ExPLoRe: Expert Patch-Level Loss Routing for Multi-Objective Masked Image Modeling

Konstantinos Georgiou, Maofeng Tang, and Hairong Qi

Min H. Kao Department of Electrical Engineering and Computer Science,
The University of Tennessee, Knoxville, TN 37996, USA
{kgeorgio, mtang4}@vols.utk.edu, hqi@utk.edu

Abstract. Multi-objective masked image modeling (MIM) combines complementary learning signals (token distillation, CLS alignment, and pixel reconstruction) but existing methods weight these objectives with global scalars, ignoring spatial heterogeneity across patches. We present ExPLoRe (Expert Patch-Level Loss Routing), which repurposes Soft Mixture of Experts (MoE) dispatch weights as learned, per-patch loss coefficients. The key mechanism is *loss-coupling*: allowing loss gradients to flow through dispatch weights to the router enables content-dependent specialization, where different patches receive different emphases across objectives. A detach ablation confirms loss-coupling as the core mechanism, degrading performance by 1.6% when gradients are blocked. On ImageNet-1K with ViT-Base, ExPLoRe improves over non-MoE baselines on two objective combinations (Token+CLS: +0.5% k-NN, +4.4% linear probe; Token+Pixel: +2.2% k-NN), achieving 80.6% linear probe and 85.3% finetuning accuracy, competitive with published methods. For downstream transfer, we develop adaptation recipes (Freeze Routing, Expert Dropout, and Freeze Attention) that improve MoE finetuning by +1.5% over the vanilla MoE, and close a 2.5–2.9 mIoU segmentation gap so that MoE models match or exceed non-MoE baselines on ADE20K.

Keywords: Masked Image Modeling · Mixture of Experts · Self-Supervised Learning · Knowledge Distillation · Multi-Task Learning

1 Introduction

Masked image modeling (MIM), inspired by BERT's masked language modeling [10], has emerged as a powerful self-supervised learning paradigm for vision [3, 15]. Teacher-guided approaches [12, 16, 22] leverage frozen feature extractors such as CLIP [25] to provide semantic targets for student encoders. Recent works combine multiple learning signals including token-level distillation, CLS alignment, and pixel reconstruction; but how to weight these diverse objectives optimally remains an open question that is largely independent of the specific teacher used.

The Patch-Level Heterogeneity Problem. Existing multi-task learning methods [8, 19] balance losses at the *global* level, applying a single scalar weight

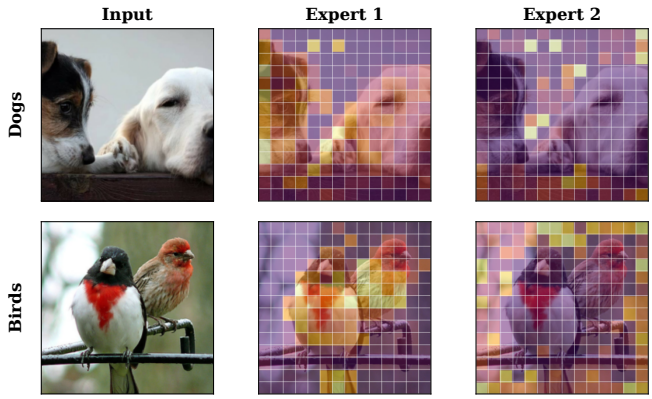

Fig. 1: Expert dispatch-weight visualization. Per-patch dispatch weights from a trained 2-expert ExPLoRe model overlaid on input images (warm = high weight, cool = low weight). The two experts learn complementary spatial specialization without explicit supervision: one expert assigns higher loss emphasis to foreground regions while the other focuses on background and context.

per objective uniformly across all spatial locations. However, different image regions benefit from different learning signals: semantically rich patches are better served by teacher distillation, while texture-heavy background regions benefit from pixel reconstruction. Global methods cannot capture this spatial variation. As we demonstrate, gradient-based balancing (GradNorm [8]) can even fail catastrophically when objectives have spatially heterogeneous importance. To our knowledge, no prior work addresses *patch-level* loss weighting for masked image modeling.

We present ExPLoRe[1] (**Ex**pert **P**atch-**L**evel **Lo**ss **R**outing), which addresses patch-level heterogeneity by repurposing Soft Mixture of Experts (Soft-MoE) [24] dispatch weights as learned, per-patch loss coefficients. Unlike traditional MoE methods that use routing for capacity scaling [13, 26, 28] or task-specific expert assignment [5, 33], ExPLoRe uses continuous, sample-adaptive weighting through a *loss-coupling* mechanism: gradients from each loss objective flow back through the dispatch weights to the router, enabling the model to learn content-dependent specialization. We evaluate two objective combinations (*Token+CLS* and *Token+Pixel*) with expert scaling from 2 to 64 under sparse encoding [15], and develop downstream transfer recipes for competitive finetuning. Figure 1 visualizes the resulting routing patterns.

ExPLoRe provides content-dependent per-patch loss emphasis that prior methods lack; competitive results across classification and dense prediction demonstrate this mechanism does not sacrifice representation quality. Our contributions center on a single mechanism with two supporting analyses:

– **Core contribution: dispatch-weight loss weighting.** We introduce a mechanism that repurposes Soft-MoE dispatch weights as per-patch loss co-

[1] Code available at https://github.com/aicip/ExPLoRe.

efficients for multi-objective MIM. Loss-coupling, where loss gradients flow through dispatch weights to update the router, is the core innovation enabling learned per-patch specialization. We demonstrate consistent improvements over non-MoE baselines across two objective combinations, achieving 80.6% linear probe accuracy competitive with published methods (Table 6).

- **Routing dynamics analysis.** We empirically characterize routing behavior: entropy regularization governs a sharp transition between stable routing and catastrophic collapse; optimal regularization is expert-count dependent; and a detach ablation validates loss-coupling as the mechanism driving specialization (Sections 4.2 and 4.2).
- **Downstream transfer recipes.** We develop complementary adaptation strategies, Freeze Routing (FR), Expert Dropout (ExD), and Freeze Attention (FA), that compose naturally. The combined FR+FA+ExD recipe exceeds the non-MoE baseline on both classification (+0.5%) and semantic segmentation, closing a 2.5–2.9 mIoU gap (Section 4.4).

2 Related Work

Masked Image Modeling. MIM extends masked language modeling (MLM) from NLP to vision. BEiT [3] uses *dense encoders* processing all patches with mask tokens, while MAE [15] uses *sparse encoders* processing only visible patches with shuffling for efficiency. SimMIM [32] uses simple linear projections, iBOT [37] combines masking with self-distillation, and MaskFeat [29] uses HOG features as targets.

Self-distillation approaches include data2vec [1, 2] (EMA representations), CAE/CAEv2 [6, 35] (context autoencoding), SdAE [7] (pixel + self-distillation), and BootMAE [11] (bootstrapping).

Teacher-Guided MIM with CLIP. CLIP [25] enables MIM methods with semantic targets. MaskDistill [22] provides a unified view of masked image modeling, formulating MIM as $\mathcal{L}(\mathcal{N}(\mathcal{T}(I)), \mathcal{H}(\mathcal{S}(I_m)))$ with teacher $\mathcal{T}$, student $\mathcal{S}$, projector heads $\mathcal{N}/\mathcal{H}$, and shows CLIP features outperform pixels or discrete tokens as reconstruction targets. Our base architecture builds upon MaskDistill's framework. BEiT v2 [21] combines tokenizers with CLIP distillation. MILAN [16] uses CLIP attention for semantic masking and combines CLS with patch distillation. MaskCLIP [12] explores masked self-distillation within CLIP.

Multi-Task Learning and Loss Balancing. Multiple methods balance competing objectives: uncertainty weighting [18] models task-dependent uncertainty, GradNorm [8] normalizes gradient magnitudes, PCGrad [34] projects conflicting gradients, MGDA [27] seeks Pareto-optimal solutions, and random weighting [19] samples task weights stochastically. Crucially, all operate at the *global* level, applying a single scalar weight per objective uniformly across all spatial locations. This ignores the spatial structure of images, where different regions may benefit from different loss emphasis. To our knowledge, no prior work addresses per-patch adaptive loss weighting for masked image modeling, which we identify as a key gap.

Mixture of Experts. Sparse MoE methods [13, 28] route tokens to expert subsets for conditional computation but introduce load balancing challenges. Soft-MoE [24] addresses this through fully-differentiable soft routing, where dispatch and combine weights computed via softmax avoid discrete decisions and their associated training difficulties. V-MoE [26] applied sparse MoE to vision transformers; subsequent work [14, 20] compared routing strategies, and multi-task applications [5, 33] demonstrated MoE effectiveness in vision. CR-MoE [17] first applied MoE to self-supervised learning (contrastive), discovering routing consistency challenges. However, applications to masked image modeling with dynamic task weighting have not been investigated. MoE downstream transfer remains understudied; ViMoE [14] finds that only 2–5 MoE layers suffice but focuses on supervised training. In contrast to prior work on capacity scaling [13, 26, 28] or task-specific expert assignment [5, 33], ExPLoRe repurposes Soft-MoE dispatch weights for *per-patch loss weighting* through continuous, loss-coupled routing, and addresses the MoE transfer gap with dedicated adaptation recipes (Section 4.4).

3 Method

ExPLoRe uses Soft-MoE to learn per-patch loss weighting in multi-objective masked image modeling, repurposing dispatch weights as dynamic loss coefficients (Figure 2). Given an input image divided into patches with a binary mask partitioning them into visible set $\mathcal{V}$ and masked set $\mathcal{M}$, the framework consists of: (1) a student encoder processing visible patches, (2) a frozen CLIP teacher providing semantic targets, and (3) optional auxiliary networks (decoder for pixel reconstruction, projection heads for CLS alignment).

3.1 Multi-Objective Learning Framework

Our framework combines three complementary objectives at different spatial scales. Per-image losses are defined below; MoE-weighted variants (Section 3.2) average over the batch.

Token-Level Distillation: Distills CLIP teacher features $\mathbf{t}$ into student predictions $\hat{\mathbf{t}}$ using Huber loss ($\beta = 1.0$) with layer-normalized teacher features:

$$\mathcal{L}_{\text{token}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \ell_{\text{huber}}(\hat{\mathbf{t}}_i, \text{LN}(\mathbf{t}_i)) \tag{1}$$

Global CLS Alignment: Aligns CLS tokens of student and teacher:

$$\mathcal{L}_{\text{cls}} = 1 - \frac{\hat{\mathbf{z}}_{\text{cls}} \cdot \mathbf{t}_{\text{cls}}}{\|\hat{\mathbf{z}}_{\text{cls}}\| \, \|\mathbf{t}_{\text{cls}}\|} \tag{2}$$

Pixel Reconstruction: A decoder reconstructs normalized pixel values:

$$\mathcal{L}_{\text{pixel}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \|\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\|_2^2 \tag{3}$$

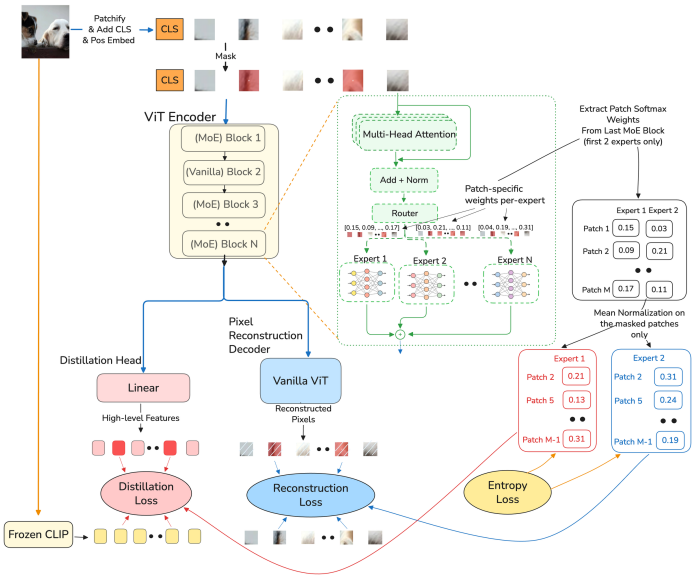

Fig. 2: ExPLoRe Framework Overview. Soft Mixture of Experts (Soft-MoE) is integrated into the student encoder for patch-level adaptive loss weighting. The student encoder (ViT-Base with alternating MoE blocks at layers {1,3,5,7,9,11}) processes patches while a frozen CLIP teacher provides semantic targets. Soft-MoE dispatch weights $\mathbf{D}$ serve as per-patch loss coefficients: each expert weights a different training objective. The two routing weight types play distinct roles: dispatch weights $\mathbf{D}$ (normalized over patches per expert) form the loss-coupling pathway that carries loss gradients back to the router, whereas combine weights $\mathbf{C}$ (normalized over experts per patch) only mix expert outputs in the forward pass. Loss-coupling, where loss gradients flow through $\mathbf{D}$ to the router, is the key mechanism enabling learned specialization.

where $\hat{\mathbf{x}} = d_\omega(\mathbf{z})$ are reconstructed patches and $\bar{\mathbf{x}}$ are normalized targets. ExPLoRe uses Soft MoE dispatch weights as learned per-patch loss coefficients for the combined objective (Section 3.2).

Encoding Strategy and Decoder. We adopt sparse encoding (MAE-style [15]) for our main model, processing only visible patches for efficiency. Sparse encoding yields dispatch weights only for visible patches, determining which losses can be MoE-weighted (Section 3.2). Dense encoding (BEiT-style [3]) is evaluated as an ablation. A lightweight 8-block decoder reconstructs masked patches; full details in supplementary Section E.

3.2 Soft Mixture of Experts for Patch-Level Dynamic Loss Weighting

Global methods apply uniform weights across patches, but different patches benefit from different signals: semantic regions need distillation, texture regions need reconstruction. We integrate Soft-MoE [24] into the student encoder for learned

patch-level adaptive weighting (Figure 2). The central idea is that Soft-MoE produces two types of weights from shared logits: *dispatch weights* $\mathbf{D}$ (softmax over patches, normalized per expert) that determine how patches contribute to each expert, and *combine weights* $\mathbf{C}$ (softmax over experts, normalized per patch) that determine the output mixture. We repurpose dispatch weights, rather than combine weights, as loss coefficients, for reasons detailed below.

Architecture and Routing Mechanism: We adopt the Soft-MoE formulation [24] (full derivation in supplementary Section G). Soft-MoE replaces selected MLP layers with E expert networks, each a standard MLP with independent parameters. Given patch representations $\mathbf{X} \in \mathbb{R}^{B\times N\times d}$, learnable expert parameters $\mathbf{\Phi} \in \mathbb{R}^{d\times E}$, and a learned scale s, routing logits are computed as:

$$\mathbf{L} = s \cdot \frac{\mathbf{X}}{\|\mathbf{X}\|_2} \cdot \frac{\mathbf{\Phi}}{\|\mathbf{\Phi}\|_2} \tag{4}$$

These shared logits yield two weight types via softmax along different dimensions: *dispatch weights* $\mathbf{D} = \mathrm{softmax}_{\mathrm{patches}}(\mathbf{L})$ with $\sum_n \mathbf{D}_{b,n,e} = 1$ per expert, and *combine weights* $\mathbf{C} = \mathrm{softmax}_{\mathrm{experts}}(\mathbf{L})$ with $\sum_e \mathbf{C}_{b,n,e} = 1$ per patch. Dispatch weights aggregate patches into expert inputs $\mathbf{S}_{\mathrm{in}} = \mathbf{D}^\top \mathbf{X}$, each expert processes its slot, and combine weights reconstruct per-patch outputs $\mathbf{Y} = \mathbf{C} \cdot \mathbf{S}_{\mathrm{out}}$. The critical distinction for loss weighting is that dispatch weights are normalized over patches (per expert) while combine weights are normalized over experts (per patch).

Dispatch-Weight-Based Loss Weighting: Our key innovation is using dispatch weights $\mathbf{D}$ (not combine weights) for patch-level loss weighting. Unlike combine weights which represent per-token expert mixtures with constraint $\sum_e \mathbf{C}_{b,n,e} = 1$ (normalized over experts for each patch), dispatch weights represent how much each patch contributes to each expert with constraint $\sum_n \mathbf{D}_{b,n,e} = 1$ (normalized over patches for each expert). This expert-centric perspective prevents a critical degeneracy issue: with combine weights, the router could minimize loss by setting all weights to zero for difficult patches, causing the loss to vanish. With dispatch weights, the normalization constraint over patches prevents this collapse, made concrete by the per-image normalization in Eq. 5 below. The router must distribute each expert's "attention" across patches and cannot zero out the entire loss.

For our main configuration (Token+CLS, sparse encoding), Expert 0 weights the token distillation loss on visible patches $\mathcal{V}$. Each patch's contribution is scaled by its dispatch weight, averaged over the batch:

$$\mathcal{L}_{\mathrm{token}}^{\mathrm{MoE}} = \frac{1}{B}\sum_{b=1}^{B} \frac{\sum_{i\in\mathcal{V}} \mathbf{D}_{b,i,0} \cdot \ell_{\mathrm{huber}}(\hat{\mathbf{t}}_{b,i}, \mathrm{LN}(\mathbf{t}_{b,i}))}{\sum_{i\in\mathcal{V}} \mathbf{D}_{b,i,0}} \tag{5}$$

Per-image normalization (dividing by the sum of dispatch weights) computes a weighted average for each image independently, preventing the router from minimizing loss by scaling weights down across the batch.

When $E > K$ (more experts than loss-weighted objectives), only K experts receive direct loss-coupling; the rest provide MoE capacity through forward-pass

gradients and entropy regularization (supplementary Section H). The pixel loss variant applies analogous weighting (supplementary Section D); this requires dense encoding since sparse mode produces dispatch weights only for visible patches $\mathcal{V}$, while pixel loss is computed on masked patches $\mathcal{M}$.

Entropy Regularization for Uniform Token Distribution: While dispatch weights prevent loss degeneracy, they don't guarantee balanced expert utilization. Without regularization, one expert might receive contributions from all patches while another receives none, leading to expert collapse. Per-expert entropy regularization encourages uniform token distributions.

For each expert e, we compute the entropy of its dispatch-weight distribution across patches:

$$H_e = -\frac{1}{B}\sum_{b=1}^{B}\sum_{n=1}^{N} p_{b,n,e}\log p_{b,n,e} \tag{6}$$

where $p_{b,n,e} = \mathbf{D}_{b,n,e}$, which already forms a distribution over patches per expert since Soft-MoE normalizes dispatch weights over the token axis ($\sum_n \mathbf{D}_{b,n,e} = 1$). Higher entropy indicates a more uniform distribution across patches.

The entropy loss encourages high entropy (uniformity) through minimization:

$$\mathcal{L}_{\text{entropy}} = -\sum_{e=0}^{E-1} \lambda_e \cdot H_e \tag{7}$$

where λ_e are per-expert weights. For Token+CLS, we use a symmetric weight λ for all experts; through hyperparameter search, we find $\lambda = 5.0$ provides stable training as default, while $\lambda = 0.5$ is optimal for 2 experts. Critically, optimal entropy weight is *expert-count dependent*: $\lambda = 0.5$ improves 2-expert routing but degrades 64-expert performance (Section 4.2). Token+Pixel requires asymmetric per-expert weights (supplementary Section C). Regularization is computed at the last MoE block (Block 11), validated via block selection ablation (Section 4.2). Entropy regularization and loss-coupling are opposing forces: entropy pushes dispatch weights toward uniformity (no specialization), while loss-coupling pushes them toward content-dependent specialization; the coefficient λ governs the trade-off, with collapse at $\lambda = 0.01$, stable specialization at the default, and over-regularization (degradation) at $\lambda \gg 5$ (Section 4.2).

MoE Placement and Implementation: MoE replaces MLP layers in alternating blocks {1, 3, 5, 7, 9, 11} of ViT-Base, balancing capacity (6 MoE blocks) with efficiency (6 standard blocks) at approximately 1.5× computation. Expert networks use hidden dimension 3072 with GELU activation; $E = 2$ experts increase parameters by 35% (86M→116M), while $E = 64$ reaches 1.86B. We use $S = 1$ slot per expert for direct expert-to-loss mapping. Full design rationale, memory analysis, and dense/sparse integration details are in the supplementary material (Sections D–G).

Loss-Coupling Mechanism: Because dispatch weights are differentiable, gradients from each loss objective flow back through $\mathbf{D}$ to the router parameters $\mathbf{\Phi}$ and s, creating a *loss-coupling* effect that enables learned specialization. A detach ablation confirms this as the core mechanism (Section 4.2).

Table 1: MoE loss weighting configurations. Which losses receive dispatch-weight modulation in each configuration. In sparse encoding, dispatch weights exist only for visible patches, so pixel loss (computed on masked patches) cannot be MoE-weighted.

Configuration	Encoding	Token Loss	CLS Loss	Pixel Loss
Token+CLS	Sparse	MoE (Exp. 0)	Global w_{cls}	—
Token+Pixel	Sparse	MoE (Exp. 0)	—	Uniform

Total Training Objective: Consolidating the per-objective losses (Eqs. 1–3) under dispatch-weight coupling, the full training objective is

$$\mathcal{L}_{\text{total}} = \sum_{e} w_e \cdot \frac{1}{|\mathcal{P}_e|} \sum_{n \in \mathcal{P}_e} \mathbf{D}_{b^\star,n,e} \cdot \mathcal{L}_e(n), \tag{8}$$

where $\mathcal{L}_e(n)$ is the per-patch loss for objective e over its patch set $\mathcal{P}_e$, w_e the global per-objective weight, and $\mathbf{D}_{b^\star,n,e}$ the dispatch weight from loss block $b^\star$ (Eq. 5; $\mathbf{D} \equiv 1$ for un-coupled objectives). The product $\mathbf{D}_{b^\star,n,e}\,\mathcal{L}_e(n)$ is the per-patch loss coefficient named in the abstract; entropy regularization (Eq. 7) is added during training.

4 Experiments

4.1 Experimental Setup

Dataset. All experiments use ImageNet-1K [9] (1.28M training images, 1000 classes).

Pretraining. ViT-Base/16 student encoder with frozen CLIP ViT-B/16 teacher, 300 epochs, batch size 4096 on 4× H100 GPUs with block masking at 40% ratio and sparse encoding. Full hyperparameters are in supplementary Tables A1–A2.

Evaluation. We evaluate across four complementary tasks: (1) **k-NN** ($k = 20$, cosine similarity) as a training-free representation quality metric; (2) **Linear probe** on frozen [CLS] features; (3) **ImageNet finetuning** with layer-wise LR decay [3]; and (4) **Semantic segmentation** on ADE20K [36] with UperNet [31]. MoE models use the adaptation recipes from Section 4.4. Full evaluation protocols are in supplementary Section A.3.

Model Configurations. We evaluate two multi-objective combinations, summarized in Table 1. Token+CLS is our primary configuration because it achieves the highest downstream accuracy and supports the full expert scaling range (2–64) without the inverted scaling observed in Token+Pixel at 64 experts.

The Multi-Objective Challenge. Existing global weighting methods fail on spatially heterogeneous MIM objectives. Despite an extensive hyperparameter search (supplementary Table A3), GradNorm [8] cannot match its own static baseline (72.8% vs. 74.2% k-NN for Token+CLS) and collapses entirely on Token+Pixel (32.5%), underperforming even its own dense baseline. Random loss

Table 2: MoE design ablations on 2-expert Token+CLS (k-NN@20). Δ relative to baseline ($\lambda = 5.0$, Block 11).

Entropy Reg.	$\lambda = 0.5$	$\lambda = 5.0$	$\lambda = 0.1$	$\lambda = 0.01$
k-NN@20	**75.5**	75.4	75.3	66.9
Δ	+0.1	—	−0.1	−8.5
Mech. Ablations Combine reg. Importance ($w = 2$). Detach weights. No entropy				
k-NN@20	75.4	75.3	73.8	2.1
Δ	+0.0	−0.1	−1.6	−73.3
Loss Block Sel.	Block 11	Block 9	Block 7	Block 5
k-NN@20	75.4	75.3	75.4	75.2
Δ	—	−0.1	+0.0	−0.2

weighting (RLW) [19] helps modestly (75.8% k-NN) but applies uniform per-patch weights, unable to capture spatial variation.

These failures motivate per-patch loss weighting: different image regions benefit from different loss emphasis, and global methods that apply a single scalar weight per objective cannot capture this spatial variation.

4.2 MoE Mechanism Validation

Having established the need for per-patch loss weighting, we validate the core design decisions of our MoE approach. Table 2 summarizes ablations on the 2-expert Token+CLS configuration.

Dispatch weights prevent degeneracy. Combine-weight loss weighting collapses to 2.1% k-NN (−73.3 points), confirming the dispatch-weight design choice (Section 3.2).

Loss-coupling is the core mechanism. We validate loss-coupling through a detach ablation: computing MoE-weighted losses with dispatch weights detached from the computation graph (i.e., treated as fixed constants during backpropagation so that loss gradients cannot update the router). This degrades k-NN by 1.6% (Table 2), confirming that the router's ability to receive gradient signal from loss objectives is essential for learned specialization. The result is reinforced by the 2-expert unweighted variant (74.1% k-NN, Table 3), which *underperforms* the non-MoE baseline (75.7%) by 1.6%: MoE capacity alone is not merely insufficient but actively harmful without loss-coupling, making the coupling mechanism essential rather than incremental.

Importance loss targets the wrong distribution. Importance-based regularization ($w = 2$) penalizes dispatch-weight imbalance across experts, but Soft-MoE's softmax already ensures balanced dispatch. The resulting scale parameter collapse (s: 0.90 → 0.22 at the loss block) degrades performance by 0.1%. Supplementary Section D provides visualizations of dispatch-weight patterns across all six MoE blocks, confirming spatially coherent, content-dependent routing.

4.3 Expert Scaling Analysis

Table 3 and Figure 3 present pretraining results across expert counts for both objective combinations. We include our MaskDistill [22] reproduction (token distillation only, dense encoding) as a reference point. For *Token+CLS*, starting from this MaskDistill baseline (75.6% k-NN), adding CLS alignment without MoE improves k-NN to 75.7%. Introducing 2-expert ExPLoRe with dispatch-weight loss weighting reveals an instructive pattern: k-NN decreases slightly to 75.4%, but linear probe accuracy jumps to **79.6%** (+3.4% over No MoE), indicating that the routing mechanism reshapes the feature space toward improved linear separability.

k-NN scales monotonically from 2 to 64 experts, reaching **76.2%** at 64 experts, with smooth convergence across all configurations (Figure 3a). Linear probe, evaluated at the 2- and 64-expert endpoints, shows the same trend: 79.6% → **80.6%**. Critically, comparing weighted vs. unweighted variants *at the same expert count* isolates loss weighting from the parameter effect (Figure 3b). The 2-expert unweighted variant (74.1%) demonstrates that loss weighting provides the +1.3% gain that makes ExPLoRe competitive. The 64-expert unweighted variant (75.3%) confirms that loss weighting contributes +0.9% at this scale. The relative gain decreases from +1.3% (2 experts) to +0.9% (64 experts) as inherent diversity from more expert parameters provides partial dispatch variation independently of loss-coupling.

For *Token+Pixel*: no MoE (71.5%) → 32-expert without dispatch weighting (73.0%, +1.5%) → 32-expert with dispatch weighting (**73.7%**, +2.2% total). Dispatch weighting adds +0.7% over the unweighted variant at 32 experts, confirming the mechanism's effectiveness across objective combinations. This +2.2% total improvement is substantially larger than Token+CLS's +0.5%, indicating that dispatch-weight loss weighting is particularly effective when objectives have different spatial characteristics.

However, Token+Pixel exhibits an inverted scaling pattern: k-NN peaks at 32 experts then drops at 64 experts (69.3%), a phenomenon not observed in Token+CLS. We attribute this to weaker gradient signals from pixel reconstruction, which may be insufficient to train 64 expert sets. MoE variants use the default pixel loss weight ($w_{\text{pixel}} = 1.0$); the non-MoE baseline uses tuned weighting ($w_{\text{pixel}} = 0.01$). While this 100× difference could partially explain the Token+Pixel gap, the comparison isolates the MoE contribution: dispatch weights provide implicit loss modulation that partially substitutes for manual weight tuning, and the weighted vs. unweighted comparison (+0.7% at 32 experts) controls for this confound since both use $w_{\text{pixel}} = 1.0$.

Parameter cost. Expert scaling increases parameters substantially: 86M (no MoE) → 116M (2 experts, +35%) → 1.86B (64 experts, 21.6×). The weighted vs. unweighted comparisons at matched expert counts (Table 3) isolate loss weighting from the parameter effect; a detailed mechanism isolation analysis is in supplementary Section I. We view the 2-expert configuration as the practical recommendation (35% parameter increase for 79.6% linear probe), while

Table 3: Expert scaling analysis. Token+CLS scales smoothly to 64 experts (see also Figure 3a); Token+Pixel peaks at 32. All MoE models use sparse encoding, 300 epochs. W=dispatch-weight loss weighting. ⋆ Dense encoding (BEiT-style [3]). † Tuned pixel loss weight ($w_{\text{pixel}}=0.01$).

Token + CLS	MaskDistill⋆	No MoE	2	2+W	64	**64+W**
k-NN@20	75.6	75.7	74.1	75.4	75.3	**76.2**
Linear	75.7	76.2	—	79.6	—	**80.6**

Token + Pixel	No MoE†	32	**32+W**	64+W
k-NN@20	71.5	73.0	**73.7**	69.3
Linear	—	—	79.4	—

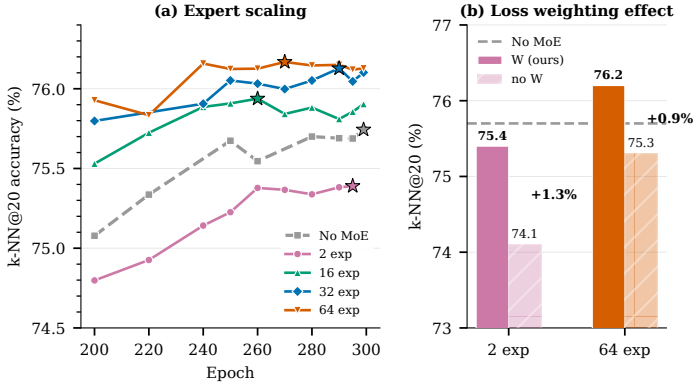

Fig. 3: Expert scaling and mechanism isolation (Token+CLS). **(a)** k-NN@20 trajectories over epochs 200–300 for No MoE and 2/16/32/64-expert configurations (all with dispatch-weight loss weighting). Stars mark peak accuracy per configuration; more experts yield higher final accuracy. **(b)** Mechanism isolation: weighted (W) vs. unweighted (no W) at 2 and 64 experts. Dispatch-weight loss weighting contributes +1.3% at 2 experts and +0.9% at 64 experts at matched parameter counts, confirming the mechanism's effect beyond the parameter contribution.

the 64-expert scale demonstrates that dispatch-weight loss weighting improves representation quality at scale.

Entropy Regularization and Stability Transitions. We sweep entropy regularization weight λ on the 2-expert Token+CLS configuration (Table 2). We use two diagnostics: the dispatch-weight coefficient of variation (CV) across patches, and the *weighted-to-unweighted loss ratio* (MoE-weighted loss divided by the same loss computed with uniform weights; 1.0 indicates neutral routing, $\ll 1.0$ indicates the router is redistributing weight away from difficult patches). A sharp transition exists between $\lambda = 0.01$ (collapsed, 66.9% k-NN) and $\lambda = 0.1$ (stable, 75.3%): CV jumps 32× and the loss ratio drops to 0.13, indicating deceptive loss minimization rather than meaningful specialization.

At $\lambda = 0.5$ (optimal for 2 experts), k-NN reaches 75.5% (+0.1% over the default $\lambda = 5.0$). Critically, optimal entropy weight is *expert-count dependent*: while $\lambda = 0.5$ helps 2 experts (+0.1%), it degrades 64-expert performance (−0.2% vs. $\lambda = 5.0$). With 64 experts, inherent diversity from more expert parameters provides natural dispatch variation, and lower regularization leads to instability rather than improved contrast.

Loss block selection. We apply dispatch-weight loss weighting at the last MoE block (Block 11). Ablations over blocks {5, 7, 9, 11} show Block 11 performs best (75.4% k-NN), with a narrow 0.2% spread across blocks, confirming that the last block captures the most task-relevant routing patterns. Multi-block fusion (combining weights from blocks 7+9+11) underperforms single-block selection, suggesting that intermediate blocks add noise rather than complementary signal.

CLS token specialization. At Block 11 (the loss block), Expert 1 concentrates 91–95% of its dispatch weight on the CLS token, effectively becoming a CLS specialist. This concentration emerges exclusively at the loss-coupled block; non-loss blocks show no such pattern. The phenomenon provides direct evidence that loss-coupling drives expert specialization: the router learns to route the CLS token heavily to the expert weighting CLS-aligned loss.

4.4 Downstream Transfer Recipes

While MoE loss weighting improves pretraining representations, transferring MoE models to downstream tasks is non-trivial: standard finetuning can overwrite pretrained routing patterns, and extra expert parameters increase overfitting risk. We evaluate three complementary strategies: **Freeze Routing (FR)** freezes router parameters ($\mathbf{\Phi}$, s) to preserve pretrained patch-expert assignments [38]; **Freeze Attention (FA)** additionally freezes attention weights [38]; and **Expert Dropout (ExD)** applies dropout (p=0.4) to expert outputs [13]. Full rationale is in supplementary Section J.

Table 4 presents our recipe campaign on the Token+CLS 64-expert model, building each step on the previous best. Without any adaptation recipe, standard finetuning with unfrozen routing achieves 83.8%, as the router overwrites pretrained routing patterns with task-specific shortcuts. Freeze Routing (FR) improves to 84.2% (+0.4%) by preserving pretrained patch-expert assignments. Adding Freeze Attention (FA) yields 84.3%, a modest further gain. Adding Expert Dropout to FR provides the largest single gain (+0.7%), yielding FR+ExD at 84.9% by preventing over-reliance on dominant experts. The full **FR+FA+ExD** recipe reaches **85.3%**, a +1.5% improvement over vanilla MoE finetuning (83.8%), exceeding the non-MoE baseline (84.8%) by +0.5%.

The recipe transfers across objective combinations and expert counts: the Token+Pixel 32-expert model improves from 83.6% to 84.8% with FR+FA+ExD (+1.2%), exceeding the Token+Pixel non-MoE baseline (84.6%). The 2-expert Token+CLS model with FR alone achieves 84.1%, below the non-MoE baseline (84.8%), indicating that the full recipe stack is needed to overcome the additional overfitting risk from expert parameters even at the 2-expert scale.

Table 4: Downstream adaptation recipes. FR=Freeze Routing, FA=Freeze Attention, ExD=Expert Dropout ($p = 0.4$). Top: Token+CLS 64-expert recipe ablation. Bottom: cross-configuration transfer.

Configuration	FR	FA	ExD	Top-1
Token+CLS 64-Expert				
Vanilla (unfrozen routing)				83.8
FR only	✓			84.2
FR+FA	✓	✓		84.3
FR+ExD	✓		✓	84.9
FR+FA+ExD	✓	✓	✓	**85.3**
Cross-Configuration Transfer				
Token+CLS (no MoE)	—	—	—	84.8
Token+CLS 2-expert (FR)	✓			84.1
Token+Pixel 32-expert (Vanilla)				83.6
Token+Pixel 32-expert (FR+FA+ExD)	✓	✓	✓	84.8
Token+Pixel (no MoE)	—	—	—	84.6

Table 5: Semantic segmentation on ADE20K (UperNet, 160K iters, 32-expert). T+P=Token+Pixel, T+C=Token+CLS. Def=default finetuning (no recipe), W=dispatch-weight loss weighting, **All**=FR+FA+ExD.

T+P	No MoE	Def	Def+W	FR+W	**All+W**
mIoU	52.5	49.6	50.1	50.8	**52.8**

T+C	No MoE	Def	Def+W	FR+W	**All+W**
mIoU	50.8	48.3	48.9	50.3	**51.1**

Semantic Segmentation Table 5 evaluates semantic segmentation on ADE20K for both objective combinations at 32 experts, enabling controlled comparison at a common scale (the optimal expert count for Token+Pixel, Table 3). MoE models present a well-known challenge for downstream transfer [13, 38]: routing patterns optimized during pretraining can conflict with the uniform spatial coverage required for dense prediction. Without adaptation recipes, our MoE models underperform non-MoE baselines by 2.5–2.9 mIoU, consistent with prior findings that MoE finetuning is non-trivial [38]. Even in this default setting, dispatch-weight loss weighting helps modestly: +0.5 mIoU (Token+Pixel) and +0.6 mIoU (Token+CLS) over unweighted variants.

The adaptation recipes progressively close this gap. Freeze Routing (FR) alone recovers 0.7–1.4 mIoU by preserving pretrained routing patterns. The full FR+FA+ExD recipe fully closes the gap, with MoE models reaching parity or slight improvement over non-MoE baselines: Token+Pixel reaches **52.8** mIoU (+0.3 over non-MoE 52.5), and Token+CLS reaches **51.1** (+0.3 over non-MoE 50.8). This recovery arc, from MoE hurting dense prediction to matching baselines with proper adaptation, demonstrates that the recipes are essential for realizing MoE benefits beyond classification.

Table 6: Comparison with published methods on ImageNet-1K. All use ViT-Base encoders. GFLOPs=inference cost (analytical MACs at $N = 197$ tokens, cross-validated against fvcore within 1.1%). Lin.=linear probe top-1, FT=finetuning top-1, Seg.=ADE20K mIoU. "—" = not reported. ∗ Our reproduction with sparse Token+CLS encoding; Lin./FT are ours, Seg. from [22]. † 400 epochs. ‡ YFCC-15M pretrained. § Token+Pixel 32-exp with FR+FA+ExD (Section 4.4).

Method	Teacher	Params	GFLOPs	Ep.	Lin.	FT	Seg.
MAE [15]	None	86M	17.45	1600	68.0	83.6	48.1
SdAE [7]	Self	86M	17.45	300	—	84.1	48.6
BootMAE [11]	Self	86M	17.45	800	—	83.6	—
BEiT [3]	DALL-E	86M	17.45	800	56.7	83.0	45.6
MVP [30]	CLIP-B	86M	17.45	300	75.4	84.4	52.4
MaskDistill* (repr.) [22]	CLIP-B	86M	17.45	300	76.2	84.8	**53.8**
BEiT v2 [21]	CLIP-B	86M	17.45	300	80.1	85.0	52.7
MILAN† [16]	CLIP-B	86M	17.45	400	79.9	**85.4**	52.7
CAE v2 [35]	CLIP-B	86M	17.45	300	80.5	85.3	52.9
MaskCLIP‡ [12]	CLIP-B	86M	17.45	25	73.7	83.6	50.5
ExPLoRe (ours, 2-exp)	CLIP-B	116M	**11.93**	300	79.6	84.1	—
ExPLoRe (ours, 64-exp)	CLIP-B	1.86B	**13.86**	300	**80.6**	85.3	52.8§

Comparison with Published Methods Table 6 compares ExPLoRe against published methods using ViT-Base encoders. Our 2-expert model (116M) achieves 79.6% linear probe, +3.4% over MaskDistill (76.2%). Scaling to 64 experts reaches **80.6%** linear probe, the highest among compared methods, with 85.3% finetuning accuracy matching CAE v2. Semantic segmentation with the full recipe reaches 52.8% mIoU (Token+Pixel 32-expert, Table 5), competitive with BEiT v2 (52.7%) and MILAN (52.7%). MaskDistill's 53.8% segmentation is from the original paper [22]; our controlled baselines in Table 5 provide the direct comparison.

Although the 64-expert model has 1.86B parameters, Soft-MoE with one slot per expert decouples parameters from inference cost: each MoE layer routes all tokens into only E expert slots, so the expert MLPs process E rather than N inputs and only the routing projections scale with E. Both ExPLoRe configurations therefore use *fewer* inference GFLOPs than every ViT-B/16 comparator (11.93 and 13.86 vs. 17.45; Table 6). Per-FLOP, ExPLoRe is favorably positioned: E=2 trades −32% cost for a −0.9% linear-probe gap to CAE v2, and E=64 matches CAE v2 at −21% cost (cost–quality plot in the supplementary).

Robustness and Additional Probing Protocols Seed and mask-ratio robustness. Across two additional seeds and a 50% mask ratio, the 2-expert Token+CLS model varies tightly: standard deviation 0.078 (k-NN), 0.037 (linear probe), 0.039 (Efficient Probing), with mask=50% within this band (40% is the established block-masking optimum [3,22]). Our mechanism-isolation deltas (+1.3% weighted vs. unweighted, −1.6% detach) compare configurations sharing

pretraining randomness and are not single-run artifacts; full per-seed numbers are in the supplementary.

Efficient Probing and fine-grained transfer. Under Efficient Probing (EP) [23], a lightweight attentive protocol recovering patch-level information that CLS-aggregated probes under-measure, the 2-expert model reaches **80.19%** vs. **78.77%** for the non-MoE baseline (+1.41%, ∼3× the k-NN gain on the same backbones). On fine-grained Food-101 [4], EP gives **89.55%** vs. **87.68%** (+1.87%), indicating the routing emphasis transfers beyond ImageNet.

5 Conclusion

We presented ExPLoRe, a framework that repurposes Soft-MoE dispatch weights as learned per-patch loss coefficients for multi-objective masked image modeling. Three findings emerge from our study. **(1) Loss-coupling drives specialization:** detaching this connection degrades performance by 1.6%, and MoE without loss-coupling underperforms the non-MoE baseline (74.1% vs. 75.7%), confirming the mechanism is essential. While absolute k-NN gains are modest (+0.5% for Token+CLS, +2.2% for Token+Pixel), the +4.4% linear probe improvement suggests loss-coupled routing reshapes the feature space in ways not fully captured by nearest-neighbor evaluation. **(2) Routing dynamics exhibit sharp transitions:** entropy regularization governs a sharp transition between stable routing and catastrophic collapse, with the optimal weight being expert-count dependent ($\lambda = 0.5$ for 2 experts, $\lambda = 5.0$ for 64). **(3) MoE transfer requires dedicated recipes:** our FR+FA+ExD recipe improves MoE fine-tuning by +1.5% over vanilla MoE (83.8% → 85.3%), exceeding the non-MoE baseline (84.8%). Without recipes, MoE models underperform on segmentation by 2.5–2.9 mIoU; the full recipe closes this gap (+0.3 mIoU for both objective combinations).

Limitations and Future Work. Our experiments use ViT-Base with up to 64 experts (GPU-memory bounded) and a frozen CLIP teacher, though the mechanism is teacher-agnostic and applies to any multi-objective MIM framework. Natural extensions include a *dedicated loss-weighting router* separate from the feature router (enabling top-k routing across all experts per loss), *multiple slots per expert* ($S > 1$), *learned block aggregation* of dispatch weights across MoE layers, and scaling to larger models and longer schedules.

References

1. Baevski, A., Babu, A., Hsu, W.N., Auli, M.: data2vec 2.0: Highly efficient self-supervised learning for vision, speech and language. In: International Conference on Machine Learning. pp. 1694–1714 (2023)
2. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: data2vec: A general framework for self-supervised learning in speech, vision and language. In: International Conference on Machine Learning. pp. 1298–1312 (2022)
3. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: International Conference on Learning Representations (2022)

4. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: European Conference on Computer Vision (ECCV) (2014)
5. Chen, T., Chen, X., Du, X., Rashwan, A., Yang, F., Chen, H., Wang, Z., Li, Y.: Adamv-moe: Adaptive multi-task vision mixture-of-experts. In: IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17346–17357 (2023)
6. Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., Wang, J.: Context autoencoder for self-supervised representation learning. International Journal of Computer Vision **132**(1), 208–223 (2024)
7. Chen, Y., Liu, Y., Jiang, D., Zhang, X., Dai, W., Xiong, H., Tian, Q.: Sdae: Self-distillated masked autoencoder. In: European Conference on Computer Vision. pp. 108–124 (2022)
8. Chen, Z., Badrinarayanan, V., Lee, C.Y., Rabinovich, A.: Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In: International Conference on Machine Learning. pp. 794–803 (2018)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. pp. 4171–4186 (2019)
11. Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., Yu, N.: Bootstrapped masked autoencoders for vision bert pretraining. In: European Conference on Computer Vision. pp. 247–264 (2022)
12. Dong, X., Bao, J., Zheng, Y., Zhang, T., Chen, D., Yang, H., Zeng, M., Zhang, W., Yuan, L., Chen, D., et al.: Maskclip: Masked self-distillation advances contrastive language-image pretraining. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10995–11005 (2023)
13. Fedus, W., Zoph, B., Shazeer, N.: Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research **23**(120), 1–39 (2022)
14. Han, X., Wei, L., Dou, Z., Wang, Z., Qiang, C., He, X., Sun, Y., Han, Z., Tian, Q.: Vimoe: An empirical study of designing vision mixture-of-experts. arXiv preprint arXiv:2410.15732 (2024)
15. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16000–16009 (2022)
16. Hou, Z., Sun, F., Chen, Y.K., Xie, Y., Kung, S.Y.: Milan: Masked image pretraining on language assisted representation. arXiv preprint arXiv:2208.06049 (2022)
17. Jiang, Z., Zheng, G., Cheng, Y., Awadallah, A.H., Wang, Z.: Cr-moe: Consistent routed mixture-of-experts for scaling contrastive learning. Transactions on Machine Learning Research (2024)
18. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7482–7491 (2018)
19. Lin, B., Ye, F., Zhang, Y., Tsang, I.W.: Reasonable effectiveness of random weighting: A litmus test for multi-task learning. Transactions on Machine Learning Research (2022), arXiv:2111.10603

20. Liu, T., Blondel, M., Riquelme, C., Puigcerver, J.: Routers in vision mixture of experts: An empirical study. Transactions on Machine Learning Research (TMLR) (2024)
21. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: Beit v2: Masked image modeling with vector-quantized visual tokenizers. arXiv preprint arXiv:2208.06366 (2022)
22. Peng, Z., Dong, L., Bao, H., Ye, Q., Wei, F.: A unified view of masked image modeling. arXiv preprint arXiv:2210.10615 (2022)
23. Psomas, B., Christopoulos, D., Baltzi, E., Kakogeorgiou, I., Aravanis, T., Komodakis, N., Karantzalos, K., Avrithis, Y., Tolias, G.: Attention, please! revisiting attentive probing through the lens of efficiency. In: International Conference on Learning Representations (ICLR) (2026)
24. Puigcerver, J., Riquelme, C., Mustafa, B., Houlsby, N.: From sparse to soft mixtures of experts. In: International Conference on Learning Representations (ICLR) (2024)
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763 (2021)
26. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Susano Pinto, A., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. In: Advances in Neural Information Processing Systems (NeurIPS) (2021)
27. Sener, O., Koltun, V.: Multi-task learning as multi-objective optimization. In: Advances in Neural Information Processing Systems. vol. 31 (2018)
28. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In: International Conference on Learning Representations (ICLR) (2017)
29. Wei, C., Fan, H., Xie, S., Wu, C.Y., Yuille, A., Feichtenhofer, C.: Masked feature prediction for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14668–14678 (2022)
30. Wei, L., Xie, L., Zhou, W., Li, H., Tian, Q.: Mvp: Multimodality-guided visual pre-training. arXiv preprint arXiv:2203.05175 (2022)
31. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision. pp. 418–434 (2018)
32. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9653–9663 (2022)
33. Yang, X., Lu, J., Qiu, H., Li, S., Li, H.: Astrea: A moe-based visual understanding model with progressive alignment. arXiv preprint arXiv:2503.09445 (2025)
34. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: Advances in Neural Information Processing Systems. vol. 33, pp. 5824–5836 (2020)
35. Zhang, X., Yuan, J., Wei, X., Wei, Y., Hong, S., Wang, J.: Cae v2: Context autoencoder with clip target. arXiv preprint arXiv:2211.09799 (2024)
36. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barber, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 633–641 (2017)
37. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. In: International Conference on Learning Representations (2022)

38. Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., Fedus, W.: St-moe: Designing stable and transferable sparse expert models. arXiv preprint arXiv:2202.08906 (2022)