Enhancing Mixture-of-Experts Specialization via Cluster-Aware Upcycling

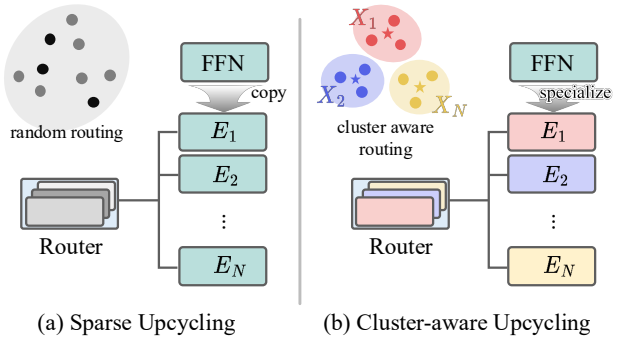
Sanghyeok Chu[*] [1,2] Pyunghwan Ahn[†] [2] Gwangmo Song[2] Seung Hwan Kim[2]
Honglak Lee[2,3] Bohyung Han[†] [1,4]
[1]ECE & [4]IPAI, Seoul National University [2]LG AI Research [3]University of Michigan
{sanghyeok.chu, bhhan}@snu.ac.kr {p.ahn, gwangmo.song, sh.kim, honglak}@lgresearch.ai

Abstract

Sparse Upcycling provides an efficient way to initialize a Mixture-of-Experts (MoE) model from pretrained dense weights instead of training from scratch. However, since all experts start from identical weights and the router is randomly initialized, the model suffers from expert symmetry and limited early specialization. We propose Cluster-aware Upcycling, a strategy that incorporates semantic structure into MoE initialization. Our method first partitions the dense model's input activations into semantic clusters. Each expert is then initialized using the subspace representations of its corresponding cluster via truncated SVD, while setting the router's initial weights to the cluster centroids. This cluster-aware initialization breaks expert symmetry and encourages early specialization aligned with the data distribution. Furthermore, we introduce an expert-ensemble self-distillation loss that stabilizes training by providing reliable routing guidance using an ensemble teacher. When evaluated on CLIP ViT-B/32 and ViT-B/16, Cluster-aware Upcycling consistently outperforms existing methods across both zero-shot and few-shot benchmarks. The proposed method also produces more diverse and disentangled expert representations, reduces inter-expert similarity, and leads to more confident routing behavior.

1. Introduction

Scaling model size has been a reliable strategy for improving performance [5, 8, 19, 23, 45]. However, this scaling incurs a significant computational cost. In standard dense architectures, where all parameters are activated for every input, training and inference costs grow linearly with model size. Mixture-of-Experts (MoE) architectures offer a sparse alternative by activating only a subset of parameters for each input token [10, 21]. This conditional computation enables models to scale efficiently without a proportional increase in computational cost, particularly during inference.



(a) Sparse Upcycling (b) Cluster-aware Upcycling

Figure 1. Comparison of Sparse Upcycling and Cluster-aware Upcycling. Unlike Sparse Upcycling, which inherently introduces expert symmetry, our method leverages semantic structure to initialize both experts and the router, promoting expert specialization.

However, training MoE models from scratch remains prohibitively expensive. Therefore, Sparse Upcycling [20] has emerged as an effective warm-start strategy that initializes an MoE model by reusing pretrained dense weights. This approach preserves the dense model's functionality at initialization, leading to faster convergence. However, initializing all experts with identical weights alongside a randomly initialized router inherently introduces expert symmetry; consequently, the model lacks a meaningful basis for early specialization. Prior work has attempted to break this symmetry through noise injection [4, 20, 28, 38], but these approaches often yield marginal performance gains. Drop-Upcycling [30] partially reinitializes expert parameters to encourage diversity, but this inevitably disrupts the pretrained representation space and undermines the core benefits of upcycling. Other methods fine-tune experts on manually defined domains to induce specialization [11, 36], but this requires explicit domain partitioning and additional fine-tuning stages, which fail to scale effectively to models with a large number of experts.

The key insight of our work is that the representations of a pretrained dense model already contain semantic information that can effectively guide the initialization of both experts and router parameters. Rather than treating the dense

[*]Work done during an internship at LG AI Research.
[†]Corresponding authors.

model as a monolithic entity to be replicated, we leverage its activation manifold to initialize experts into distinct subspaces while faithfully preserving pretrained knowledge.

Building on this insight, we propose Cluster-aware Upcycling, a strategy that incorporates semantic structure into MoE initialization. We utilize spherical k-means clustering to partition the activation space based on cosine similarity, which directly aligns with the routing mechanism. Each expert is initialized to capture the subspace of its corresponding cluster via truncated SVD. The router is initialized using the cluster centroids, ensuring that early routing decisions align with the underlying semantic structure of the data. This cluster-aware initialization breaks expert symmetry and provides the router with a meaningful prior for expert specialization from the onset of training.

In addition to the initialization strategy, we propose the expert-ensemble self-distillation (EESD) loss to further enhance training. Tokens with near-uniform routing probabilities often lack clear expert assignments, which can hinder the development of expert specialization. To address this, EESD distills predictions from a dense EMA ensemble teacher, motivated by [29], to provide stable supervision for the sparse MoE model, particularly for ambiguous tokens.

When evaluated on CLIP ViT-B/32 and ViT-B/16 models, Cluster-aware Upcycling achieves consistent improvements over existing upcycling methods on several zero-shot and few-shot benchmarks. More importantly, our analysis quantitatively confirms that the proposed method successfully resolves the problems of expert symmetry and redundancy. This structural improvement—characterized by significantly lower inter-expert similarity and more disentangled subspaces—translates directly into enhanced zero-shot and few-shot generalization, thereby demonstrating the clear benefits of structured, semantic-aware initialization.

In summary, our key contributions are organized as:

- We propose Cluster-aware Upcycling, a novel initialization strategy that considers latent semantic structures for both expert and router parameters, effectively breaking expert symmetry from the onset of training.
- We introduce the EESD loss, which provides stable ensemble-level supervision for tokens with high routing uncertainty, thereby preserving and enhancing expert specialization and robustness.
- We demonstrate that Cluster-aware Upcycling consistently outperforms upcycled CLIP models across various zero-shot and few-shot benchmarks, while promoting diverse and disentangled expert representations.

2. Background

2.1. Mixture-of-Experts

Scaling model size is a well-established path to improving performance. However, in dense architectures where all parameters are activated for every token, training and inference costs grow proportionally to model size.

Mixture-of-Experts (MoE) architectures offer a sparse alternative that decouples model capacity from the computing cost, thus improving efficiency. The key idea is to scale up the total number of parameters while activating only a small subset for each input token, such that different tokens follow different computational paths through the model. This sparsity allows the model to achieve the benefits of massive capacity without a proportional increase in computation, especially at inference time.

In standard dense Transformers, each block contains a feed-forward network (FFN), which is defined as

$$\mathrm{FFN}(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{W}), \tag{1}$$

where $\boldsymbol{W}$ is a learnable parameter matrix.

In MoE architectures, this FFN is replaced by a sparse MoE layer containing N_e expert networks $\{E_i\}_{i=1}^{N_e}$ and a router. Other modules, such as attention, are shared across tokens. Each expert shares the same architecture as the dense FFN but has its own set of parameters $\boldsymbol{W}_i$ as:

$$E_i(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{W}_i). \tag{2}$$

A router, parameterized by $\boldsymbol{W}_r$, produces routing probabilities that dispatch each token across experts:

$$g(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{W}_r \boldsymbol{x}), \tag{3}$$

where $g(\boldsymbol{x}) \in \mathbb{R}^{N_e}$, and $g_i(\boldsymbol{x})$ denotes the routing probability of assigning token $\boldsymbol{x}$ to expert E_i. The MoE layer output is given by:

$$y_{\text{MoE}}(\boldsymbol{x}) = \sum_{i \in \mathcal{T}_k(\boldsymbol{x})} \tilde{g}_i(\boldsymbol{x})\, E_i(\boldsymbol{x}), \tag{4}$$

where $\mathcal{T}_k(\boldsymbol{x})$ denotes the indices of the top-k entries of $g(\boldsymbol{x})$, and $\tilde{g}_i(\boldsymbol{x}) = g_i(\boldsymbol{x}) / \sum_{j \in \mathcal{T}_k(\boldsymbol{x})} g_j(\boldsymbol{x})$ is the renormalized probability over the selected experts.

MoE models are optimized using a task-specific loss $\mathcal{L}_{\text{task}}$, *e.g.*, cross-entropy or contrastive loss, together with an auxiliary load-balancing loss $\mathcal{L}_{\text{lb}}$ that encourages uniform expert utilization and prevents dead experts [10, 21]:

$$\mathcal{L}_{\text{lb}} = \sum_{i=1}^{N_e} a_i \mathbb{E}_{\boldsymbol{x}}[g_i(\boldsymbol{x})], \tag{5}$$

where a_i denotes the fraction of tokens routed to expert E_i.

2.2. Sparse Upcycling

Training MoE models from scratch is computationally expensive. Sparse Upcycling [20] offers a warm-start strategy by transforming a pretrained dense model into an MoE model. Concretely, each dense FFN is replaced with an

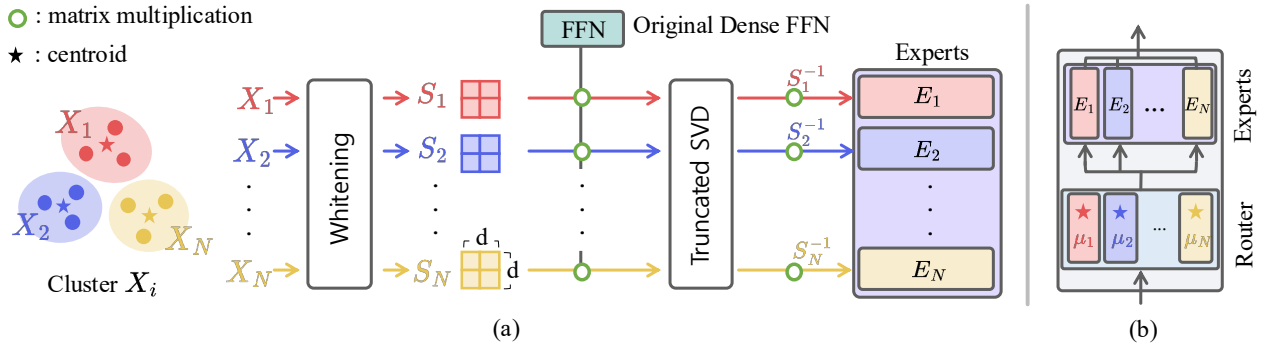


Figure 2. An illustration of how Cluster-aware Upcycling initializes the MoE layer. (a) Input activations are clustered to obtain whitening matrices and cluster centroids, which are used to initialize expert and router parameters, respectively. (b) The resulting MoE layer, where experts capture the subspace of their corresponding clusters, and the router aligns with the underlying semantic structure of the data.

MoE layer consisting of N_e experts and a router. All experts are initialized by copying the parameters from the original dense FFN, *i.e.*, $\forall i, \boldsymbol{W}_i \leftarrow \boldsymbol{W}$, while the router weight $\boldsymbol{W}_r$ is randomly initialized. All other components of the original dense model remain unchanged. This replication ensures that the upcycled model is functionally equivalent to the dense model at initialization, providing a warm start for subsequent MoE training.

While Sparse Upcycling efficiently leverages pretrained knowledge, it initializes all experts with identical parameters, which inherently hampers expert diversity. Since the experts are initially indistinguishable and the router is randomly initialized, establishing a meaningful basis for specialized routing in the early phase is challenging, failing to induce effective expert specialization.

3. Method

Our goal is to achieve expert diversity and structured routing while preserving the foundational knowledge of the pretrained dense model. To this end, we propose Cluster-aware Upcycling, a method that extracts semantic structures from the dense model's representations to initialize MoE parameters. The proposed strategy comprises three components: 1) partitioning the dense model's activations into clusters to capture the underlying semantic structure, 2) initializing expert parameters using the subspace representations of their corresponding clusters, and 3) initializing the router parameters with the cluster centroids. This process alleviates expert symmetry at the onset of training by aligning both experts and the router with the data's semantic manifold. Moreover, we introduce the EESD loss, which provides stable ensemble-level supervision particularly for ambiguous tokens. Figures 2 and 3 illustrate our cluster-aware initialization and the EESD loss, respectively.

3.1. Clustering Input Activations

To provide a meaningful basis for expert specialization, we first partition the dense model's activation space to extract a semantic prior for MoE initialization. We extract input token activations $\boldsymbol{X} = \{\boldsymbol{x}_j\}_{j=1}^{M}$, from each FFN block of the pretrained dense model, using a small calibration dataset that represents the training distribution. We then perform spherical k-means clustering using cosine similarity to group these activations. This clustering objective is deliberately chosen to align with the router's logit computation (i.e., $\boldsymbol{W}_r\boldsymbol{x}$ in Eq. (3)), which fundamentally measures the directional alignment. This ensures that activations with similar semantic directions, which should be assigned to the same expert, are clustered together.

Given ℓ_2-normalized activation vectors, the spherical k-means clustering objective is defined as

$$\{\boldsymbol{\mu}_i\}_{i=1}^{N_e} = \underset{\{\hat{\boldsymbol{\mu}}_i : \|\hat{\boldsymbol{\mu}}_i\|_2=1\}_{i=1}^{N_e}}{\arg\max} \sum_{j=1}^{M} \max_i \hat{\boldsymbol{\mu}}_i^T \boldsymbol{x}_j, \tag{6}$$

where N_e denotes the number of experts (and clusters), and $\boldsymbol{\mu}_i$ represents the centroid of the i^{th} cluster. Each activation vector $\boldsymbol{x}_j$ is assigned to the cluster $c_i = \arg\max_i \ \boldsymbol{\mu}_i^T \boldsymbol{x}_j$ that yields the highest cosine similarity. This process partitions the activation matrix $\boldsymbol{X}$ into N_e clusters $\{\boldsymbol{X}_i\}_{i=1}^{N_e}$ with their corresponding centroids $\{\boldsymbol{\mu}_i\}_{i=1}^{N_e}$, which together form a semantic partition of the dense activation manifold, serving as a basis for initializing the experts and the router.

3.2. Cluster-Aware Expert Initialization

Given the activation clusters $\{\boldsymbol{X}_i\}_{i=1}^{N_e}$, we initialize each expert to specialize in its corresponding cluster. To formalize this, we jointly optimize expert weights $\{\boldsymbol{W}_i\}_{i=1}^{N_e}$ to minimize the within-cluster reconstruction error between each expert's output and that of the dense FFN, parameter-

ized by $\boldsymbol{W}$, while discouraging redundant experts:

$$\min_{\{\boldsymbol{W}_i\}_{i=1}^{N_e}} \sum_{i=1}^{N_e} \Big[\|\boldsymbol{W}\boldsymbol{X}_i - \boldsymbol{W}_i\boldsymbol{X}_i\|_F^2 - \gamma \sum_{j\neq i} \|\boldsymbol{W}\boldsymbol{X}_i - \boldsymbol{W}_j\boldsymbol{X}_i\|_F^2 \Big]. \tag{7}$$

where $\gamma = \frac{1}{N_e - 1}$. The first term encourages each expert to approximate the dense model within its cluster. In contrast, the second term discourages experts from collapsing to similar solutions, thereby reducing redundancy and encouraging more specialized expert behaviors.

In practice, rather than directly optimizing Eq. (7) or fine-tuning experts on their clusters, we initialize expert parameters using a data-aware truncated SVD [40, 42], which preserves principal subspaces associated with each cluster.

Truncated SVD provides a principled low-rank approximation that preserves the leading singular components of a weight matrix. Given an expert weight $\boldsymbol{W}_i$, a standard truncated SVD produces the best rank-r_i approximation $\widetilde{\boldsymbol{W}}_i = T_{r_i}(\mathrm{SVD}(\boldsymbol{W}_i))$, where $T_{r_i}(\cdot)$ denotes the truncation function that retains the top-r_i singular directions. However, this depends solely on the weight matrix and ignores how $\boldsymbol{W}_i$ interacts with the input distribution of each cluster.

To address this limitation, data-aware truncated SVD extends the standard formulation by incorporating input statistics. Specifically, for each cluster $\boldsymbol{X}_i$, a whitening matrix $\boldsymbol{S}_i$ satisfying $\boldsymbol{S}_i\boldsymbol{S}_i^T = \boldsymbol{X}_i\boldsymbol{X}_i^T$ is obtained by Cholesky decomposition, and truncated SVD is then performed on $\boldsymbol{W}_i\boldsymbol{S}_i$:

$$\widetilde{\boldsymbol{W}}_i = T_{r_i}(\mathrm{SVD}(\boldsymbol{W}_i\boldsymbol{S}_i))\boldsymbol{S}_i^{-1}. \tag{8}$$

The rank r_i is defined as the effective rank of $\boldsymbol{W}_i\boldsymbol{S}_i$, where $\sigma_{i,j}$ are its singular values, such that $\sum_{j=1}^{r_i}\sigma_{i,j}^2 / \sum_j \sigma_{i,j}^2 \geq \tau$, thereby retaining at least τ of the total spectral energy.

This formulation ensures that the truncation loss under the data distribution is exactly given by the sum of squared discarded singular values:

$$\left\| \boldsymbol{W}_i\boldsymbol{X}_i - \widetilde{\boldsymbol{W}}_i\boldsymbol{X}_i \right\|_F^2 = \sum_{j>r_i} \sigma_{i,j}^2. \tag{9}$$

This implies that the retained components correspond to the principal directions associated with each cluster, while the discarded components correspond to low-energy directions.

3.3. Cluster-Aware Router Initialization

Since each expert is associated with a cluster, the corresponding cluster centroid provides a natural prior for routing. Thus, we initialize the router parameters $\boldsymbol{W}_r \in \mathbb{R}^{N_e \times d}$ using the ℓ_2-normalized cluster centroids $\{\boldsymbol{\mu}_i\}_{i=1}^{N_e}$ obtained from spherical k-means clustering:

$$\boldsymbol{W}_r = [\boldsymbol{\mu}_1^T; \boldsymbol{\mu}_2^T; \ldots; \boldsymbol{\mu}_{N_e}^T]. \tag{10}$$

This initialization aligns early routing decisions with the underlying semantic structure of the data, allowing experts to receive semantically coherent tokens from the start rather than arbitrary assignments induced by random routing.

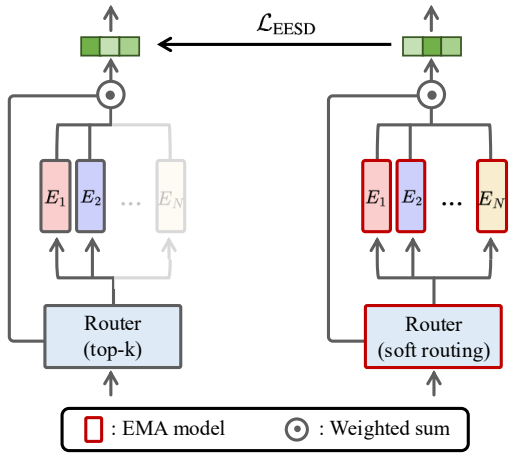


Figure 3. Expert-ensemble self-distillation (EESD). The dense EMA ensemble (right) performs soft routing over all experts and provides stable supervision for the sparse top-k MoE model (left), particularly for tokens with high routing uncertainty.

3.4. Expert-Ensemble Self-Distillation

Routing probabilities reflect how confidently the router assigns each token to experts. Tokens with near-uniform routing probabilities, *i.e.*, high routing uncertainty, indicate weak alignment between input and experts, making it difficult for model to reinforce consistent expert specialization.

To preserve and reinforce the experts' specialization seeded by cluster-aware initialization, we introduce an expert-ensemble self-distillation (EESD) loss, which is motivated by [29]. This loss uses a dense EMA ensemble to provide stable supervision for the sparse MoE model, particularly when routing decisions are uncertain.

Specifically, we construct a teacher model by applying exponential moving average (EMA) updates to the MoE parameters. Unlike the sparse activations in a standard MoE layer, this teacher operates in a dense, full-capacity mode by activating all experts simultaneously:

$$y_{\text{ens}}(\boldsymbol{x}) = \sum_{i=1}^{N_e} g_i^{\text{ema}}(\boldsymbol{x})\, E_i^{\text{ema}}(\boldsymbol{x}), \tag{11}$$

where g_i^{ema} and E_i^{ema} denote EMA router and expert, respectively, whose parameters are updated as $\boldsymbol{W}_{\boldsymbol{i}}^{\text{ema}} \leftarrow \beta \boldsymbol{W}_{\boldsymbol{i}}^{\text{ema}} + (1-\beta)\boldsymbol{W}_{\boldsymbol{i}}$. Since this teacher aggregates the knowledge of all EMA experts, it provides a stable distillation target that the top-k prediction aims to approximate. Recall that, in contrast, the MoE layer prediction is computed using only the top-k activated experts as defined in Eq. (4).

The EESD loss minimizes the discrepancy between the

Table 1. Comparison of upcycling methods on CLIP ViT-B/32 and ViT-B/16 across zero-shot retrieval and classification benchmarks, measured by Recall@1 and Accuracy, respectively. Cluster-aware Upcycling achieves the best performance across most benchmarks.

Model			MSCOCO			ImageNet-1k							VTAB
Arch.	MoE Init	Samples	I→T	T→I	Avg.	Val	V2	A	R	Sketch	ObjNet	Avg.	Natural
ViT-B/32													
Dense	-	4.0B	25.5	42.5	34.0	49.6	41.9	9.7	56.7	34.9	31.0	37.3	52.4
		4.0B+1.3B	30.8	47.5	39.2	56.7	48.5	13.9	64.0	41.2	36.1	43.4	58.3
MoE	Drop-Upcycling [30]	4.0B+1.3B	29.7	46.5	38.1	56.0	47.7	12.9	63.4	40.8	34.3	42.5	57.8
	Sparse Upcycling [20]		30.8	48.0	39.4	57.1	49.1	13.8	64.3	41.8	36.0	43.7	58.0
	CLIP-MoE [48]		29.5	46.8	38.2	56.6	48.1	**14.3**	64.2	41.4	35.7	43.4	58.8
	DeRS-LM [15]		**31.0**	47.7	39.4	56.8	48.6	13.9	64.2	41.1	36.4	43.5	58.1
	Cluster-aware Upcycling		**31.0**	**48.2**	**39.6**	**57.3**	**49.2**	14.0	**65.2**	**42.3**	**36.5**	**44.1**	**59.1**
ViT-B/16													
Dense	-	4.0B	32.5	49.1	40.8	59.4	51.7	20.1	67.3	42.9	39.4	46.8	58.1
		4.0B+1.3B	34.3	50.8	42.6	62.5	54.4	23.5	70.6	45.8	42.5	49.9	62.6
MoE	Drop-Upcycling [30]	4.0B+1.3B	34.1	51.3	42.7	62.0	54.5	22.7	70.8	45.7	42.9	49.8	60.9
	Sparse Upcycling [20]		34.9	50.9	42.9	63.0	**55.1**	23.7	71.2	46.3	42.3	50.3	62.0
	CLIP-MoE [48]		34.0	51.5	42.8	62.9	54.9	**24.5**	71.6	46.2	43.4	50.6	62.8
	Cluster-aware Upcycling		**35.4**	**51.6**	**43.5**	**63.2**	**55.1**	24.1	**72.1**	**46.8**	**43.5**	**50.8**	**63.3**

outputs of the sparse MoE and dense EMA ensemble as:

$$\mathcal{L}_{\text{EESD}} = \frac{1}{T} \sum_{\boldsymbol{x}} \left\| \text{sg}(y_{\text{ens}}(\boldsymbol{x})) - y_{\text{MoE}}(\boldsymbol{x}) \right\|_2^2, \tag{12}$$

where T is the total number of tokens, and $\text{sg}(\cdot)$ denotes the stop-gradient operator, which prevents gradients from flowing into the teacher prediction.

When routing probabilities are nearly uniform, the discrepancy between the mixture of top-k output and the dense ensemble prediction tends to be larger, and the loss provides stronger guidance. Conversely, for confident tokens with sharp routing probabilities, the top-k output closely aligns with the ensemble prediction, so the loss remains small and does not interfere with expert specialization.

The overall training objective combines the task, load-balancing, and EESD losses:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{lb}} \mathcal{L}_{\text{lb}} + \lambda_{\text{EESD}} \mathcal{L}_{\text{EESD}}. \tag{13}$$

4. Experiments

4.1. Implementation Details

We use CLIP [32] ViT-B/16 and ViT-B/32 as our dense baselines. For dense pretraining, we follow the public CLIP configuration trained on LAION-400M [35], with patch dropout 0.5 and 5.3B seen samples over 20 epochs. The global batch size is set to $16K$ for ViT-B/32 and $65K$ for ViT-B/16, with linear warmup for the first 2% of total steps followed by linear decay. The maximum learning rate is 0.005, and the minimum is 0.

For MoE upcycling, we use the dense model parameters from the checkpoint at 15 epochs, corresponding to 4B seen samples, and replace every other FFN with an MoE layer. We use the DeepSpeed-MoE [33] implementation with token-choice, top-2 routing, and 8 experts per layer. The capacity factor is set to 1.5 for MoE layers in the B/16 model and 2.0 in the B/32 model during training, and 2.0 for both at inference. Each upcycled model is trained on LAION-400M for 1.3B seen samples over 5 epochs, using global batch sizes of $16K$ for B/32 and $32K$ for B/16. The learning rate schedule uses 2% linear warmup followed by linear decay to zero, with the peak learning rate set to match the dense model's learning rate at the upcycling checkpoint, following [20]. Optimizer states are reinitialized rather than loaded from the dense model checkpoint. The load-balancing loss coefficient λ_{lb} is set to 0.001.

For clustering, we reduce the activation dimensionality by a factor of eight using PCA, and then perform spherical k-means clustering using Faiss [18] on $128K$ sampled image–text pairs from training set. For expert initialization, we set the effective rank r_i to the smallest value that preserves at least a $\tau = 0.95$ fraction of the spectral energy, while ensuring that r_i is larger than half of the full rank. Cluster-aware expert initialization is applied only to the first linear layer of each expert FFN, as clustering is performed on the FFN input activations and the identified cluster structure is not directly aligned in subsequent layers. For Expert-Ensemble Self-Distillation, we use an EMA coefficient β of 0.999 and set λ_{EESD} to 1.0 for ViT-B/32 and 0.1 for ViT-B/16, excluding padding tokens in the loss computation.

Table 2. ImageNet-1k few-shot and full fine-tuning results for the upcycled ViT-B/16 model. Cluster-aware Upcycling consistently outperforms other upcycling methods.

Model		ImageNet-1k		
Arch.	MoE Init	5-shot	10-shot	FT
Dense	-	50.4	57.1	72.8
MoE	Sparse Upcycling [20]	50.9	57.8	73.0
	Drop-Upcycling [30]	51.1	57.9	73.1
	CLIP-MoE [48]	51.3	58.0	73.2
	Cluster-aware Upcycling	**51.5**	**58.2**	**73.3**

4.2. Evaluation Setup

We evaluate our method using CLIP-Benchmark [3], which covers zero-shot retrieval, zero-shot classification, few-shot classification, and full fine-tuning tasks.

We compare our method with several upcycling baselines. Sparse Upcycling [20] copies the dense FFN weights to all experts while initializing the router randomly. Drop-Upcycling [30] partially reinitializes expert channels to introduce diversity with a randomly initialized router. DeRS-LM [15] employs an expert-shared base weight and represents experts as low-rank matrices. CLIP-MoE [48] introduces a multi-stage contrastive learning strategy tailored for MoE upcycling in CLIP. All methods use the same MoE configuration, dataset, and training schedule as in Section 4.1 to ensure a fair comparison. Experiments are conducted on 64 H200 GPUs.

4.3. Quantitative Results

4.3.1. Zero-shot Results

Table 1 summarizes zero-shot cross-modal retrieval and classification performance for the upcycled ViT-B/32 and ViT-B/16 models.

For the upcycled ViT-B/32 model, Cluster-aware Upcycling achieves the strongest overall performance, while the baselines do not exhibit consistent gains across benchmarks. In Drop-Upcycling, partial reinitialization perturbs the pretrained representation more aggressively, which appears to weaken the benefits of warm-start upcycling and leads to lower overall performance. Sparse Upcycling generally provides competitive results but shows noticeably lower performance on VTAB-Natural, suggesting weaker generalization to out-of-distribution. CLIP-MoE, which is tailored for contrastive learning, achieves strong results on several benchmarks, *e.g.*, ImageNet-A and VTAB-Natural, but its cross-modal retrieval performance falls even below the dense baseline. DeRS-LM achieves competitive results on some benchmarks despite using low-rank experts, but performs worse on average. In contrast, Cluster-aware Upcycling achieves the best performance on most benchmarks.

A similar trend is observed for the ViT-B/16, where the performance gap becomes even more pronounced. These improvements arise from the combined effect of cluster-aware initialization together with EESD, which collectively encourage early expert specialization and maintain it throughout training.

This specialization is consistent with the improved out-of-distribution generalization observed in VTAB-Natural. Moreover, the advantage becomes clearer for larger models and as training progresses (see Supplementary Section B), further supporting the scalability of our approach.

Table 3. Ablation study on cluster-aware initialization and EESD loss for the upcycled ViT-B/16 model.

Model		MSCOCO		ImageNet-1k	
Cluster-init.	EESD	I→T	T→I	Val	10-shot
		34.9	50.9	63.0	57.8
✓		35.1	51.1	**63.2**	58.1
	✓	34.6	51.4	62.7	57.8
✓	✓	**35.4**	**51.6**	**63.2**	**58.2**

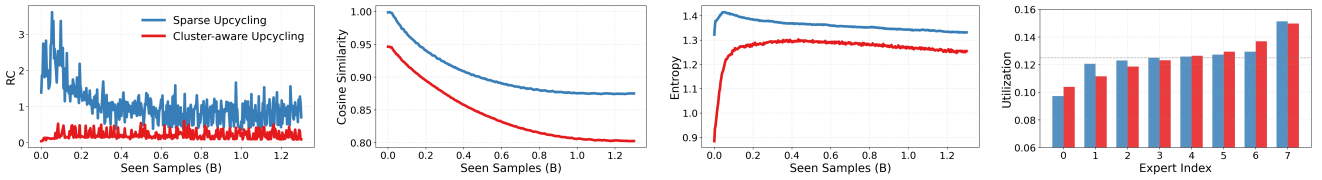
4.3.2. Fine-tuning Results

Table 2 presents 5-shot, 10-shot, and full fine-tuning accuracy on ImageNet for the upcycled ViT-B/16. Cluster-aware Upcycling consistently outperforms comparison methods across all settings. The improvements are most pronounced in few-shot regimes, where initialization quality is critical due to limited training signals. As more labeled data becomes available, the relative advantage becomes smaller but remains consistent, indicating that cluster-aware initialization provides a stronger starting point that persists throughout adaptation. These results suggest that semantic structure at initialization yields expert representations that generalize better not only in zero-shot transfer but also across diverse fine-tuning settings.

4.4. Ablation Study

We evaluate the contribution of the two proposed components, cluster-aware initialization and the EESD loss. As shown in Table 3, cluster-aware initialization alone provides consistent improvements over the Sparse Upcycling baseline across both retrieval and classification metrics. In contrast, the standalone effect of EESD is modest and does not show a clear or consistent improvement trend. However, when combined with cluster-aware initialization, EESD yields further gains across all benchmarks, indicating that the two components play complementary roles. More specifically, cluster-aware initialization establishes a meaningful basis for early specialization by breaking expert symmetry, while EESD helps preserve and reinforce this specialization during training by providing ensemble-level guidance under uncertain routing.

Because the EMA teacher aggregates only the expert



(a) Relative Compactness (b) Expert Similarity (c) Routing Entropy (d) Expert utilization

Figure 4. (a) Relative Compactness measures the overlap between intra- and inter-expert variance, where lower values indicate more disentangled subspaces. (b) Expert similarity shows pairwise cosine similarity between expert weights, with Cluster-aware Upcycling maintaining higher parameter diversity. (c) Routing entropy captures routing uncertainty, with our model achieving lower entropy and more stable expert assignments. (d) Expert utilization patterns across experts, showing balanced routing without routing collapse.

outputs instead of running a full-model ensemble, the EESD loss introduces only modest overhead, approximately 5.3% in wall-clock time and 2.8% in memory in our experiments. This design makes EESD practical even for large-scale MoE architectures, where full model ensembles would otherwise be prohibitive.

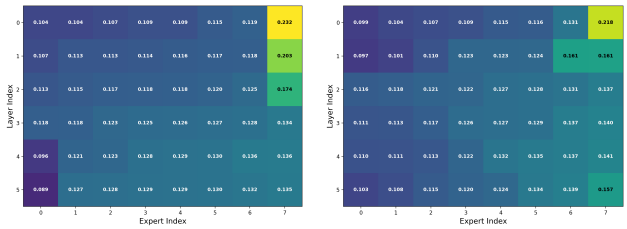
4.5. Analysis

To understand how Cluster-aware Upcycling influences expert specialization, we analyze four aspects of the trained MoE model, relative compactness, expert diversity, routing entropy, and expert utilization, summarized in Figure 4.

Relative compactness To assess how experts structure their feature subspaces during training, we measure the Relative Compactness (RC). Specifically, we compute the within-expert covariance $\Sigma_{\boldsymbol{W}}$ as the covariance of token representations within each expert, and the between-expert covariance $\Sigma_{\boldsymbol{B}}$ as the covariance of the mean output vectors of all experts. RC is then computed as $\mathrm{RC} = \mathrm{Tr}(\Sigma_{\boldsymbol{W}} \Sigma_{\boldsymbol{B}}^{\dagger})$, which measures how strongly the within-expert variance aligns with the between-expert variance directions. Lower RC values indicate that each expert's internal variance lies in directions orthogonal to those of other experts, implying disentangled and non-redundant expert subspaces. As shown in Figure 4a, our model consistently yields lower RC throughout training, indicating geometrically independent expert representations rather than overlapping or redundant ones.

Expert diversity We assess expert diversity through pairwise cosine similarity between expert parameters [27]. As illustrated in Figure 4b, Sparse Upcycling exhibits high similarity among experts, indicating limited diversification. In contrast, Cluster-aware Upcycling shows lower similarity, indicating that the experts are more clearly separated in weight space, consistent with the disentangled latent subspaces captured by relative compactness analysis.

Routing entropy We analyze routing entropy to assess how confidently the model assigns tokens to experts. As shown in Figure 4c, our model begins with low entropy due to the cluster-aware initialization, increases gradually during training as the load-balancing loss encourages exploration, and eventually stabilizes at a lower level. In contrast, the model trained with Sparse Upcycling maintains consistently higher entropy throughout training, indicating less confident and less specialized routing behavior.



(a) Sparse Upcycling (b) Cluster-aware Upcycling

Figure 5. Detailed expert utilization across mixture-of-experts layers in vision encoders. Best viewed in color.

Expert utilization We analyze expert utilization across layers by measuring the fraction of tokens assigned to each expert. As shown in Figure 4d, both Sparse Upcycling and Cluster-aware Upcycling maintain balanced utilization across experts, indicating that no routing collapse occurs. Furthermore, our method exhibits more stable utilization patterns across layers, suggesting that the proposed method enables structured specialization without inducing routing imbalance. The detailed layer-wise patterns provided in Figure 5 further show that Cluster-aware Upcycling maintains slightly more diverse yet still well-balanced expert utilization patterns across layers.

5. Related Work

Mixture-of-Experts Mixture-of-Experts (MoE) architectures enable efficient scaling by activating only a subset of expert networks for each input, rather than processing through all parameters. This conditional computation strategy has proven effective across language models [10, 16, 21, 49], vision [34, 41], and multimodal ar-

chitectures [1, 25, 37]. In a typical MoE layer, a gating network routes each token to a small number of experts that process the input in parallel, while leaving other experts inactive. This approach allows models to scale total parameter count while keeping computational cost per token approximately constant. Recent architectural improvements refine this framework by combining shared and specialized experts [6, 33], introducing hierarchical expert organizations [33], or replacing experts with more efficient representations such as lookup-table-based structures [17].

Despite these advances, MoE systems still face challenges in routing stability and expert specialization. Load-balancing losses promote expert usage but may reduce diversity, motivating techniques such as orthogonality constraints, variance-based regularization [13], and auxiliary-loss–free routing strategies [39]. MoE models also exhibit redundancy among experts, especially when initialized from dense checkpoints, leading to overlapping expert functions. Decomposition-based methods [9, 12, 15, 22, 46] demonstrate that experts can be approximated using shared bases with small residuals, revealing that expert specialization is often weaker than desired.

Mixture-of-Experts Upcycling Sparse Upcycling [20] initializes a sparse MoE model by reusing weights from a pre-trained dense model, rather than training from scratch, substantially reducing training cost. However, since all experts start from identical weights, Sparse Upcycling often results in limited expert diversity and poor specialization. Empirical analysis confirms this tendency, showing that higher expert similarity degrades performance, whereas models trained from scratch maintain low similarity due to random initialization [27, 43]. To mitigate this issue, subsequent works have explored various strategies to enhance expert diversity, including injecting Gaussian noise into router or expert weights [20, 24, 28], permuting or partially reinitializing feed-forward networks [4, 20, 38], and adjusting learning rates across components [20]; however, these approaches fail to bring noticeable improvement in expert specialization. BTX [36] and Nexus [11] instead fine-tune pre-trained dense language models on multiple domains and use them to initialize expert parameters, introducing domain-level specialization among experts. Several works also apply specialized distillation to guide expert divergence after initialization during reinforcement learning stages [14, 26, 44, 47]. Although effective, these methods rely on explicit domain partitioning and several fine-tuning stages to achieve specialization. DeRS-LM [15] employs an expert-shared base weight and represents experts as low-rank matrices; however, it does not address expert symmetry during upcycling. Drop-Upcycling [30] initializes a subset of expert weights with dense weights, while the remaining subset is re-initialized by sampling from a Gaussian distribution parameterized by the original parameters' estimated statistics, partially alleviating the redundancy issue. For vision-language models, CLIP-UP [41] applied Sparse Upcycling to CLIP. CLIP-MoE [48] introduces multi-stage contrastive learning for MoE upcycling, though it is limited to a contrastive learning objective.

Clustering perspective on Mixture-of-Experts MoE architectures can be viewed through the lens of differentiable clustering, where the router assigns tokens to experts according to similarity in representation space. From this viewpoint, experts behave as learnable cluster centroids, and routing defines a soft partition of the token distribution. Earlier work provides theoretical support for this interpretation. Chen et al. [2] shows that gradient descent on nonlinear MoEs can recover latent cluster structure. Chi et al. [4] demonstrates that router logits form manifolds around expert embeddings. Dikkala et al. [7] show that routers initialized from random training samples can recover distinct clusters in well-separated mixtures. Building on this line of thought, ACMoE [31] extends this perspective by introducing an adaptive clustering router that enhances expert specialization by scaling features based on cluster tightness. Collectively, these studies suggest that the geometry of the representation space strongly shapes MoE behavior and that cluster-aware initialization can promote better specialization and training stability, motivating our method.

6. Conclusion

In this work, we introduced Cluster-aware Upcycling, a method that mitigates the expert symmetry problem in MoE upcycling. Rather than replicating pretrained dense weights identically across experts, our method partitions the activation space into semantic clusters and uses them to provide a meaningful basis for expert and router initialization. The data-aware truncated SVD preserves pretrained knowledge within each cluster while promoting diversity across experts, and the cluster-informed router initialization aligns early routing with the underlying representation structure. In addition, EESD preserves and enhances expert specialization by providing ensemble-level supervision, especially when routing remains ambiguous. Experiments on upcycled CLIP models show that Cluster-aware Upcycling consistently improves over baseline upcycling methods on zero-shot and few-shot benchmarks. Our analysis also reveals lower inter-expert similarity, more disentangled expert subspaces, and more specialized yet balanced routing dynamics, demonstrating that leveraging the semantic structure of the pretrained dense model offers a principled path to effective expert specialization in MoE upcycling. These results suggest that incorporating semantic structure into MoE upcycling is a simple yet powerful strategy that may generalizes beyond vision-language pretraining.

Acknowledgements This work was supported in part by the National Research Foundation of Korea (NRF) [RS-2022-NR070855, Trustworthy Artificial Intelligence], Institute of Information & Communications Technology Planning & Evaluation (IITP) [RS2022-II220959 (No.2022-0-00959), (Part 2) Few-Shot Learning of Causal Inference in Vision and Language for Decision Making; RS-2025-25442338, AI Star Fellowship Support Program (Seoul National Univ.); No.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], funded by the Korea government (MSIT). It was also partly supported by AI-driven Scientific Safety e-Report Information Analysis Technology Development Program [RS-2024-00509777, AI-driven Safety e-Reporting and Value Assessment Technology] funded by the Ministry of the Interior and Safety (MOIS).

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv*, 2025. 8

[2] Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. *NeurIPS*, 2022. 8

[3] Mehdi Cherti and Romain Beaumont. CLIP benchmark, 2022. 6

[4] Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, et al. On the representation collapse of sparse mixture of experts. *NeurIPS*, 2022. 1, 8

[5] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv*, 2025. 1

[6] Damai Dai, Chengqi Deng, Chenggang Zhao, Rx Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *ACL*, 2024. 8

[7] Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In *EMNLP*, 2023. 8

[8] 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*, 2024. 1

[9] Chenghao Fan, Zhenyi Lu, Sichen Liu, Chengfeng Gu, Xiaoye Qu, Wei Wei, and Yu Cheng. Make LoRA great again: Boosting LoRA with adaptive singular values and mixture-of-experts optimization alignment. In *ICML*, 2025. 8

[10] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 2022. 1, 2, 7

[11] Nikolas Gritsch, Qizhen Zhang, Acyr Locatelli, Sara Hooker, and Ahmet Üstün. Nexus: Specialization meets adaptability for efficiently training mixture of experts. *arXiv*, 2024. 1, 8

[12] Hao Gu, Wei Li, Lujun Li, Zhu Qiyuan, Mark G Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for moe-based llms compression. In *ICML*, 2025. 8

[13] Hongcan Guo, Haolang Lu, Guoshun Nan, Bolun Chu, Jialin Zhuang, Yuan Yang, Wenhao Che, Sicong Leng, Qimei Cui, and Xudong Jiang. Advancing expert specialization for better moe. *arXiv*, 2025. 8

[14] Ailin Huang, Ang Li, Aobo Kong, Bin Wang, Binxing Jiao, Bo Dong, Bojun Wang, Boyu Chen, Brian Li, Buyun Ma, et al. Step 3.5 flash: Open frontier-level intelligence with 11b active parameters. *arXiv*, 2026. 8

[15] Yongqi Huang, Peng Ye, Chenyu Huang, Jianjian Cao, Lin Zhang, Baopu Li, Gang Yu, and Tao Chen. DeRS: Towards extremely efficient upcycled mixture-of-experts models. In *CVPR*, 2025. 5, 6, 8, 2

[16] 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, et al. Mixtral of experts. *arXiv*, 2024. 7

[17] Shibo Jie, Yehui Tang, Kai Han, Yitong Li, Duyu Tang, Zhi-Hong Deng, and Yunhe Wang. Mixture of lookup experts. In *ICML*, 2025. 8

[18] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019. 5

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv*, 2020. 1

[20] Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse Upcycling: Training mixture-of-experts from dense checkpoints. In *ICLR*, 2023. 1, 2, 5, 6, 8

[21] 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 *ICLR*, 2021. 1, 2, 7

[22] Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G Lee, Shengjie Sun, Wei Xue, and Yike Guo. MoE-SVD: Structured mixture-of-experts llms compression via singular value decomposition. In *ICML*, 2025. 8

[23] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *CVPR*, 2023. 1

[24] Seng Pei Liew, Takuya Kato, and Sho Takase. Scaling laws for upcycling mixture-of-experts language models. In *ICML*, 2025. 8

[25] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. MoE-LLaVA: Mixture of experts for large vision-language models. *CoRR*, 2024. 8

[26] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv*, 2025. 8

[27] Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models. In *NAACL Findings*, 2025. 7, 8, 2

[28] Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoe: Open mixture-of-experts language models. In *ICLR*, 2025. 1, 8

[29] Jonghwan Mun, Kimin Lee, Jinwoo Shin, and Bohyung Han. Learning to specialize with knowledge distillation for visual question answering. In *NIPS*, 2018. 2, 4

[30] Taishi Nakamura, Takuya Akiba, Kazuki Fujii, Yusuke Oda, Rio Yokota, and Jun Suzuki. Drop-upcycling: Training sparse mixture of experts with partial re-initialization. In *ICLR*, 2025. 1, 5, 6, 8, 2

[31] Stefan Nielsen, Rachel Teo, Laziz Abdullaev, and Tan Minh Nguyen. Tight clusters make specialized experts. In *ICLR*, 2025. 8

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 5

[33] Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. In *ICML*, 2022. 5, 8

[34] 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. *NeurIPS*, 2021. 7

[35] Christoph Schuhmann, Robert Kaczmarczyk, Aran Komatsuzaki, Aarush Katta, Richard Vencu, Romain Beaumont, Jenia Jitsev, Theo Coombes, and Clayton Mullis. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. In *NeurIPS Workshop Datacentric AI*, 2021. 5

[36] Sainbayar Sukhbaatar, Olga Golovneva, Vasu Sharma, Hu Xu, Xi Victoria Lin, Baptiste Roziere, Jacob Kahn, Shang-Wen Li, Wen-tau Yih, Jason E Weston, et al. Branch-train-mix: Mixing expert llms into a mixture-of-experts llm. In *CoLM*, 2024. 1, 8

[37] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv*, 2025. 8

[38] Qwen Team et al. Qwen2 technical report. *arXiv*, 2024. 1, 8

[39] Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv*, 2024. 8

[40] Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. SVD-LLM V2: Optimizing singular value truncation for large language model compression. In *NAACL*, 2025. 4

[41] Xinze Wang, Chen Chen, Yinfei Yang, Hong-You Chen, Bowen Zhang, Aditya Pal, Xiangxin Zhu, and Xianzhi Du. Clip-up: A simple and efficient mixture-of-experts clip training recipe with sparse upcycling. *arXiv*, 2025. 7, 8

[42] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *ICLR*, 2025. 4

[43] Tianwen Wei, Bo Zhu, Liang Zhao, Cheng Cheng, Biye Li, Weiwei Lü, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Liang Zeng, et al. Skywork-MoE: A deep dive into training techniques for mixture-of-experts language models. *arXiv*, 2024. 8

[44] Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-Flash technical report. *arXiv*, 2026. 8

[45] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv*, 2025. 1

[46] Shen Yuan, Yin Zheng, Taifeng Wang, Binbin Liu, and Hongteng Xu. MoORE: SVD-based model MoE-ization for conflict- and oblivion-resistant multi-task adaptation. In *NeurIPS*, 2025. 8

[47] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv*, 2025. 8

[48] Jihai Zhang, Xiaoye Qu, Tong Zhu, and Yu Cheng. CLIP-MoE: Towards building mixture of experts for clip with diversified multiplet upcycling. In *EMNLP*, 2025. 5, 6, 8, 2

[49] 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*, 2022. 7