 3.1 support the claims that the MxDS better navigate the sparsity-accuracy frontier in models up to 3B parameters. Section 3.2 quantitatively supports the claims that the features are similarly specialized.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are stated explicitly in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The proofs of Lemmas 1 and 2 are provided in the appendix in Appendix A.2, and referenced from the main paper. It is made explicit in Lemma 1 under exactly which technical conditions the rank property holds. A small summary of the proof of Lemma 1 is given in the main paper, too.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Yes, all details, configurations, and resources used to train the models are included in Appendix D. Implementations of the models are provided in notebooks explicitly linked to from the main paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: A link to the code to train the models is included in the submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details about the main training runs are provided in Table 10, and the details about the sparse probing experiments are provided in Appendix B.7.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The largest model runs take on the order of multiple days per value of K —thus, training multiple models is not feasible on our resource budget.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Yes, this is provided in Table 10, including the specific GPU types and wall-clock runtime.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: No violations of the code of ethics are made. In particular, the paper focuses on designing more transparent and interpretable mechanisms, and thus we don't consider the paper to cause any direct negative societal issues. However, we acknowledge that there is often an inherent dual-use nature to much interpretability work, including our own. Furthermore, whilst our largest training runs take multiple days, each uses a single GPU, therefore the environmental impact is limited.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss how our design of increasingly performant yet interpretable models may provide better understanding, debugging, and editing of pre-trained models. We do not explicitly discuss the downsides of the work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The proposed work only explores existing pre-trained models that are openly accessible.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We include links to all used models' huggingface pages in Table 10.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are released.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No human subjects are used.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human subjects are used.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [\[Yes\]](#)

Justification: We use 2 LLMs via public APIs for our steering evaluation. The full prompt is provided in Appendix [D.1](#), and the exact model endpoint names are provided for reproducibility.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

Table of Contents

| | |
|--|-----------|
| A Proofs and additional technical results | 25 |
| A.1 Proof of rank equality | 25 |
| A.2 Proof of MxD forward pass equivalence | 26 |
| A.3 Intuition for weight parameterization through the lens of tensor methods | 26 |
| A.4 GLU encoders are a mixture of rank-1 linear experts | 27 |
| A.5 Hadamard-factorized tensors generalize MoVs | 28 |
| A.6 TopK activation function | 28 |
| B Additional quantitative results and ablations | 29 |
| B.1 Faithfulness in output space | 29 |
| B.2 Additional reconstruction metrics | 30 |
| B.3 Results on additional layers | 30 |
| B.4 Initial results on Gemma2-27B | 31 |
| B.5 Expert rank | 32 |
| B.6 Computational cost of sparse layers | 34 |
| B.7 Sparse probing | 34 |
| B.8 Ablations | 35 |
| C Feature balance and shared experts | 40 |
| C.1 Expert/feature balance | 40 |
| C.2 Shared experts | 42 |
| D Detailed experimental setup | 42 |
| D.1 Feature steering details | 44 |
| E Additional qualitative results | 44 |

A Proofs and additional technical results

A.1 Proof of rank equality

Proof of Lemma 1. We first derive the expression for expert n 's weight matrix $\mathbf{W}_n = \mathbf{D} \operatorname{diag}(\mathbf{c}_n) \in \mathbb{R}^{H \times O}$ and then show the rank equality that follows. First, recall that we have the third-order weight tensor defined as

$$\mathcal{W}(n, h, :) = \mathbf{c}_n * \mathbf{d}_h \in \mathbb{R}^O,$$

for matrices $\mathbf{C} \in \mathbb{R}^{N \times O}$, $\mathbf{D} \in \mathbb{R}^{H \times O}$. We can express each element of the tensor $\mathcal{W} \in \mathbb{R}^{N \times H \times O}$ in terms of elements of the two matrices as

$$\mathcal{W}(n, h, o) = c_{no} \cdot d_{ho} = (\mathbf{D})_{ho} \cdot c_{no}. \quad (6)$$

Equation (6) shows that for a fixed expert n , the n^{th} row $\mathbf{c}_n \in \mathbb{R}^O$ essentially scales the columns of matrix $\mathbf{D} \in \mathbb{R}^{H \times O}$. This is equivalent to right-multiplying matrix \mathbf{D} by a diagonal matrix formed from $\mathbf{c}_n \in \mathbb{R}^O$. Indeed, the (h, o) entry of such matrix product is

$$[\mathbf{D} \operatorname{diag}(\mathbf{c}_n)]_{ho} = \sum_{i=1}^O (\mathbf{D})_{hi} \operatorname{diag}(\mathbf{c}_n)_{io} \quad (7)$$

$$= (\mathbf{D})_{ho} \operatorname{diag}(\mathbf{c}_n)_{oo} \quad (8)$$

$$= d_{ho} \cdot c_{no}, \quad (9)$$

since all off-diagonal terms (i.e., $i \neq o$) in Equation (7) vanish and $\text{diag}(\mathbf{c}_n)_{oo} = c_{no}$ by construction. Comparing Equation (6) and Equation (9) shows that, for every $h \in \{1, 2, \dots, H\}$ and $o \in \{1, 2, \dots, O\}$ we have

$$\mathbf{W}(n, h, o) = [\mathbf{D} \text{diag}(\mathbf{c}_n)]_{ho}.$$

Hence, indexing into the first mode of the tensor alone gives us the matrix-valued expression for expert n as claimed:

$$\mathbf{W}(n, :, :) = \mathbf{W}_n = \mathbf{D} \text{diag}(\mathbf{c}_n) \in \mathbb{R}^{H \times O}.$$

Finally, a standard result in linear algebra [99] has that $\text{rank}(\mathbf{AB}) = \text{rank}(\mathbf{A})$ for any $\mathbf{A} \in \mathbb{R}^{H \times O}$ and invertible matrix $\mathbf{B} \in \mathbb{R}^{O \times O}$. Since matrix $\text{diag}(\mathbf{c}_n) \in \mathbb{R}^{O \times O}$ is invertible by assumption in Lemma 1, setting $\mathbf{A} = \mathbf{D}$ and $\mathbf{B} = \text{diag}(\mathbf{c}_n)$ yields the rank equality. \square

A.2 Proof of MxD forward pass equivalence

Recall we have input vector $\mathbf{z} \in \mathbb{R}^H$, expert coefficients $\mathbf{a} \in \mathbb{R}^N$, and layer weights $\mathbf{W} \in \mathbb{R}^{N \times H \times O}$. The weights are defined in Equation (4) element-wise through the Hadamard product $*$ as

$$\mathbf{W}(n, h, :) = \mathbf{c}_n * \mathbf{d}_h \in \mathbb{R}^O, \quad \forall n \in \{1, \dots, N\}, h \in \{1, \dots, H\},$$

for learnable parameters $\mathbf{C} \in \mathbb{R}^{N \times O}$, $\mathbf{D} \in \mathbb{R}^{H \times O}$. Lemma 2 states that MxD’s forward pass can be equivalently expressed as

$$\sum_{n=1}^N a_n (\mathbf{W}_n^\top \mathbf{z}) = (\mathbf{C}^\top \mathbf{a}) * (\mathbf{D}^\top \mathbf{z}).$$

Proof of Lemma 2. The LHS can first be re-written as an explicit sum over the hidden dimension

$$\hat{\mathbf{y}} = \sum_{n=1}^N a_n (\mathbf{W}_n^\top \mathbf{z}) = \sum_{n=1}^N \sum_{h=1}^H a_n (\mathbf{w}_{nh} z_h) \in \mathbb{R}^O. \quad (10)$$

Plugging in the definition of $\mathbf{w}_{nh} \in \mathbb{R}^O$ from Equation (4) then yields

$$\hat{\mathbf{y}} = \sum_{n=1}^N \sum_{h=1}^H a_n (\mathbf{w}_{nh} z_h) \quad (11)$$

$$= \sum_{n=1}^N \sum_{h=1}^H a_n ((\mathbf{c}_n * \mathbf{d}_h) z_h) \quad (12)$$

$$= \left(\sum_{n=1}^N a_n \mathbf{c}_n \right) * \left(\sum_{h=1}^H z_h \mathbf{d}_h \right) \quad (13)$$

$$= (\mathbf{C}^\top \mathbf{a}) * (\mathbf{D}^\top \mathbf{z}), \quad (14)$$

which is exactly the RHS of Equation (5), showing the MxD forward pass is equivalent to the Hadamard product of $\mathbf{C}^\top \mathbf{a}$ and $\mathbf{D}^\top \mathbf{z}$. \square

A.3 Intuition for weight parameterization through the lens of tensor methods

A second complementary way of viewing the MxD layer’s parameterization (and its full-rank properties) is through the lens of tensor methods [34]. A tensor-based motivation for MxD’s weight tensor parameterization and forward pass is presented in Appendix A.3.1 and Appendix A.3.2, respectively.

Notation and definitions A brief primer is first included below, based on [34] (and can be safely skipped for those already familiar):

- The **mode- n fibers** of an N^{th} order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ are the I_n -dimensional column vectors obtained by fixing every index except that of the n^{th} mode (e.g., $\mathbf{x}_{i_2 i_3} \in \mathbb{R}^{I_1}$ are the mode-1 fibers of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$). Stacking all mode- n fibers column-wise yields the so-called **mode- n unfolding** $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \bar{I}_n}$, with number of columns given by the product of remaining dimensions $\bar{I}_n = \prod_{\substack{t=1 \\ t \neq n}}^N I_t$.
- The **Khatri-Rao product** (denoted by \odot) between two matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, is the column-wise Kronecker product (denoted by \otimes):

$$\mathbf{A} \odot \mathbf{B} := [\mathbf{a}_{:1} \otimes \mathbf{b}_{:1} \ \dots \ \mathbf{a}_{:K} \otimes \mathbf{b}_{:K}] \in \mathbb{R}^{(I \cdot J) \times K}$$
- The **mode- n (vector) product** of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with a vector $\mathbf{u} \in \mathbb{R}^{I_n}$ is denoted $\mathcal{X} \times_n \mathbf{u}$ and has entries $(\mathcal{X} \times_n \mathbf{u})_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} u_{i_n}$.

A.3.1 MxD weight tensors through the Khatri-Rao product

MxDS construct the collective weight tensor through the Khatri-Rao product \odot [34] of the two factor matrices $\mathbf{C} \in \mathbb{R}^{N \times O}$, $\mathbf{D} \in \mathbb{R}^{H \times O}$. Concretely, the mode-3 unfolding³ of the third-order weight tensor $\mathcal{W} \in \mathbb{R}^{N \times H \times O}$ in MxDS from Equation (4) is alternatively given by:

$$\mathbf{W}_{(3)} := (\mathbf{C} \odot \mathbf{D})^\top \in \mathbb{R}^{O \times (N \cdot H)}. \quad (15)$$

Given that the factor matrices are learned end-to-end without constraints, they are likely of full column-rank, i.e. $\text{rank}(\mathbf{D}) = \text{rank}(\mathbf{C}) = O$ (as $N > O$, $H = 4 \cdot O > O$ in practice given the MLP layers' larger bottleneck). Consequently, their Khatri-Rao product parameterizing the collective N experts' weights will be of maximum rank O too, through Lemma 1 of [100]. As a result, parameterized this way, the O -dimensional fibers likely span the full output space.

A.3.2 Tensorized MxD forward pass

Furthermore, the layer's forward pass can then be viewed as performing two tensor contractions between the third-order weight tensor $\mathcal{W} \in \mathbb{R}^{N \times H \times O}$ (collecting all N experts' $H \times O$ -dimensional matrices) and expert coefficients $\mathbf{a} \in \mathbb{R}^N$ and hidden activations $\mathbf{z} \in \mathbb{R}^H$. This can be expressed in terms of the so-called mode- n product (denoted by \times_n) [34] as follows:

$$\begin{aligned} \hat{\mathbf{y}} &= \sum_{n=1}^N a_n \cdot (\mathbf{W}_n^\top \mathbf{z}) \\ &= \sum_{n=1}^N a_n \sum_{h=1}^H \mathbf{w}_{nh} z_h = \sum_{n=1}^N \sum_{h=1}^H a_n z_h \mathbf{w}_{nh} \\ &= \mathcal{W} \times_1 \mathbf{a} \times_2 \mathbf{z} \in \mathbb{R}^O. \end{aligned} \quad (16)$$

A.4 GLU encoders are a mixture of rank-1 linear experts

Both the proposed MxDS and Gated Linear Units (GLUs) [33] share a similar functional form, using the element-wise product. However, there are crucially important differences between GLUs and MxDS that make both their interpretation and model capacity different.

In short, the technical results here in our paper show that GLUs' encoder can be viewed as a linear mixture of expert layer with rank-1 experts. Furthermore, GLUs can be modified and extended to MxDS with two additions to their model form as detailed at the end of this subsection. First, recall that the GLU encoder [33] computes:

$$\mathbf{y}_{\text{GLU}} = \psi(\mathbf{E}_{\text{GLU}}^\top \mathbf{x}) * (\mathbf{E}^\top \mathbf{x}) \in \mathbb{R}^H, \quad (17)$$

for input vector $\mathbf{x} \in \mathbb{R}^I$, learnable weights $\mathbf{E}_{\text{GLU}}, \mathbf{E} \in \mathbb{R}^{I \times H}$, and activation function $\psi(\cdot)$. To transform Equation (17) into the same model form as MxDS, we first pre-multiply the LHS by the identity matrix to match the MxD model form of Equation (5), yielding:

$$\mathbf{y}_{\text{GLU}} = (\mathbb{I}^\top \mathbf{a}) * (\mathbf{E}^\top \mathbf{x}), \quad (18)$$

³which is simply a reshaping of a higher-order tensor into a matrix, arranging all N expert matrices' column vectors along the columns of a new matrix.

where $\mathbf{a} = \psi(\mathbf{E}_{\text{GLU}}^\top \mathbf{x}) \in \mathbb{R}^H$ and $\mathbb{I} \in \mathbb{R}^{H \times H}$ is the H -dimensional identity matrix. Next, we can write this explicitly in terms of a linear MoE with expert weights $\mathbf{W}_n \in \mathbb{R}^{I \times H}$ as follows:

$$\mathbf{y}_{\text{GLU}} = (\mathbb{I}^\top \mathbf{a}) * (\mathbf{E}^\top \mathbf{x}) \quad (19)$$

$$= \sum_{n=1}^H a_n (\mathbf{W}_n^\top \mathbf{x}) \quad (20)$$

$$= \sum_{n=1}^H a_n (\mathbf{E} \text{diag}((\mathbb{I})_n))^\top \mathbf{x}, \quad (21)$$

where $(\mathbb{I})_n \in \mathbb{R}^H$ is the n^{th} row of the H -dimensional identity matrix (i.e. a one-hot vector with its only non-zero element at index n). We draw particular attention to how the n^{th} expert’s matrix $\mathbf{W}_n = \mathbf{E} \text{diag}((\mathbb{I})_n) \in \mathbb{R}^{I \times H}$ essentially picks out the n^{th} column of \mathbf{E} , leaving all remaining $H - 1$ columns as zero vectors. **Therefore, GLU encoders compute a MoE with linear expert weights of (at most) rank 1.** This relationship between GLUs and conditional computation is consistent with prior work interpreting individual GLU column vectors as experts [101]. Whilst GLUs’ encoders’ model form does not put any inherent restrictions on the total number of rank-1 terms that can contribute to the output, the sparsity necessary for specialization does.

We conclude this section by summarizing the two technical changes needed to transform GLUs into full-rank linear MoEs based on the Hadamard product:

1. Replace \mathbb{I} in Equation (18) with learnable, non-diagonal weight matrices for full-rankness.
2. Choose $\psi(\cdot)$ to produce non-negative, sparse coefficients to encourage specialization through sparsity among the experts (for example, a softmax function, or a ReLU activation followed by TopK).

The first of the steps above provides full-rankness, whilst the second brings the sparsity and non-negativity needed for specialization. We include a notebook showing this connection in PyTorch at: <https://github.com/james-oldfield/MxD/blob/main/glus-to-moes.ipynb>.

A.5 Hadamard-factorized tensors generalize MoVs

Prior work [37] proposes to linearly combine N many (IA)³ adapters [38] for parameter-efficient MoEs for instruction fine-tuning. The implementation results in a very similar functional form to the factorized forward-pass in MxDS. Interestingly, the Hadamard product parameterization of the third-order weight tensor in Equation (4) provides a more general framework through which one can also derive MoVs’ model form, shedding light on the relationship to the proposed MxDS and their benefits. Concretely, factorizing the weight tensor instead along the *second* mode as $\mathbf{W}(n, :, o) = \mathbf{c}_n * \mathbf{d}_o \in \mathbb{R}^H$ in our framework immediately recovers MoV [37] as a special case. In particular, in contrast to the MxD in Appendix A.3 whose weight tensor can be parametrized equivalently through its mode-3 unfolding [34], MoV’s implicit weight tensor can be given in terms of its mode-2 unfolding in terms of a similar Khatri-Rao product of two factor matrices.

Instead, MoVs in analogy would yield expert weights by pre-multiplying \mathbf{D} as: $\mathbf{W}_n = \text{diag}(\mathbf{c}_n) \mathbf{D} \in \mathbb{R}^{H \times O}$ for much larger $\mathbf{C} \in \mathbb{R}^{N \times H}$. Due to $H \gg O$, **our proposed MxD formulation yields around 4× the number of specialized units as MoVs** with the same parameter budget (yet MoVs’ experts are of no higher rank than MxDS’), making MxDS a much more suitable and efficient class of layer for our goal of scalable specialization. We therefore see that the proposed lens of tensor methods for unification provides valuable insights about how to design more interpretable layers with the minimum trade-off to capabilities.

A.6 TopK activation function

Formally, for activations $\mathbf{z} \in \mathbb{R}^H$ (possibly with the ReLU activation applied), the TopK activation function can be formulated elementwise as the following:

$$\text{TopK}(\mathbf{z})_h = \begin{cases} z_h, & \text{if } z_h \geq \tau_K(\mathbf{z}) \\ 0, & \text{otherwise} \end{cases},$$

where $\tau_K(\mathbf{z}) \in \mathbb{R}$ returns the value of the K^{th} largest element of vector $\mathbf{z} \in \mathbb{R}^H$. Intuitively, the TopK activation retains only the largest $K \leq H$ elements of the H neurons, setting the rest to zero. We note here that the definition above does not handle ties (although this is very unlikely in practice, given the activations are continuous).

B Additional quantitative results and ablations

B.1 Faithfulness in output space

Experiments in Section 3.1.2 of the main paper report the number of future predicted words that are identical from the original model and from the LLM with the sparse layer replacements. We also show qualitative examples of the first 8 prompts and the subsequent ‘diffs’ (using Python 3’s `difflib`) of the generated tokens in Figures 7 and 8—we see MxDs’ superior ability to preserve model functionality as it propagates through to the output space of future tokens.

Generation 1, Pythia-410m

GT: There are times when you need to take a break from your daily routine and just
 MxD: There are times when you need to take a break from your daily routine and just
 STC: There are times when you need to be able to do something that is not easy.
 TC: There are times when you need to take a break from your daily routine and just
 SAE: There are times when you need to be in the air, but not in a high-speed

Generation 2, Pythia-410m

GT: Humanitarian chief warns capacity of US to be tested By Staff reports Published: Friday, May 30,
 MxD: Humanitarian chief warns capacity of US to be tested By Staff reports Published: Friday, May 30,
 STC: Humanitarian chief warns capacity of US to be tested By Associated Press Published: Tuesday, March 29,
 TC: Humanitarian chief warns capacity of US to be tested By Associated Press | January 24, 2013
 SAE: Humanitarian chief warns capacity of the U.S. military to be a threat The U.S. military is

Generation 3, Pythia-410m

GT: During the Trump administration, the Department of Homeland Security has been tasked with overseeing immigration enforcement.
 MxD: During the Trump administration, the Department of Homeland Security (DHS) has been tasked with protecting Americans
 STC: During the Trump administration, the Department of Justice has been accused of using “bribery” to influence
 TC: During the Trump administration, the Department of Homeland Security has been tasked with overseeing immigration enforcement.
 SAE: During the Trump administration, the president-elect’s campaign was a “crisis” that could be solved by a

Generation 4, Pythia-410m

GT: Romanian newspaper ZF.ro cites a report from the Ministry of Interior and Security (MIS) that shows
 MxD: Romanian newspaper ZF.ro cites a report from the Ministry of Interior and Security (MIS) that shows
 STC: Romanian newspaper ZF.ro cites a report by the European Commission that shows that Romania is not
 TC: Romanian newspaper ZF.ro cites a report from the Ministry of Foreign Affairs and Trade (MFA) that
 SAE: Romanian newspaper ZF.ro cites a report that the country’s police officers are not allowed to use

Generation 5, Pythia-410m

GT: Democratic Virginia Gov. Terry McAuliffe (D) on Tuesday said he will not seek re-election in 2020,
 MxD: Democratic Virginia Gov. Terry McAuliffe (D) on Tuesday said he will not seek re-election in 2020,
 STC: Democratic Virginia Gov. Terry McAuliffe (D) is running for president in the 2020 election, but he’s
 TC: Democratic Virginia Gov. Terry McAuliffe (D) is facing a challenge from a group of Democratic state
 SAE: Democratic Virginia Gov. Terry McAuliffe (D) is a Democrat, but he’s not the most popular governor

Generation 6, Pythia-410m

GT: A federal court has ruled that the Trump administration’s travel ban on people from seven Muslim-majority
 MxD: A federal court has ruled that the Trump administration’s travel ban on people from seven Muslim-majority
 STC: A federal court has ruled that the Trump administration’s travel ban on people from seven Muslim-majority
 TC: A federal court has ruled that the Trump administration’s travel ban on people from seven Muslim-majority
 SAE: A federal court has ordered a former U.S. ambassador to the United Nations, William H. Taylor,

Generation 7, Pythia-410m

GT: British Columbia takes in \\$1.5 billion from the sale of its oil and gas reserves The
 MxD: British Columbia takes in \\$1.5 billion from the sale of its oil and gas reserves The
 STC: British Columbia takes in \\$1.5 billion in tax breaks The British Columbia government has announced that
 TC: British Columbia takes in the world The British Columbia government has announced that it will spend
 SAE: British Columbia takes in the third year of a plan to expand its coal-mining operations, but

Generation 8, Pythia-410m

GT: Earlier this month, I wrote about the upcoming release of the first episode of The Walking
 MxD: Earlier this month, I wrote about the upcoming release of the first episode of The Walking
 STC: Earlier this month, I wrote about the upcoming release of the “Sonic Boom” soundtrack. The album
 TC: Earlier this month, I wrote about the upcoming release of the “Sonic Mania” game. The game
 SAE: Earlier this month, I was invited to a conference in the city of Toronto. The event

Figure 7: **Pythia-410m:** The first few generated tokens from the base model (‘GT’) and the corresponding tokens from the model when the sparse layers replace the target MLP. **Red** denotes tokens that are removed, **orange** denotes newly inserted tokens, and **green** denotes matching tokens.

Generation 1, GPT2-124m

GT: There are times when you need to be a little more careful with your food. You
MxD: There are times when you need to be on the lookout for a new job. But,
STC: There are times when you need to get out of bed and go to the bathroom.
TC: There are times when I feel like I'm being judged. I've been told that my grades
SAE: There are times when you need to get a little bit more than your usual. You're

Generation 2, GPT2-124m

GT: Humanitarian chief warns capacity to handle refugees is 'unprecedented' The UN refugee agency has warned that
MxD: Humanitarian chief warns capacity to handle refugees 'is at risk' The UN refugee agency has warned
STC: Humanitarian chief warns capacity to handle emergencies is at risk The UK government has warned that
TC: Humanitarian chief warns capacity to handle emergencies is at risk The UK government has warned that
SAE: Humanitarian chief warns capacity to hold up The BBC's political correspondent, Peter Robinson, says the government

Generation 3, GPT2-124m

GT: During the Trump administration, the White House has been working to make sure that its immigration
MxD: During the Trump administration, the White House has been working to make sure that its immigration
STC: During the Trump administration, the White House has been working to make sure that it is
TC: During the Trump administration, the White House has been working to make sure that no one
SAE: During the Trump administration, the White House has been working on a plan to build a

Generation 4, GPT2-124m

GT: Romanian newspaper ZF.ro cites a report by the European Commission that the country's government is considering
MxD: Romanian newspaper ZF.ro cites a report by the European Commission that the country's government is considering
STC: Romanian newspaper ZF.ro cites a report by the European Commission that the country's government is considering
TC: Romanian newspaper ZF.ro cites a report by the European Commission that the EU is considering imposing
SAE: Romanian newspaper ZF.ro cites the "revelation" of the "death of the country's economy" as a key

Generation 5, GPT2-124m

GT: Democratic Virginia Gov. Terry McAuliffe (D) said he would not seek re-election in 2018, but he
MxD: Democratic Virginia Gov. Terry McAuliffe (D) said he would not seek re-election in 2018, but he
STC: Democratic Virginia Gov. Terry McAuliffe (D) has said he will not support a bill that would
TC: Democratic Virginia Gov. Terry McAuliffe (R) has said he will not seek re-election in 2018, but
SAE: Democratic Virginia Gov. Terry McAuliffe (R) has signed a bill that would allow the state to

Generation 6, GPT2-124m

GT: A federal court has ordered the Department of Justice to pay \\$1.5 million to a group
MxD: A federal court has ordered the Department of Justice to pay \\$1.5 million to a group
STC: A federal court has ordered the Department of Justice to stop using a search warrant to
TC: A federal court has ordered the Department of Justice to stop using a search warrant to
SAE: A federal court has ordered the Department of Justice to pay \\$1.5 million to a former

Generation 7, GPT2-124m

GT: British Columbia takes in \\$1.5 billion annually from the federal government, but only about half of
MxD: British Columbia takes in \\$1.5 billion in foreign aid annually, according to the government's latest report
STC: British Columbia takes in \\$1.5 billion in foreign aid annually, according to the Canadian Council of
TC: British Columbia takes in \\$1.5 billion annually from the federal government, but it's not a big
SAE: British Columbia takes in \\$1.5 million in revenue from the province's tourism industry, according to a

Generation 8, GPT2-124m

GT: Earlier this month, I wrote about the state of our industry. We've been in a bit
MxD: Earlier this month, I wrote about the state of our industry. We've been in a state
STC: Earlier this month, I wrote about the state of the art in building a solid foundation
TC: Earlier this month, I wrote about the state of the art for creating a custom 3D
SAE: Earlier this month, I was asked to write a piece about the "anti-Semitic" movement in Germany.

Figure 8: **GPT2-124m:** The first few generated tokens from the base model ('GT') and the corresponding tokens from the model when the sparse layers replace the target MLP. **Red** denotes tokens that are removed, **orange** denotes newly inserted tokens, and **green** denotes matching tokens.

B.2 Additional reconstruction metrics

To highlight the scale of difference in the reconstructions between MxDS and the baselines, we also plot in Figure 9 the normalized MSE at the end of training for all models and LLMs. At the smallest values of K (which we care about most for interpretability), **MxDS' normalized MSE is up to an order of magnitude smaller than Transcoders**'.

B.3 Results on additional layers

We also fully train all models and baselines (with 4 different values of K) on different target layers for each model. The results are shown in Figure 10 for 48 additional trained layers for the same setup in the original paper, using different colours to highlight that these are new results. As can be seen, the same trend holds: MxDS significantly outperform the baselines at small K in all LLMs.

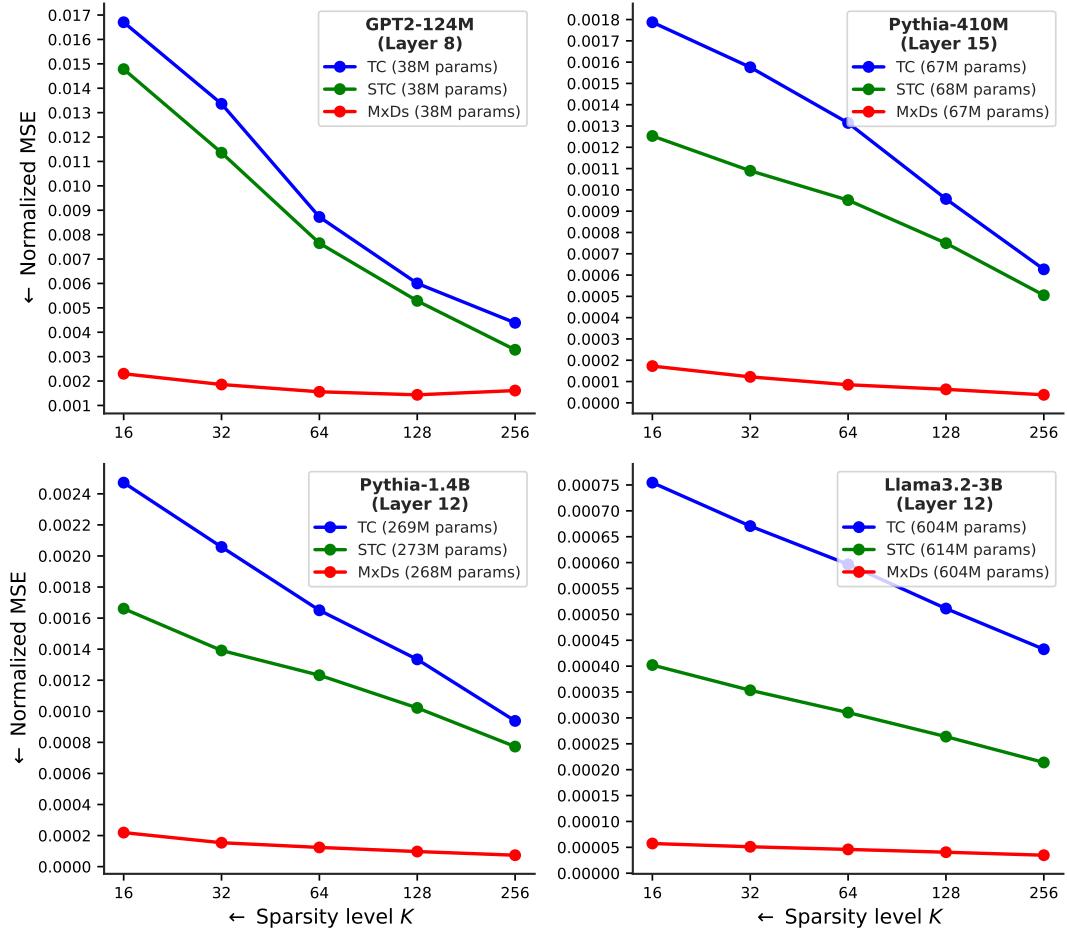


Figure 9: **Normalized MSE** at the end of training Sparse MLP layers, as a function of the number of active units (i.e., hidden neurons vs experts); with differences as large as an order of magnitude in error.

B.4 Initial results on Gemma2-27B

In the main paper, we perform experiments on base models with up to 3B parameters. For initial evidence of scalability to even larger models, we perform here additional large-scale experiments on the 27B Gemma2 model (in half-precision, and with a smaller batch size to fit into memory). We see MxDs continue to outperform on the sparsity-accuracy frontier, as shown by the (normalized) reconstruction loss in Table 3.

Table 3: **Normalized MSE** (\downarrow) on Gemma2-27B, Layer 20, $K = 32$, partially trained for 50k iterations. To fit in memory, the model is loaded in half precision. We use 1/8 the batch size and # buffers stored, and 1/2 the multiplier on the number of latent features (with values of 4, 16, and 16, respectively).

| Iterations | 10k | 20k | 30k | 40k | 50k |
|-----------------|--------------|--------------|--------------|--------------|--------------|
| Transcoder | 0.147 | 0.132 | 0.128 | 0.123 | 0.119 |
| Skip Transcoder | 0.128 | 0.107 | 0.102 | 0.100 | 0.093 |
| MxD | 0.098 | 0.096 | 0.093 | 0.086 | 0.069 |

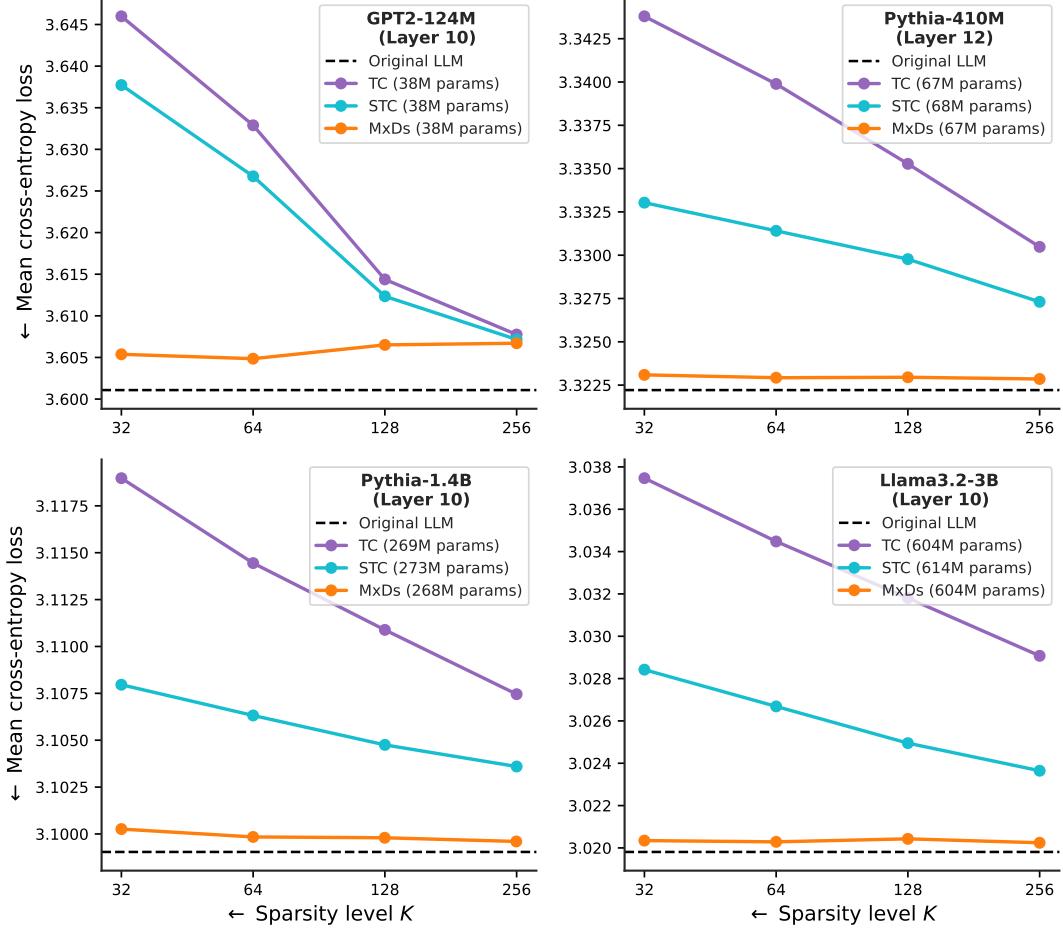


Figure 10: **Additional layer results:** model cross entropy loss preserved when replacing MLPs with Transcoders [27], Skip Transcoders [26], and MxDs, as a function of the number of active units (hidden neurons/experts). These results complement those in the main paper, but here we train a new set of additional models on different layers.

B.5 Expert rank

This section concerns the matrix rank of the linear experts in parameter-efficient MoEs. We first compare to low-rank MoEs in Appendix B.5.1 to demonstrate the benefits of full-rankness, and then follow up in Appendix B.5.2 by confirming that the learned MxD expert ranks are close to maximum in the trained models.

B.5.1 Comparisons to low-rank MoEs

In this section, we study the impact of expert rank on the ability of efficient MoE layers to reconstruct pre-trained MLP layers’ mappings. One compelling alternative to MxDs for efficient conditional computation is the μ MoE layer [40], which imposes low-rankness on expert weights to achieve parameter-efficiency. Whilst μ MoEs are found to perform competitively in the pre-training setting, the impact of low-rankness on approximations of *existing* layers will determine their suitability in the sparse layer approximation setting studied in this work.

We therefore compare to μ MoE layers, which we use to compute a linear MoE in place of the MLP’s decoder. In CP μ MoEs, N experts’ weight matrices are jointly parameterized through low-rank tensor structure with the CP decomposition [102, 103] for chosen rank $R \in \mathbb{N}^+$. With the same learnable encoder and expert gating matrices producing the expert coefficients $\mathbf{a} \in \mathbb{R}^N$ and hidden units $\mathbf{z} \in \mathbb{R}^H$ generated the same way as in the main paper, we train μ MoE layers to approximate the

original MLP layer’s output with:

$$\mu\text{MoE}(\mathbf{x}) = \sum_{n=1}^N \sum_{h=1}^H \sum_{r=1}^R a_n z_h \mathbf{D}(r, h) \cdot \mathbf{C}(r, n) \cdot \mathbf{W}(:, r) \in \mathbb{R}^O, \quad (22)$$

where $\mathbf{C} \in \mathbb{R}^{R \times N}$, $\mathbf{D} \in \mathbb{R}^{R \times H}$, $\mathbf{W} \in \mathbb{R}^{O \times R}$ are the learnable low-rank terms of the implicit third-order tensor parameterizing all N collective experts’ weights.

We match the MxD experimental configuration as closely as possible for a fair comparison. For the encoders, we mirror MxDs and use the GELU activation function, which we find through ablations in Appendix B.8 to perform the best. We initialize the parameters the same as MxDs and Skip Transcoders: we use the standard PyTorch linear layer initialization for \mathbf{D} , \mathbf{C} (and the encoder layers), and initialize \mathbf{W} as the zero matrix.

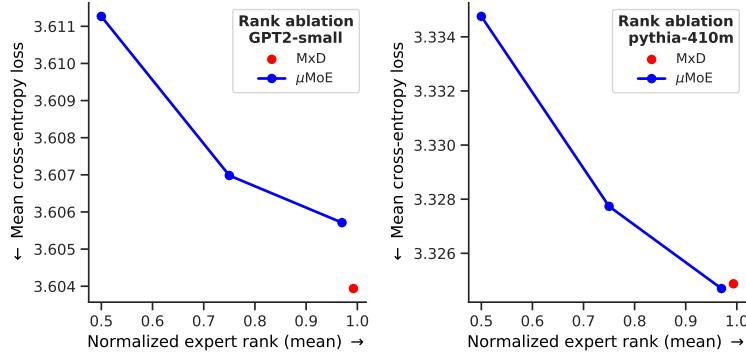


Figure 11: Comparisons to μ MoEs for various choices of (normalized) rank: high rank weights best-preserve the models’ downstream cross-entropy loss.

We vary the μ MoE layer rank R , training fully 3 sparse approximation layers for $K = 32$ active experts, varying the total number of experts N to keep the parameter count the same—isolating the impact of the choice of rank. As with the main experiments, we record the downstream model loss when we splice in the trained layer to replace the MLP layers, shown in Figure 11.

As can be seen, the μ MoE layers perform well when they are close to full-rank (i.e. when the normalized rank $\frac{R}{O} \rightarrow 1$). Crucially, however, performance drops off notably when the rank is reduced. Whilst μ MoEs still perform far better than neuron-level sparsity methods (i.e. the corresponding CE loss results in Figure 3), we observe that full-rankness is necessary for the most faithful layer approximations—which the proposed MxDs provide by design.

As a motivating example, for why SparseMoEs and SoftMoEs are not practical: SparseMoEs [36] and SoftMoEs [72] require 2.16 **trillion** parameters for a single layer, for the same 86k experts we use for Llama-3.2-3B. This is orders of magnitude more parameters than the entire base network itself, making it prohibitively costly for SparseMoEs to scale to sufficiently high expert counts.

B.5.2 MxD empirical expert rank

Next, we show experimentally that the learned experts’ matrices $\mathbf{W}_n = \mathbf{D} \text{ diag}(\mathbf{c}_n) \in \mathbb{R}^{H \times O}$ are very nearly full-rank in practice, corroborating the properties of expert matrices shown theoretically in Lemma 1. We compute the mean ‘normalized rank’, which we take for MxDs to be the empirical matrix rank of the learned expert’s weights, over the maximum possible rank given the dimensions:

$$\frac{1}{N} \sum_{n=1}^N \frac{\text{rank}(\mathbf{W}_n)}{\min\{H, O\}}. \quad (23)$$

We show in Table 4 the normalized rank across all 4 base models: MxD’s learned experts exhibit no rank deficiencies, providing further evidence of the large potential capacity of MxD layers despite their sparsity constraints on the expert-level.

Table 4: Mean normalized expert matrix rank of Equation (23) across models for the first 2k experts in $K = 32$ trained MxDs – the learned expert matrices are very close to full column rank.

| GPT2-124M | Pythia-410M | Pythia-1.4B | Llama-3.2-3B |
|------------------|------------------|------------------|------------------|
| 0.99 ± 0.005 | 0.99 ± 0.007 | 0.99 ± 0.005 | 0.99 ± 0.002 |

B.6 Computational cost of sparse layers

We show here that there is minimal difference between the computational cost of sparse layer variants. We first report theoretical layer FLOPs, and then report empirical benchmarks.

Parameter count & FLOPs We first tabulate in Table 5 the theoretical parameter counts and inference-time FLOPs for MxDs vs Sparse MLPs. To be consistent with the popular PyTorch library [104] we count one fused multiply-add as one FLOP, and count the Hadamard product between two d -dimensional vectors as requiring $d/2$ FLOPs. Note that, For a chosen expert count N , we set the width of Sparse MLPs (e.g. Transcoders) to $H := N + H^*$ to parameter-match the models.

Table 5: Here I, O denotes the input and output dimensions, H^* the original model’s hidden width, N the MxD expert count, and H the width of the sparse MLPs.

| | Parameter count | FLOPs |
|-------------------|-------------------------|-------------------------------|
| Sparse MLP | $H(I + O)$ | $H(I + O)$ |
| MxD | $N(I + O) + H^*(I + O)$ | $N(I + O) + H^*(I + O) + O/2$ |

Empirical benchmarks We next run benchmarks for a Sparse MLP layer vs MxD with a batch size of 512, and dimensions: $I = H^* = O = 1024$, and number of experts/features as $N = 8192$, $H = 9216$. The results are tabulated in Table 6, where we see performance (and cost) is similar for the two layers.

Table 6: Here I, O denotes the input and output dimensions, H^* the original model’s hidden width, N the MxD expert count, and H the width of the sparse MLPs.

| | Peak memory usage (MiB) | Latency (ms) | Parameter count | Reported FLOPs ([104]) |
|-------------------|-------------------------|--------------|-----------------|------------------------|
| Sparse MLP | 386.50 | 1.394 | 18,874,368 | 18,874,368 |
| MxD | 389.50 | 1.457 | 18,874,368 | 18,874,880 |

B.7 Sparse probing

Sample-level probing Here, we follow the SAE Bench [19] evaluation protocol. In this ‘sample-level’ setting, each text string is labeled with a binary concept at a global level (e.g., the language of the snippet, or its sentiment). This is in contrast to what we refer to as ‘token-level probing’, where *each token* within the text samples is labeled individually (e.g., whether a word is a certain part of speech). We perform experiments on a total of 24 sample-level sparse probing tasks with the same ‘maximum mean difference’ feature filtering applied in [19]. The details of the datasets used are summarized in Table 7.

Token-level probing We also explore sparse probing for 10 features defined at the token-level. For this, we follow [49], and include experiments training probes on the mean feature activations under tokens spanning the **surnames** of the individuals. We note that this is a significantly harder task, and makes even stronger assumptions about the features the dataset includes, but is nonetheless some additional weak evidence about the relative feature-learning abilities of the sparse models. Through various surnames, we probe for 6 occupations of individuals, whether or not individuals are alive, and individuals’ labeled gender. We also experimented with probing for compound words as in [49], but found no predictive features in our trained models. Details of the surname token-level probing datasets (and the total training examples the tokenizers could parse) are included in Table 8.

Table 7: Details of sample-level sparse probing datasets used.

| Dataset | # Training examples | # Test examples | Classification task description | Number of classes |
|--|---------------------|-----------------|------------------------------------|-------------------|
| fancyzhx/ag_news [48] | 16,000 | 4,000 | News article topic | 4 |
| codeparrot/github-code [105] | 20,000 | 5,000 | Programming language | 5 |
| amazon_reviews_mcauley_1and5_sentiment [106] | 8,000 | 2,000 | Positive/negative review sentiment | 2 |
| Helsinki-NLP/europarl [107] | 20,000 | 5,000 | European language | 5 |
| LabHC/bias_in_bios [108] | 32,000 | 8,000 | Profession from bio | 8 |

Table 8: Details of token-level sparse probing datasets used.

| Dataset | # Training examples | # Test examples | Classification task description | Number of classes |
|-----------------|---------------------|-----------------|---------------------------------|-------------------|
| Occupation [49] | 4784 | 1195 | Occupation of individual | 6 |
| Is alive? [49] | 4800 | 1199 | Are they alive | 2 |
| Gender [49] | 4800 | 1200 | Labeled gender | 2 |

Experimental setup For sample-level probing, we truncate the input strings to the first 128 tokens for all datasets but for the Github dataset, where we take the last 128 tokens to avoid license headers [19, 49]. For token-level probing, we instead take only the last 128 tokens, where the final token contains the surname of the individual in question in the datasets of [49].

Binary probes are trained on 80% of the training data (randomly shuffled) with the `sklearn` library’s `LogisticRegression` module with parameters:

- `class_weight='balanced'`
- `penalty='l2'`
- `solver='newton-cholesky'`
- `max_iter=200`

A random seed of 42 is used throughout the code to ensure reproducibility.

B.7.1 Sparse probing results

We show in Figure 12 results on 20 additional (sample-level) sparse probing tasks, where MxDs remain competitive with the baselines. We also plot the expert activation (of the single expert with the highest F1 test set score) for the positive/negative classes for all tasks split across Figures 13 and 14. One can observe a certain degree of separability between the two semantic clusters of data given by the expert coefficient, thus confirming that individual experts are learning to specialize to particular high-level features.

We also include results on 10 token-level probing tasks in Figure 15, with the corresponding activation densities displayed in Figure 16. Whilst MxDs appear to perform slightly less well here on average, they remain competitive as expected.

B.8 Ablations

We turn next to ablation studies to explore the value of the various model components below:

B.8.1 Choice of sparsity constraint

We first train a variety of MxDs on GPT2 models with the TopK activation function [23] and instead train models with a ReLU followed by an explicit $\lambda \|\cdot\|_1$ sparsity penalty on the specialized components in addition to the reconstruction loss [22]. We show the results in Figure 17, where, similarly to [26], we find the TopK activation to dominate on the sparsity-accuracy frontier—we thus use the TopK activation for all experiments.

B.8.2 Choice of MxD encoder

Secondly, we show in Figure 18 the benefits of MxDs’ flexibility in inheriting the original MLP layer’s encoder form/activation function. All models here are trained from scratch for the same number of tokens and with the same experimental setup as in Section 3.1, with $K = 32$. In the

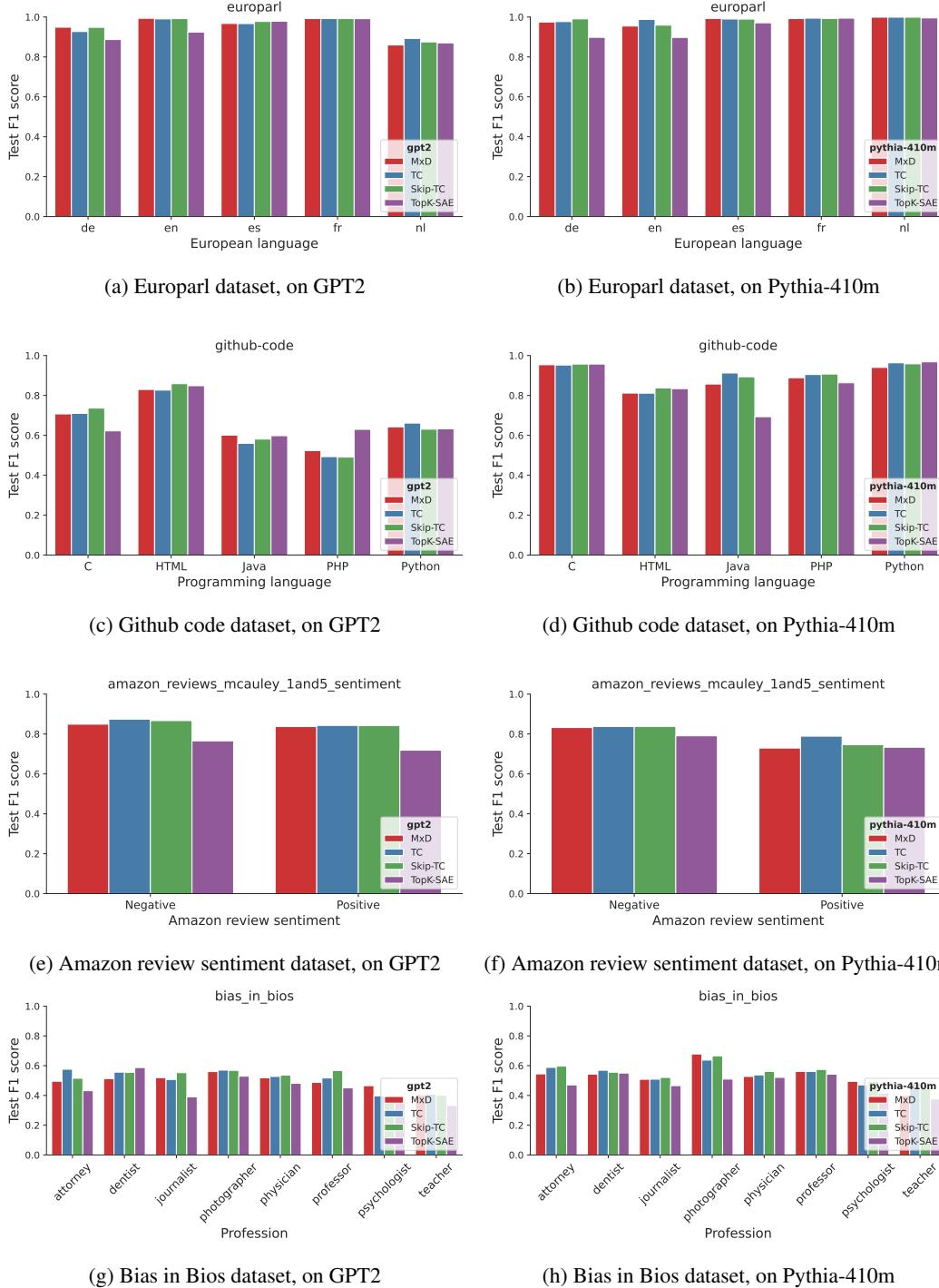


Figure 12: **Sample-level** sparse probing results on individual experts/features; the best F1 score on a held out set is presented.

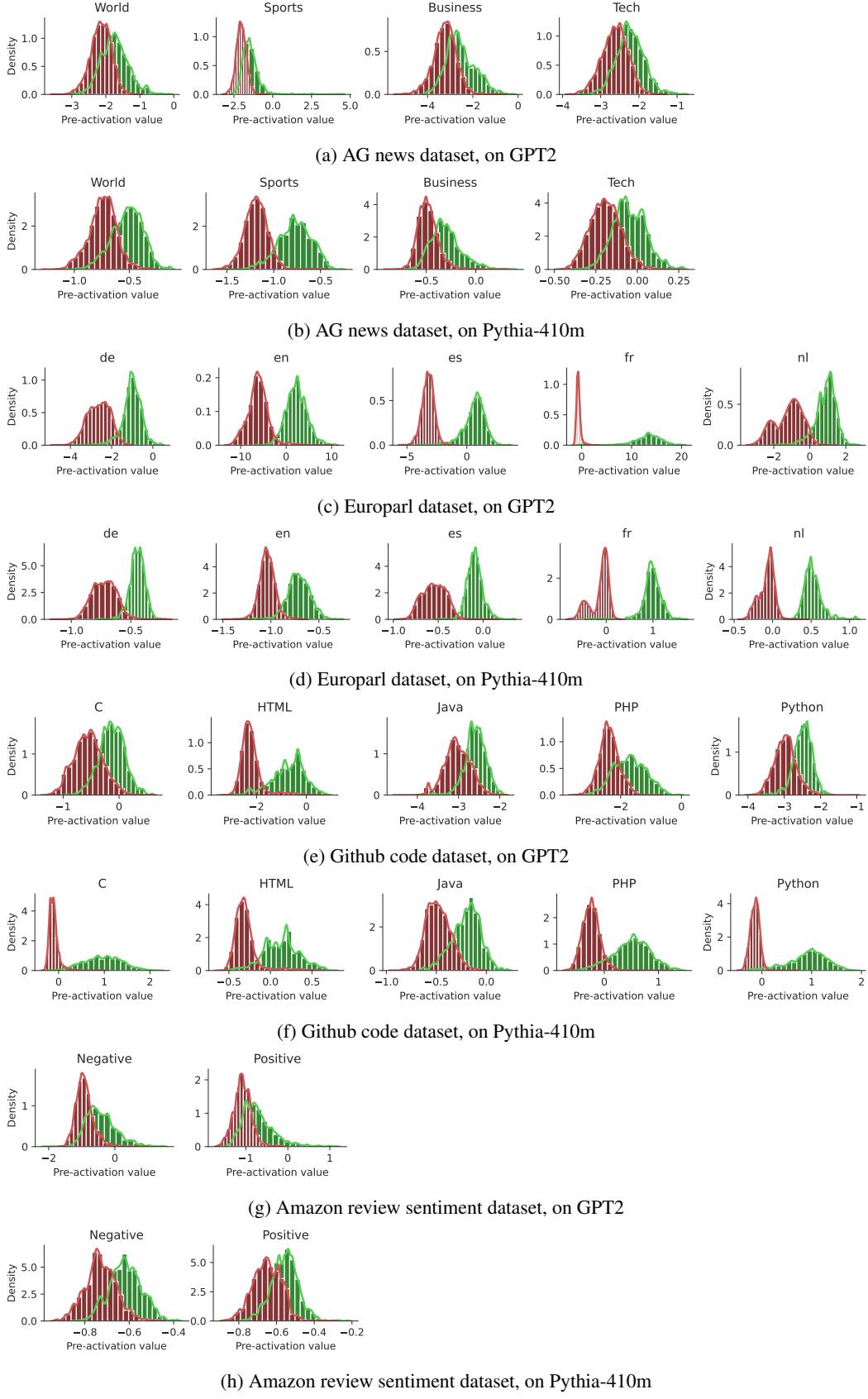
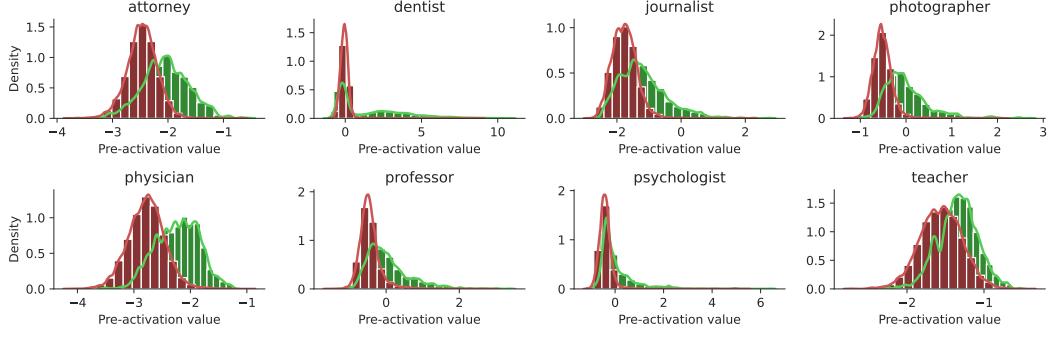
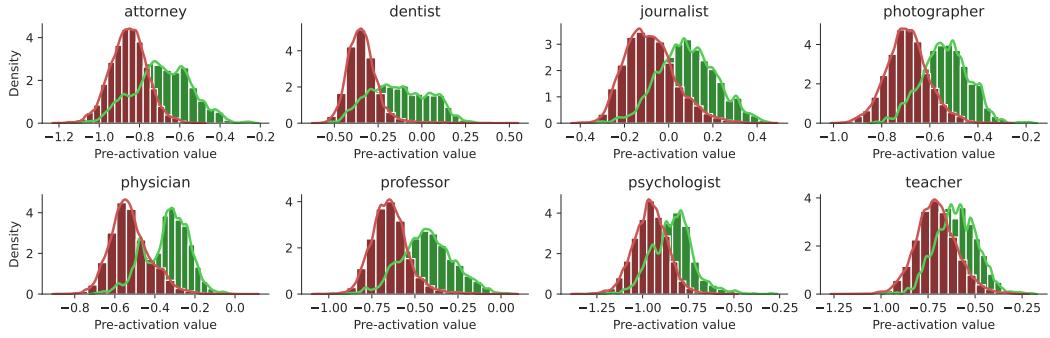


Figure 13: **[1/2] Sample-level** sparse probing results on individual experts for MxDs; here we plot the values of the expert pre-activation for **positive/other** classes (in the 1-vs-all setting).

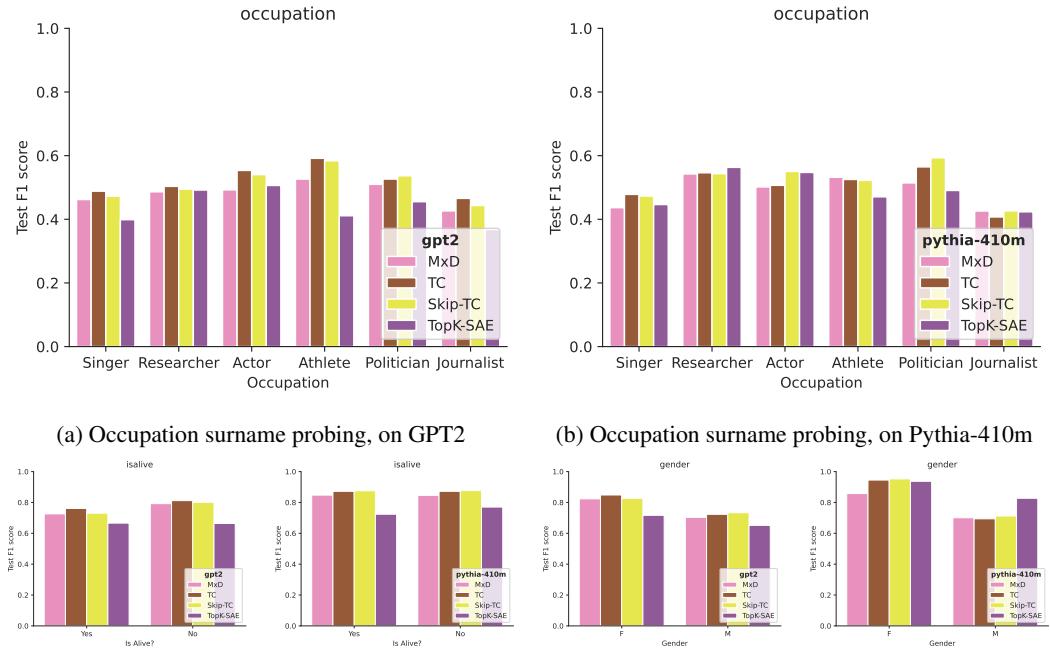


(a) Profession from biography, on GPT2



(b) Profession from biography, on Pythia-410m

Figure 14: [2/2] **Sample-level** sparse probing results on individual experts for MxDs; here we plot the values of the expert pre-activation for **positive/other** classes (in the 1-vs-all setting).



(c) “Is alive?” surname (d) “Is alive?” surname (e) Gender surname prob- (f) Gender surname prob-
prob, on GPT2 prob, on Pythia-410m ing, on GPT2 ing, on Pythia-410m

Figure 15: **Token-level** sparse probing results on individual experts/features; the best F1 score on a held out set is presented.

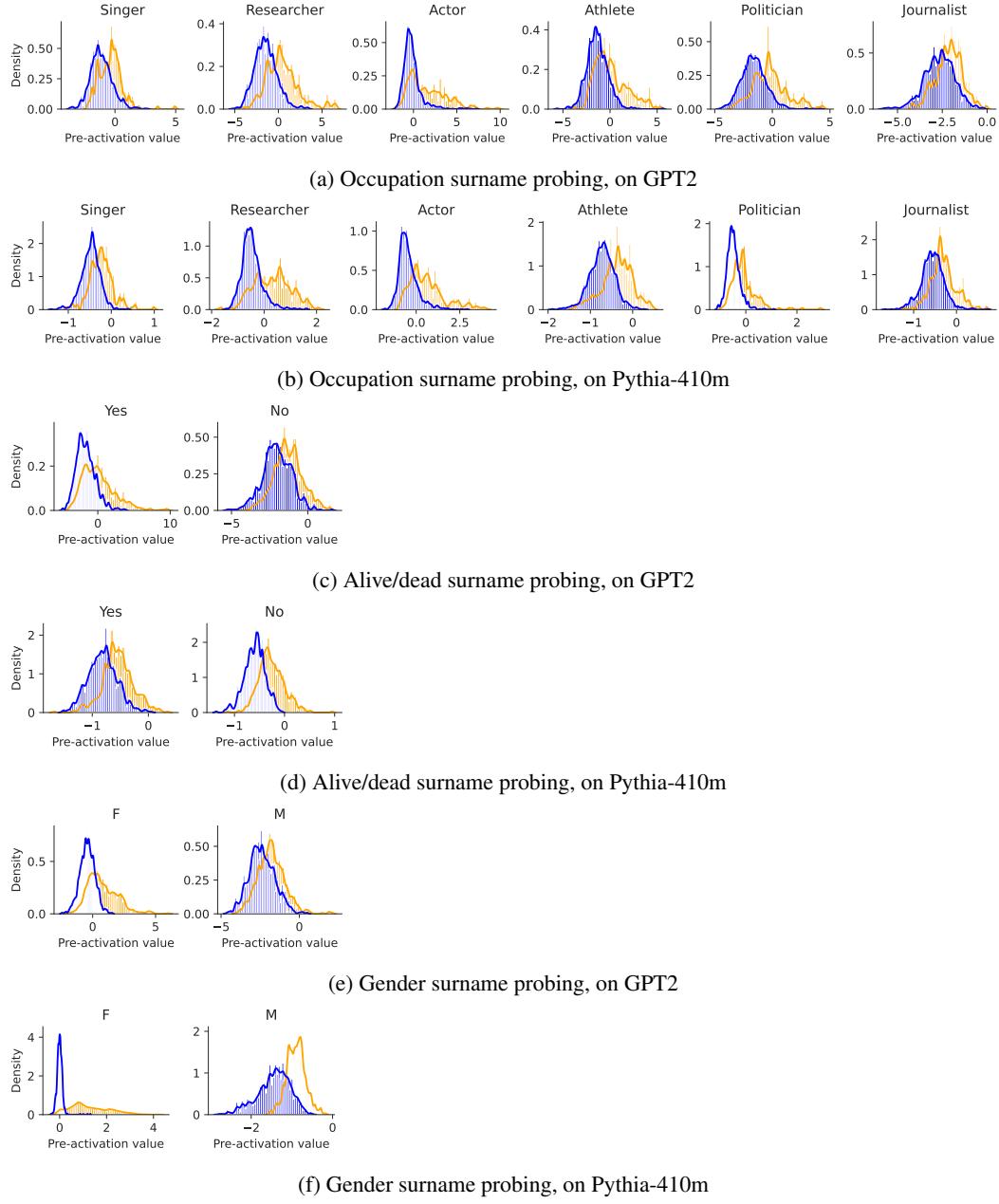


Figure 16: **Token-level** sparse probing results on individual experts for MxDS; here we plot the values of the expert pre-activation for **positive/other** classes (in the 1-vs-all setting).

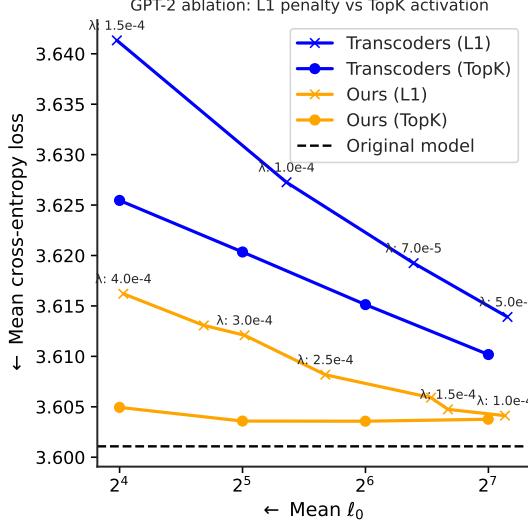


Figure 17: ReLU+TopK activation function [23] vs ReLU w/ L1 sparsity penalty [22]: both MxDS and Transcoders better recover the cross entropy loss with the TopK activation.

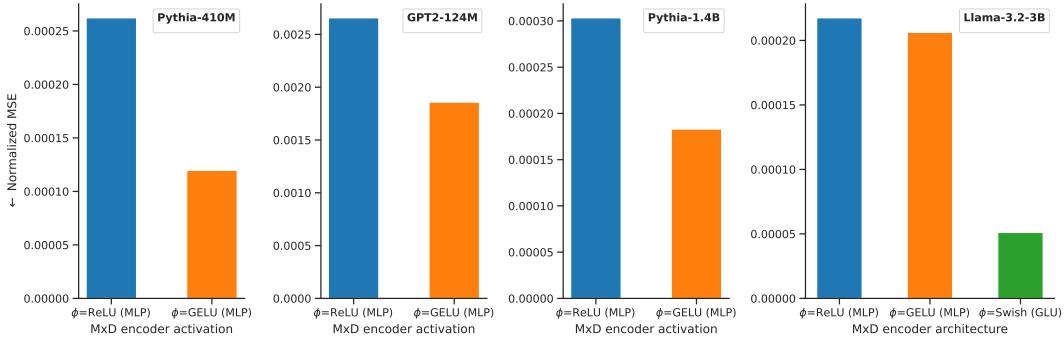


Figure 18: **Encoder architecture ablation:** MSE loss when using ReLU activation vs the GELU used by the base models; and MLPs vs GLUs for Llama (rightmost subfigure).

first 3 left-most subfigures, we see the Normalized MSE is as low as half when using GELU vs the non-native ReLU activation.

We next ablate the impact of inheriting the same encoder as the Llama-3.2-3B base model. In the rightmost subfigure of Figure 18, we train MxDS with ReLU-MLP, GELU-MLP, and Swish-GLU encoders. As can be seen, using a GLU with a Swish activation model (matching the base model architecture) yields a Normalized MSE almost an **order of magnitude** smaller than MLPs with GELU/ReLU.

C Feature balance and shared experts

C.1 Expert/feature balance

Following the code of [27, 109], we log how often each unit of specialism/feature is used, over a fixed window of $\sim 1M$ tokens. We show in Figure 19 the feature frequency at the end of training, where we observe that MxDS see a similar healthy balance of experts to the frequency of usage of features in the baselines.

Interestingly, we observe a small peak of experts that fire more frequently in MxDS (e.g., around -2 on the x-axis)—perhaps specializing in common patterns and primitives in natural language.

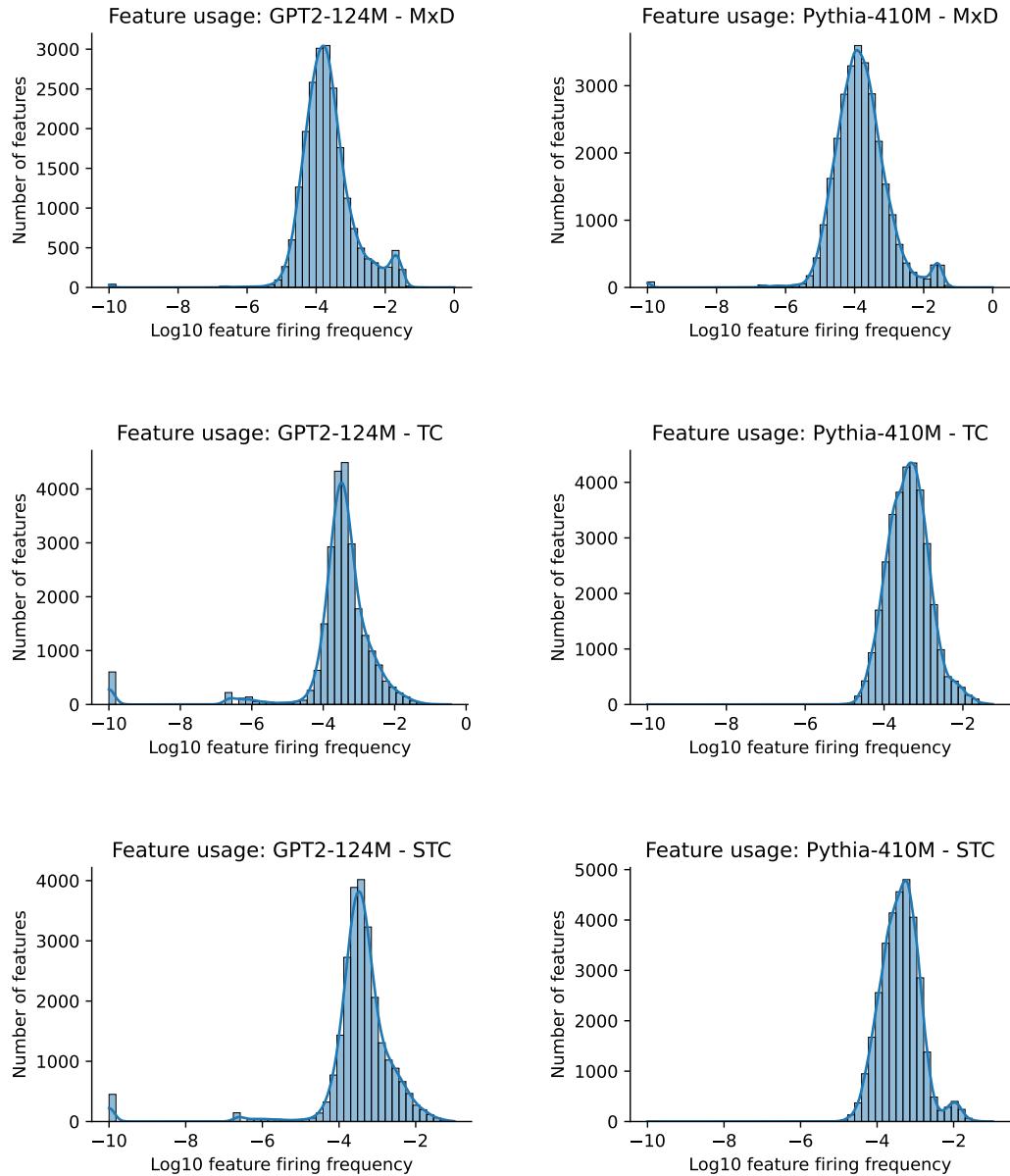


Figure 19: \log_{10} feature sparsity (following [27, 109]); MxDs' experts are well-balanced, similar to the baselines' features.

C.2 Shared experts

We find that, by default, our MxD models naturally learn to use a shared expert, with the remaining experts exhibiting strong specialization in a wide range of themes and linguistic patterns. The use of a shared expert is becoming an increasingly popular design choice, including in the latest Llama 4 models [44]—we therefore allow this pattern to emerge naturally in our base models, further justified through the evidence in [43] that shared experts can enhance specialization among the remaining experts [43]. We highlight, however, that a simple trick of sampling $\hat{K} \sim \text{Unif}\{K - K/a, K + K/a\}$ for the Top- \hat{K} activation at train-time (for e.g. $a := 2$) is sufficient to remove the dominating shared-expert at minimal hit to reconstruction performance, if desired.

We train two sets of models with a base $K = 32$ on GPT2-small and pythia-410m, using $a := 2$. We first show in Figure 20 the indices of the top-activating experts for the 2 model variants on a template prompt, after training has finished. On the left-hand side of Figure 20, the models route all tokens through the same shared expert at position 1. However, we see on the right-hand side that training with the ‘random-K’ strategy breaks the dependence on a shared expert in position 1. Furthermore, we include in Figure 21 the corresponding train-time MSE loss for the 4 models here as ablations—observing that the random-K strategy also brings comparable performance. Based on these experiments, we recommend this simple training strategy if one desires MxD models without shared experts.

Expert diversity/collapse *To what extent is the shared expert simply learning the base decoder itself?* In an initial attempt to study this, we take the same MxD trained with $K = 64$ on GPT2, and tabulate the Frobenius norm of differences between the original model’s decoder \mathbf{D}^* , our learned base decoder \mathbf{D} , and the “shared expert” weight matrix $\mathbf{W}_{\text{shared}} = \mathbf{D} \text{diag}(\mathbf{c}_{\text{shared}})$ (with index $\text{shared} = 19772$), all of which are matrices of shape $(H \times O)$. As can be seen in Table 9, their difference is significant, showing that the shared expert does not simply learn the original decoder.

As discussed in the limitations section, we highlight that MxDs as formulated contain no explicit loss terms to discourage expert collapse, or encourage expert balance or diversity. Whilst we find no need for these in our experiments, they may prove useful (or even necessary) in other settings or future work.

Table 9: Distance between learned shared expert, base decoder, and original decoder

| $\ \mathbf{W}_{\text{shared}} - \mathbf{D}^*\ _F$ | $\ \mathbf{D} - \mathbf{D}^*\ _F$ | $(\mathbf{W}_{\text{shared}} - \mathbf{D}^*)_{:,3}$ (9 elements of the difference matrix) |
|---|-----------------------------------|--|
| 208.08 | 221.6934 | $\begin{bmatrix} 0.09 & 0.03 & 0.06 \\ 0.21 & 0.12 & 0.07 \\ 0.17 & 0.15 & 0.11 \end{bmatrix}$ |

D Detailed experimental setup

We list in Table 10 the resources used for each experiment: the GPU and the indicative run-time for a single model. The `mlp_expansion_factor` column refers to the expansion factor applied to the input dimension to generate the MLP width in the sparse layers (i.e. $H := I \cdot \text{mlp_expansion_factor}$).

Table 10: Total training time and resources used to produce the $k = 32$ experiments (the required compute being roughly the same across models trained with different k).

| Model | GPU used | VRAM | Training time | d_{in} | mlp_expansion_factor | Asset link |
|--------------|---------------------|------|---------------|-----------------|----------------------|---|
| GPT2-124m | x1 GeForce RTX 3090 | 24GB | 8h 34m 37s | 768 | 32 | https://huggingface.co/docs/transformers/en/model_doc/gpt2 |
| Pythia-410m | x1 GeForce RTX 3090 | 24GB | 8h 35m 17s | 1024 | 32 | https://huggingface.co/EleutherAI/pythia-410m |
| Pythia-1.4B | x1 A100 | 80GB | 23h 25m 23s | 2048 | 32 | https://huggingface.co/EleutherAI/pythia-1.4b |
| Llama-3.2-3B | x1 A100 | 80GB | 2d 3m 50s | 3072 | 32 | https://huggingface.co/meta-llama/Llama-3.2-3B |

Model trained with fixed K

| | 1st highest expert index | 2nd highest expert index | 3rd highest expert index | 4th highest expert index |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| Token 1 | 10160 | 10962 | 19772 | 9610 |
| Token 2 | 19772 | 15461 | 2630 | 8228 |
| Token 3 | 19772 | 18694 | 7385 | 3494 |
| Token 4 | 19772 | 19466 | 10619 | 970 |

Model trained with random K

| | 1st highest expert index | 2nd highest expert index | 3rd highest expert index | 4th highest expert index |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| Token 1 | 526 | 18499 | 7257 | 8244 |
| Token 2 | 16092 | 3344 | 17100 | 7388 |
| Token 3 | 19829 | 10864 | 7720 | 5507 |
| Token 4 | 20001 | 15277 | 1905 | 11387 |

Prompt: "Who is the president of the USA?"

GPT2-small

Model trained with fixed K

| | 1st highest expert index | 2nd highest expert index | 3rd highest expert index | 4th highest expert index |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| Token 1 | 28104 | 1694 | 18149 | 2013 |
| Token 2 | 28104 | 1163 | 5124 | 11890 |
| Token 3 | 28104 | 5124 | 27687 | 3657 |
| Token 4 | 28104 | 4126 | 12814 | 23628 |

Model trained with random K

| | 1st highest expert index | 2nd highest expert index | 3rd highest expert index | 4th highest expert index |
|---------|--------------------------|--------------------------|--------------------------|--------------------------|
| Token 1 | 7412 | 13294 | 3097 | 19430 |
| Token 2 | 13439 | 24209 | 13723 | 18099 |
| Token 3 | 9587 | 3857 | 10715 | 6198 |
| Token 4 | 2809 | 3378 | 25799 | 9435 |

Prompt: "Who is the president of the USA?"

Pythia-410m

Figure 20: Top-activating experts for template prompt with and without using a randomized value of K at train-time for TopK expert selection: randomization largely prevents a shared expert. Shown are the leading 4 tokens and expert indices.

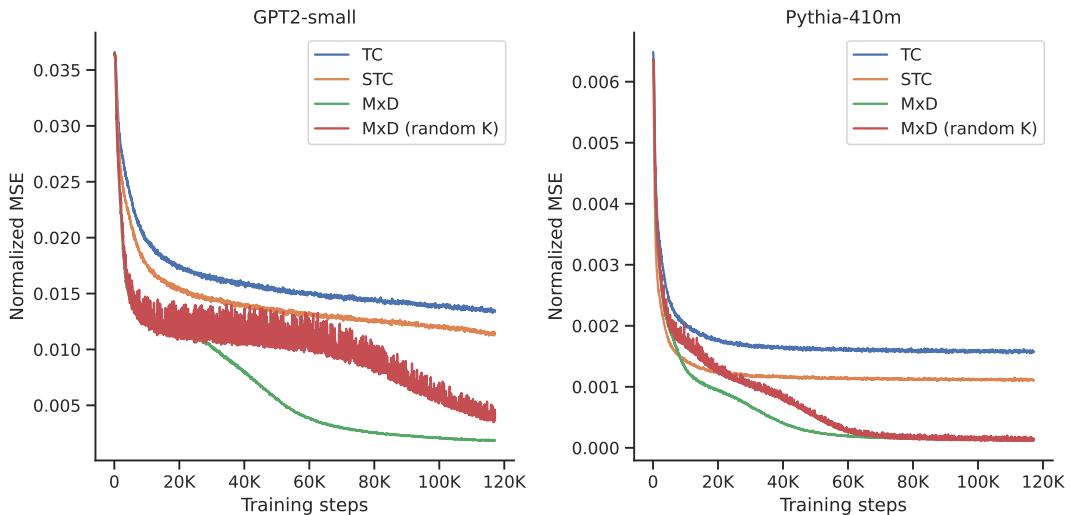


Figure 21: **MxD performance with random K sampling:** Normalized MSE loss as a function of training steps using a fixed Top $K := 32$ expert selection and when sampling $\hat{K} \sim \text{Unif}\left\{K - \frac{K}{2}, K + \frac{K}{2}\right\}$.

D.1 Feature steering details

For the steering experiments, we use two LLM judges to grade generations on two axes. The full template prompt we feed to `gemini-2.0-flash` and `llama-4-scout-17b-16e-instruct` is as follows (note that line breaks and emphases are included here only to aid visualization):

Prompt given to LLM judges

You are an expert evaluator of synthetic text.

TASK: Rate a collection of `{num_samples}` samples along two independent axes.

AXIS 1 – CONCEPT COHERENCE:

0.00 no shared concepts/themes/style.

0.25 faint overlap.

0.50 some overlap or similar structure.

0.75 mostly the same concepts or structure; a few partial drifts.

1.00 all snippets clearly share the same concepts, themes, style, or structure.

AXIS 2 – GRAMMATICAL FLUENCY:

0.00 incomprehensible.

0.25 dense errors; meaning often obscured.

0.50 frequent errors; meaning still mostly recoverable.

0.75 minor errors that rarely hinder comprehension.

1.00 completely grammatical and natural.

(Do not penalise fluency if a snippet starts or ends abruptly.).

SCORING: Choose any real value in $[0, 1]$ for each axis.

OUTPUT FORMAT: Respond with exactly two numbers formatted ‘0.00, 0.00’ in the order [coherence, fluency] and no other text or symbols.

TEXT TO EVALUATE: {samples}

E Additional qualitative results

We show in Figures 22 and 23 tokens activating the first 9 experts as they appear numerically. We sample 6 bins of expert coefficient value to show both tokens that highly activate the experts and those that do so only mildly. As can be seen, both high- and low-level specializations emerge in both GPT and Pythia models.

Whilst we observe specializations to a range of concepts (such as punctuation, MMO games, words in specific contexts), we do not notice any systemic differences between the types of expert specializations that emerge between the two models in MxD layers.

Pythia-410m: Tokens routed to specific experts

Feature 0

Activation between 3.86 and 4.63
Vol. 1, **No. 1** (2002)

Activation between 3.09 and 3.86
Vol. 1, **No. 1** (2002)
Vol. 1, **No. 2**, **May**
of firearms listed below paragraph **No. 2**. It is
Vol. 1, **No. 1** (2002)
Vol. 1, **No. 2**, **May**
NBER Working Paper **No. 21611**

Activation between 2.32 and 3.09
on Rural Free Delivery Route **No. 1**, where
Journal of **No. 1**, **May** (2002)
- Violin Sonata **No. 9** in A
- Violin sonatas **no. 1**

Activation between 2.32 and 2.32
- Shawn Patterson at **No. 18** overall,
Piano Sonata **No. 1** Op.
agio String Quartet **No. 1** - 3
- **No. 1** - 3
Artist Title String Quartet **No. 1** Brah

Activation between 0.77 and 1.54
Rams in exchange for the **No. 1** pick and
Journal of **No. 1**, **May** (2002)
(**No. 1** pick and **No. 1** pick
th). Canada is ranked **no. 10** in the International Migration Review, Vol. **27**, **No. 4** (2003). The **No. 1** pick is the **No. 1** pick in 2006 Init

Activation between 0.00 and 0.77
Vol. 1, **No. 1** (2002)
Rams in exchange for the **No. 1** overall pick
of Shepard as the **No. 2** option New
<endoftext>rd period, 1

Feature 3

Activation between 3.28 and 3.94
Kathmandu, many rescue crews had yet to arrive.
Activation between 2.63 and 3.28
Kathmandu, many rescue crews had yet to arrive. Some buildings were severely damaged, hampering rescue and humanitarian efforts.
While people see that the rescue had failed, A *judgment* minimum protection or *recovery* work, pulling a *person* from the rubble, he was *ugly*.
Activation between 1.97 and 2.63
dead, was reported, *Recovery* was generally slow, the region to *begin* *rehabilitation* operations. Box trucks, *recovery* equipment and even an *offer* to contribute a "rescue offer" like the 2013 and triggered a massive *relief* effort. Like Maria.
Activation between 1.31 and 1.97
Activation immediately began *resuscitation* procedures, including *medications* was followed by a *federal* *relief* effort, widely critiqued for its *inaction* and *negligence*. *Rescue* *team* *size* of damage and amount of *recovery* time required after the *on* *relief* *operation*. The *rehabilitation* process will also take.
Activation between 0.66 and 1.31
Smoke and began the long *cleanup* process – an RV Smoke forces passengers to *evaluate* plane in Houston cost-of-service *recovery*, just as they *on* *copter* opens fire. *Emergency services* attend to injured *afford*.
<jndotext> come to the rescue on screen. I
Activation between 0.00 and 0.66
the *recovery* *team* *size* for U-Bu while DRC is *recovering* from years of internal damage and amount of *repairing* the damage could run challenges is trying to *bounce* back against the Colorado Rapid.
Rapid.
<jndotext>/rd period. 1

Feature 6

Activation between 2.57 and 3.09
Keepers is a remarkable and enormously enjoyable documentary

Activation between 2.06 and 2.57
Keepers is a remarkable and enormously enjoyable documentary

tackled in a bizarre and post-apocalyptic dizzying, confusing, and scattering dance-like including some very challenging and difficult times in recent

<endoftext> to the strange and perilous place.

Activation between 1.54 and 2.06
his work such a varied and wonderful feel that titanic and ultimately violent music

"It is complicated and probably fraught to reasons for this are complicated and painful but here is the Up Volume: the volume and a little

Activation between 1.03 and 1.54
that were the most surprising or impactful. Hulk is a big and demanding game especially on sophisticated, disarming and even counterintuitive Multiplayer mode, and it's a delight watching Scotland turns

Energetic and clean. One of

Activation between 0.51 and 1.03
said something so misguided and distasteful that systems are so obscure and complex that it takes a quick look at the game's official

Church, including its complex and sumptuous music increasingly riven by angry, uncivil rivals with

Activation between 0.00 and 0.51
said something so misguided and powerful computers. an eye catching. If dysfunctional, we were stunningly disfunctional and weakened by in-Halloween is DUMB and predictable and DUM

<endoftext>rd period, 1

Feature 1

Activation between 3.81 and 4.57
you will. [Version 3.26 of](#)
Activation between 3.05 and 3.81
you will. [Version 3.26 of](#)
background image in the media runs [version 4.4.6](#),
to author. [Version 1.15,](#)
which was introduced in [version 3.7.](#)
[Version 1.15 \(Version 4.4 required\)"](#)

Activation between 2.28 and 3.05
What's New in [Version 2.4](#)
release of the [Wordpress 0.1.](#)
goal is [Version 2.4](#)

It was initially released as [version 7.0](#) on
be issuing some of the [version](#), while Edward
Activation between 1.9 and 2.28
this was the [version](#) that independent trials on
weakness, affecting the latest version of [Wordpress, 4](#)
h. It began as [version](#) of the Arabic dish
language, and was later adopted in [Version 2.10](#).
[Version 2.10, that version will not necessarily have](#)

Activation between 0.76 and 1.52
have to UNINSTALL this [version](#) and [Version 1.52](#)
forked off to create a [version](#) of [Bitcoin](#); the
itself with [another version](#) of the mod installed
developers to bring much newer versions of [apps to](#)
development, and the [newer versions](#) of the same song

Activation between 0.0 and 0.76
35 using a slightly modified [version](#) of [John Resig](#)
A previous [version](#) of this article suggested
All the [version](#) had a [somewhat](#) [disappointed](#) in the [2015 version](#) as there is not
[<endif> text](#), [version 1](#)

Feature 4

Feature 2

Activation between 3.0.1 and 3.6.1
of release of *Starcraft 2*, the popularity

Activation between 2.4.1 and 3.0.1
of release of *Starcraft 2*, the popularity
suggested release *Alpha* **Data 2** officially launched out
the new era of *MMO* players, which inevitably
industry; this gives **Data 2** a more personal feel
going to prefer. **Data 2** is the first

Activation between 1.81 and 2.4.1
difficulty of *Vanilla WoW* because at the end
from additional skill), **Data 2** has become hugely
impossible."A new *Overwatch* patch is currently in
circuit), the *starcraft* card just exploded in
who has been playing since the beginning

Activation between 1.20 and 1.81
difficulty of *Vanilla WoW* because at the end
top teams in *Rocket League* - Ted I
active mindset that other *MMOs* encourage, and
worry about the game, but VoidSpace
As an *SC2* fan, I am

Activation between 0.60 and 1.20
Shop items on the *Multiplayer* website. However,
attack. In this case, the *MMOs* like
, like *Dungeons & Dragons* or *Pathfinder*,
you often hear about massively **multiplayer** online games
like *World of Warcraft*.

in this way, *Mass Effect* is (mostly)

Activation between 0.0 and 0.60
Wish is a simple exploration *RPG*, but creates a
the coaching staff of *SKT T1* is really
told that the *MMOs* on the lava
of the station in the *game* themselves, be it
<endoftext> a period, 1

Feature 5

Activation between 3.51 and 4.21
order = **Met** Kiper's final
Activation between 2.81 and 3.51
order = **Met** Kiper's final
<endText> **Met** 19 Mel
Like a Saturday night at **Met** Gibson's house
Christine Sinclair **Met** Sanae Tancrez
Bashir **Met** **Met** P.
Activation between 2.11 and 2.81
ir, the producers wanted **Melora** to have a
Mawson 45 **Melora** 9 Baint
Bashir **Melora** 19 Mel
changes his mind and shoots **Melora**. A charge
in mid-air, **Melora** tells Bashir
Activation between 1.40 and 2.11
Hitting coach **Melora** 19 Mel
Bashir **Melora** 19 Mel told us
charming besides banner" **Melora** laughs but,
angry stuff and forget the **Melora** disco-
us" **Melora** 19 Mel
each other's eyes, **Melora** suddenly breaks the
Activation between 0.70 and 1.40
Kingsford Smith Drive, **Melba** 5 22 Mel
10 Lorraine 19 Mel
, **Melba** 19 19 **Melba** 15 Crosley
the permafrost melts, a potent
Activation between 0.40 and 0.70
body, son, **Melba** 19 Mel they are
21-year-old **Nicah** David-Cole
B-Type 50MM Microdrive Att
> **Melba** 19 Mel **Melba** 19 Mel
<endText> 1 points **Melba** 19 Mel
<endText> **rd** period, 1

Feature 7

Activation between 4.02 and 4.83
p.m., Pac-12 Network). Some
Activation between 3.32 and 4.02
p.m., Pac-12 Network).
12 schools in the Pac-12 without an
Three schools in the Pac-12 – California,
Activation between 2.41 and 3.22
Activation between 1.61 and 2.41
Margill, Coca-Cola, General
six companies: Coca-Cola, General
WWF and The Coca-Cola Company have
6:00 p.m., PAC-12 WORKERS
Activation between 0.80 and 1.61
East, an AFL-CIO affiliated union
the Greater New Orleans AFL-CIO, said
the St. Louis Globe-Democrat said that
union, the Colorado AFL-CIO
Activation between 0.00 and 0.80
a source on the Globe-Democrat with information
Bernardine, the AFL-CIO president, proposed the
to the University of California at Berkeley to demand
s plan, CNN-78 will be delivered
<endoftext> period, 1

Feature 8

Figure 22: Tokens activating the first 9 numerical experts on MxDs with $K = 32$ trained on Pythia-410m; we sample 6 bands of activations to show both tokens that highly activate experts and those that activate them only mildly. Magnitude of activation is denoted by the **orange** highlight. Moderate specialism emerges, e.g., to MMO games, abbreviations, and words in specific contexts.

GPT2-124m: Tokens routed to specific experts

Feature 0

Feature 3

Activation between 1.78 and 2.13 majority of the **Fourth Circuit** To them, armed
Activation between 1.42 and 1.78 majority of the **Fourth Circuit** To them, armed
and so on, the **Fourth Circuit** have analyzed
supportive of medical research spending. Related;
also greatly expanded in 2012. Gartner projects
Activation between 1.07 and 1.42 recovering in their first year, Bill Clinton⁶⁶ and George W. Bush⁶⁷ The one thing that
the future of the country⁶⁸ Despite poor view of
the Bureau in the past, its publisher [name]
Tougher than other public pols. His approval rating dipped
Activation between 0.71 and 1.07 join the nascent Zionist movement⁶⁹ They remained a
curiosity.
every year, and it is suitable. Towns and cities have
by re-election. The last time it
66 being rewritten. It⁶⁸
illegal conduct by the Government. If this conduct can
Activation between 0.36 and 0.71 the two press releases. The panels were cement
to the East. First, Gruber⁷⁰ In some cases,
between Bayview and Leslie. But don't⁶⁶
release information about this collection, Deputy
and over and over again, With the cuts to
Activation between 0.00 and 0.36 their new and chosen country. But there
, but still love them.⁶⁶
to Martin Luther King Jr. The FBI saw Dr
have access the traditional program. Instead, they would
<endoftext> him free hand,

Feature 6

- Activation between 4.52 and 5.42
states and territories – second **only** to the century.
- Activation between 3.62 and 4.52
states and territories – second **only** to the century-accuracy. They were second **only** to Arsenal when it 2004 became second **only** to the century.
- Activation between 2.71 and 3.62
Obama and came in second **only** to Oprah Winfrey
- Activation between 1.81 and 2.71
- Activation between 0.90 and 1.81
a dozen wins, second **only** to Washington's 17-Activation between 0.00 and 0.90
of the slot (behind only Golden Tate). And Muslim travel market, behind **only** Malaysia and the United DB ranks second **only** to CYP3A
<|endoftext|> him from behind,

Feature 1

Activation between 5.79 and 6.95
the works performed by [E.g.](#) Kremer

Activation between 6.63 and 5.79
the works performed by [E.g.](#) Kremer

```
<endofText> <endofText> <print>
```

and its variants, [E.g.](#) [6.6](#)
variety of sources, [E.g.](#) newspapers,
newspaper clippings, [E.g.](#) [6.6](#)

Activation between 3.47 and 4.63
<endofText> hardware constructs ([E.g.](#), defining the
reduced cross sections ([E.g.](#), I-
omega, [E.g.](#) [3.47](#), the
UV layer, [E.g.](#), that it
of PFC ([E.g.](#), Diamond and
Krisztian, [E.g.](#) [3.47](#))

Activation between 2.32 and 3.47
fish oil consumption, [E.g.](#) Krisztian
to behave defensively, [E.g.](#) sting or
Germany's parliament ([i.e.](#), the Bund
entirely to the right, [i.e.](#), the right
Chinese Tribune publication, [E.g.](#) Jason Wambs

Activation between 1.16 and 2.32
service was held in [Ex](#) Peck Greene Park,
Spirit, [E.g.](#) Joseph, [E.g.](#) [1.16](#)
of the DDR ([i.e.](#), two pairs of
Communist government headed by [E.M.](#), Nam
explanations of religion, [E.E.](#) Evans

Activation between 0.90 and 1.16
and former [E.M.](#), [W.O.](#)'s
SSB mode), [S.](#) McGreevy,
THE AUTHOR, A.J., Kingsniss
<endofText> him, he has never been
<endofText> him, from behind,

Feature 4

Activation between 4.21 and 5.05
council hasn't ruled out additional forms of
Activation between 3.37 and 4.21
council hasn't ruled out additional forms of
Bipartisan group that plans most
waiting for the DOJ to rule on a decision by
the Supreme Court to rule on what options are
the Librarian of Congress ruled that phones purchased
Activation between 2.52 and 3.37
the Forest Service has not ruled out closing the area
consultancy due next month before deciding whether to
throw it out
Deeply shocked & saddened by #Un
The Viking King Guthrun ruled the Danewel
seems completely unconscious of what you are doing that
Activation between 1.68 and 2.52
#. In a game obsessed with negative responses to
No options should be ruled out.
and an Radio Network that decided to escape after
reading it.
Included in the list of following matters is **60**
at USDA's Systematic Mycology and Micro
Activation between 0.00 and 1.68
"we are all perceived Al"
. Aside from commenting on how I feel
when they're feeling really bitter, they'll point
Humanity out there all along.
2) criticize edit -
Activation between 0.00 and 0.84
= sink ^ (a : var string
a warrior young man listening to your
at Will Home Wife is today is the
Then, in 2004, the Sharon government

Feature 7

Activation between 6.05 and 7.26
the New York Post's [Zach Braziller](#).

Activation between 4.84 and 6.05
the New York Post's [Zach Braziller](#),
kin, Brandon Graham, [Zach Ertz](#), Col-
from the [Zach Ertz](#) who got involved
early with an injury. [Zach Ertz](#) played a
ll on the departure of [Zach Parise](#).

Activation between 3.63 and 4.84
you can't just [Zach Parise](#) for the
value because, well, [Zach Ertz](#) is awesome
the case of crime boss [Zachary Kalashov](#)
not much out there until [Zachary Drive](#). They

Activation between 2.20 and 3.63
Phil Kessel, [Zach Parise](#),
Brisebois, [Zack MacEwen](#) or
while at the plate, [Troy Frane \(.313/](#)
[Victoria Arlen](#)) and [Political Director](#)
[You can't replace a \[Zach Parise\]\(#\), his](#)

Activation between 1.21 and 2.42
office reportedly told [AAP - ZDNet.com](#)
about his husband? [Zach Parise](#)
still hanging around, [Zach Parise](#) pays employees
[Tulip Experience](#) [Zibi Gotinnes](#)
[season, #66](#) said Zona, the study

Activation between 0.60 and 1.21
for Tang and [Zach Parise](#), [Morales](#) at
[VALUATIONS](#), RED ZONE RESIDENTS
his eventual contribution to the zerg lexicon
reason to complain about the [Zenkar](#) lands
[Aftermath](#) and [Zach Parise](#).

Feature 2

Activation between 5.35 and 6.42
When Kermit claims to hear voices, perhaps

Activation between 4.28 and 5.35
When Kermit claims to hear voices, perhaps
conservative protestants to claim to speak with the dead
Petersburg and Moscow often claimed to represent the
interests of

Activation between 6.66 and 7.77 by claiming to have been under sniper
Nuru who claimed to be a minor in

Activation between 3.21 and 4.28
, while Donald Trump claims to love his decidedly first
egar, when claim to have been shot
was that I would probably shoot the web,
claims Palmer was wrongly claiming to represent him to
developers

Car Paleasing pretend to fight, while cutting

Activation between 2.14 and 3.21
who claimed, falsely, to have copied it from
what the totalitarian approach claims to have revealed,
but

early 6.60% to be the country's
Orleans, Louisiana, some claim to
this person. "The claim to have outed the

Activation between 0.07 and 2.14
and the most notorious to hear, but in
it couldn't even pretend to have in 2016,
a video Monday purporting to show pro-government
Luther, who claimed to have shared such pur
even to and claim to be the craziest

Activation between 0.00 and 1.07
few people would readily admit to being racist, it
on Craigslist – pretending to be her ex-f
<|endoftext> who appeared to essentially ignore the
please

Jesus if we claim to be part of His church
<|endoftext> him from behind,

Feature 5

Activation between 6.55 and 7.86 the outside oversight was undemocratic and impractical;

Activation between 5.24 and 6.55 the outside oversight was undemocratic and impractical;

Activation between 3.93 and 5.24 he presented a radical undemocratic and unconstitutional practice

Activation between 3.93 and 5.24 the outside oversight was undemocratic and impractical;

never
the demand for properties in undamaged areas of the figures went on relatively undisturbed. They relied

Activation between 2.62 and 3.93 which was the result of the actions by Detroit's
-State supercell undisturbed by nearby conve
who are ~~dismayed~~ dismayed by the initiative
Ballot initiative to be passed. As a

Activation between 1.31 and 2.62
<endoftext> | nondescript Saturday, with
law reform groups remain undeterred,
might be the result of the recent incidents
anger and anxiety still predominating, helping to
Marine R12. Underlining

Activation between 0.76 and 1.31
IRE 7 that this dream runner occasionally served
partner had ~~underestimated~~ the
<endoftext> bias in his collective from
1. Decided to
<endoftext> him from behind;

Feature 8

Figure 23: Tokens activating the first 9 numerical experts on MxDs with $K = 32$ trained on GPT2-124m; we sample 6 bands of activations to show both tokens that highly activate experts and those that activate them only mildly. Magnitude of activation is denoted by the **orange** highlight. Moderate specialism emerges, e.g., to punctuation, names, and months.