L-MoL: Mitigating Multilingual Interference via a Language-Guided Mixture of Low-Rank Experts

Anonymous authors
Paper under double-blind review

Abstract

Can a trained dense multilingual translation model benefit from sparse adaptation without acquiring new training data? We propose L-MoL, a language-guided mixture of low-rank residual experts attached to decoder feed-forward networks. The adaptation protocol freezes the dense backbone and trains the expert branches and router. Concatenating a target-language embedding with each token representation gives the router a language-specific logit bias while retaining token-dependent selection. On the 188 available OPUS-100 test directions, L-MoL improves its dense seeds by 2.69 BLEU (Base) and 1.54 BLEU (Large), with additional parameters equivalent to 3.8% and 8.3% of the respective seed sizes. At Base scale, it exceeds the strongest tested language-agnostic routing adaptation by 1.84 BLEU and its own hidden-state-only ablation by 1.95 BLEU. On five selected language pairs, mean routing-overlap reductions are approximately 81% for the three distant pairs and 7.5% for the two related pairs. These results support explicit target-language conditioning as a useful inductive bias for multilingual expert allocation; routing overlap is a diagnostic, not a direct measurement of gradient conflict.

1 Introduction

Multilingual neural machine translation (MNMT) aims to support many translation directions within a single model. Large systems such as M2M-100 (Fan et al., 2021) and NLLB (Team, 2024) show strong cross-lingual transfer, especially for low-resource languages, but fixed-capacity multilingual models often suffer from the "curse of multilinguality" Conneau et al. (2020): languages compete for limited representation capacity and performance degrades.

Sparse Mixture-of-Experts (MoE) architectures Shazeer et al. (2017); Lepikhin et al. (2021); Fedus et al. (2022) address this bottleneck by activating different expert subsets, but face two critical challenges: (1) *Prohibitive Training Costs*, since high-performance MoE models are typically trained from scratch on massive data; and (2) *Routing Inefficiency*, since conventional routers are language-agnostic and must infer language identity from token representations alone, which can cause "representation collapse" Chi et al. (2022) or conflicting expert allocation across language families Zhao et al. (2024).

As a substitute for training from scratch, Parameter-Efficient Fine-Tuning (PEFT) methods, particularly Low-Rank Adaptation (LoRA) Hu et al. (2022), adapt models with far fewer trainable parameters. Recent MoE+LoRA and LoRA-composition methods further show that multiple low-rank modules can be routed, merged, or composed efficiently Wu et al. (2024); Li et al. (2024; 2025). However, these methods are mostly designed as general LLM adaptation or adapter-composition frameworks. They do not directly study frozen-checkpoint sparse adaptation for MNMT, nor do they analyze how expert routing interacts with cross-lingual interference across different translation directions.

To answer this, we propose **L-MoL**, a **Language-Guided Mixture of Low-rank Experts** for parameter-efficient sparse adaptation of multilingual NMT checkpoints. L-MoL is inspired by *Sparse Upcycling* Komatsuzaki et al. (2023) in that it starts from a trained dense checkpoint, but differs from the original weight-copying formulation: it freezes the dense FFN backbone and adds zero-initialized routed low-rank residual experts in the decoder FFN branch. A language-aware router combines

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

decoder hidden states with trainable target-language embeddings to organize expert sharing and specialization across translation directions.

Our contribution is a practical dense-to-sparse specialization recipe for multilingual NMT under a strict parameter-efficient constraint. Rather than introducing a new MoE primitive, L-MoL studies how frozen dense translation knowledge, routed low-rank residual experts, and language-conditioned routing can be combined effectively.

- **Parameter-efficient sparse adaptation:** L-MoL improves Dense-Base by 2.69 BLEU and Dense-Large by 1.54 BLEU while reusing the original corpus. Added parameters are 3.8% and 8.3% of the respective dense seeds; adaptation requires additional optimization steps.
- **Controlled routing comparisons:** With a common dense seed and low-rank expert pool, L-MoL exceeds the strongest tested adapted routing baseline by 1.84 BLEU at Base scale. Removing its language embedding reduces BLEU by 1.95.
- **Routing diagnostics:** A heatmap and pairwise overlap measurements show language-dependent expert usage, with greater separation on the selected distant pairs than on the related pairs. No language-family labels are used in training.

2 Related Work

2.1 Sparse Mixture-of-Experts in Multilingual Machine Translation

Sparse MoE models such as GShard Lepikhin et al. (2021), Switch Transformer Fedus et al. (2022) and NLLB Team (2024) scale multilingual capacity through conditional computation. Sparse Upcycling Komatsuzaki et al. (2023) reduces the cost by initializing MoE experts from a dense model but still fine-tunes the full model; SCoMoE Zeng & Xiong (2023) improves sparse training efficiency through structured communication; and Lingual-SMoE Zhao et al. (2024) introduces language-level routing for multilingual specialization, but trains the whole model from random initialization. L-MoL differs from all of these: it freezes the dense NMT checkpoint and adds trainable low-rank residual experts as an adaptation stage, so the trainable budget stays small while the seed model's translation competence is preserved.

2.2 Parameter-Efficient and MoE-Style Adaptation

PEFT methods such as LoRA Hu et al. (2022), language-specific adapters Pfeiffer et al. (2020), and recent multilingual PEFT methods Whitehouse et al. (2024); Lee et al. (2025); Zhao et al. (2025) adapt models with a small number of trainable parameters. Related MoE-style PEFT methods, including LoRAMoE Dou et al. (2024), AdaMix Wang et al. (2022), MoLE Wu et al. (2024), MixLoRA Li et al. (2024), and LoRA-Mixer Li et al. (2025), further route, merge, or compose lightweight experts for efficient adaptation.

L-MoL focuses on dense-checkpoint adaptation for encoder–decoder multilingual translation. Its distinguishing choice is explicit target-language conditioning of token-level expert selection. Our comparisons adapt selected routing rules to this common architecture; they are not full reproductions of the original methods, whose training objectives and composition procedures can differ.

3 Methodology

In this section, we present **L-MoL**, a language-guided mixture of low-rank experts for parameter-efficient sparse adaptation. We first define the adaptation setup, then detail the L-MoL architecture and routing mechanism, and finally analyze how routing probabilities modulate cross-lingual gradient interference.

3.1 Sparse Adaptation Setup

Let $\mathcal{M}_{dense}$ be a trained multilingual Transformer with parameters θ_{dense}. Instead of converting every FFN into a full-rank MoE and fine-tuning the entire model, L-MoL freezes θ_{dense} and attaches a sparse

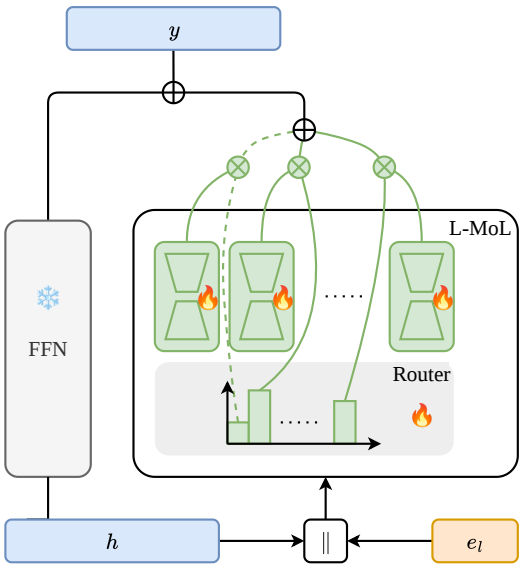

Figure 1: The architecture of L-MoL. We adopt a **Sparse Adaptation** strategy involving two parallel paths: a **Frozen Path** containing the pre-trained dense FFN (left), and a **Trainable Path** comprised of L-MoL module (right). The router combines the decoder hidden state $\mathbf{h}$ and language-aware routing embedding $\mathbf{e}_l$ to compute gating probabilities. "$\|$" denotes a concatenation operation. Outputs are fused via a residual connection ($\oplus$).

trainable branch to each decoder FFN. This setup treats the dense model as the stable translation backbone and confines new specialization capacity to the router and low-rank experts:

$$\theta_{\text{train}} = \theta_{\text{router}} \cup \{\mathbf{A}_i, \mathbf{B}_i\}_{i=1}^{N} \cup \{\mathbf{e}_l\}_{l \in \mathcal{L}} \tag{1}$$

The goal is to add sparse language-aware capacity with a small trainable parameter budget while preserving the translation behavior of the dense checkpoint.

3.2 L-MoL Architecture

As illustrated in Figure 1, L-MoL modifies the decoder FFN blocks by introducing a parallel adaptation branch. The computation flow is split into two streams:

1. **Frozen dense stream:** the original FFN preserves the behavior of the dense checkpoint.
2. **Trainable sparse stream:** routed low-rank experts provide additional language-aware adaptation capacity.

The requested target language l indexes an additional trainable routing embedding $\mathbf{e}_l \in \mathbb{R}^{d_{\text{aux}}}$. This embedding is distinct from the sequence's language-ID token embedding. Each decoder FFN router receives both $\mathbf{e}_l$ and the current token state $\mathbf{h} \in \mathbb{R}^d$. The target ID is supplied at training and inference; it is not inferred from the source script.

Formally, for a post-normalization decoder hidden state $\mathbf{h}$, the final output $\mathbf{y}$ is the residual sum of both streams:

$$\mathbf{y} = \mathbf{h} + \underbrace{\text{FFN}_{\text{fixed}}(\mathbf{h})}_{\text{Universal Backbone}} + \alpha \cdot \underbrace{\sum_{i \in \mathcal{T}} g_i(\mathbf{h}; \mathbf{e}_l) \cdot \text{Expert}_i(\mathbf{h})}_{\text{Specific Adaptation}} \tag{2}$$

where α is a scaling factor, $\mathbf{e}_l$ denotes the trainable language-aware routing embedding used by L-MoL, and $\mathcal{T}$ is the set of top-k experts selected by the router.

3.3 Low-Rank Experts as Residual Functions

To keep adaptation parameter-efficient while maintaining expressivity, each expert is implemented as a low-rank residual update with a non-linear activation. Expert i is parameterized by two low-rank matrices $\mathbf{A}_i \in \mathbb{R}^{d\times r}$ and $\mathbf{B}_i \in \mathbb{R}^{r\times d}$, where the rank $r \ll d$. The computation is defined as:

$$\text{Expert}_i(\mathbf{h}) = \phi(\mathbf{h}\mathbf{A}_i)\mathbf{B}_i \tag{3}$$

where ϕ denotes the non-linear activation function (e.g., ReLU or GELU) consistent with the dense backbone. Each expert contains $2rd$ matrix parameters. With a nonlinearity in the bottleneck, this is a LoRA-style residual adapter rather than a linear update that can necessarily be merged into a dense weight matrix.

To ensure a smooth transition from the dense baseline, we initialize $\mathbf{B}_i = 0$ while $\mathbf{A}_i \sim \mathcal{N}(0, \sigma^2)$. This ensures that at the very beginning of training (step $t = 0$), $\text{Expert}_i(\mathbf{h}) = \mathbf{0}$. Consequently, the model output $\mathbf{y}$ is mathematically identical to the pre-trained dense baseline, avoiding the "cold start" problem.

3.4 Language-Guided Routing Mechanism

Several major challenges in multilingual MoEs include **Routing Collapse** (over-allocating specific experts) and **Routing Ambiguity** (identical subwords in different languages routed to the same expert despite semantic divergence).

To resolve these, we introduce a **Language-Guided Router**. Instead of relying solely on token representations, the router explicitly conditions on both the decoder hidden state $\mathbf{h} \in \mathbb{R}^d$ and a trainable language embedding $\mathbf{e}_l \in \mathbb{R}^{d_{\text{lang}}}$. The routing logit s_i for expert i is derived from their concatenation:

$$s_i(\mathbf{h}, l) = [\mathbf{h}; \mathbf{e}_l] \cdot \mathbf{w}_i^T \tag{4}$$

where $\mathbf{w}_i \in \mathbb{R}^{d+d_{\text{lang}}}$ is the learnable weight for expert i. To understand the elegance of this mechanism, we can mathematically deconstruct the projection weight $\mathbf{w}_i$ into a token-specific component $\mathbf{w}_{i,h} \in \mathbb{R}^d$ and a language-specific component $\mathbf{w}_{i,l} \in \mathbb{R}^{d_{\text{lang}}}$:

$$s_i(\mathbf{h}, l) = \underbrace{\mathbf{h} \cdot \mathbf{w}_{i,h}^T}_{\text{Token Dynamics}} + \underbrace{\mathbf{e}_l \cdot \mathbf{w}_{i,l}^T}_{\text{Language Scalar Bias } (b_{i,l})} \tag{5}$$

The language term shifts each expert's preference for a target language, while the token term permits different choices within the same sentence. We select $\mathcal{T} = \text{TopK}(s)$ and set $g_i = \exp(s_i)/\sum_{j\in\mathcal{T}} \exp(s_j)$ for $i \in \mathcal{T}$, with $g_i = 0$ otherwise. This parameterization permits language-dependent allocation but does not guarantee expert separation or establish superiority over language-only routing.

3.5 Routing-Weighted Gradient Interaction

Conflicting task gradients are one possible source of multilingual interference (Wang et al., 2021). For a given layer and fixed incoming hidden state, an expert's local parameter gradient is multiplied by its routing weight. Disjoint expert support therefore removes direct cross-language gradient interaction on those expert parameters. With overlapping support, the interaction can be beneficial or harmful depending on gradient alignment. Appendix C states this local argument and its assumptions. It does not imply that training must learn disjoint support, or that shared router parameters cease to interact. The overlap measurements in Section 5.5 characterize allocation, not gradient signs or magnitudes.

4 Experimental Setup

4.1 Datasets and Metrics

We use English-centric **OPUS-100** (Zhang et al., 2020): 99 non-English languages define 198 possible training directions, whereas validation/test sets are available for 94 language pairs, yielding **188 evaluated directions**. Dataset statistics and resource groups are given in Table A1.

We report SacreBLEU (Post, 2018) on detokenized outputs, using `zh` for Chinese targets and `13a` otherwise. These choices are held fixed in internal comparisons; `13a` can be insensitive for scripts without whitespace word boundaries, and absolute scores from other tokenization protocols are not directly comparable.

4.2 Baselines

We compare L-MoL against dense Transformers and reported MoE systems across Base and Large scales.

Dense Models (Seed Backbones). We train two dense Transformers from scratch on OPUS-100 to serve as strong baselines and as initialization checkpoints for our adaptation: **Dense-Base** (6 encoder/decoder layers, $d_{\text{model}} = 512$, $d_{\text{ffn}} = 2048$) and **Dense-Large** (12 layers, $d_{\text{model}} = 1024$, $d_{\text{ffn}} = 4096$), the latter verifying whether L-MoL still helps atop a high-capacity model.

Reported MoE Systems. We include **GShard** Lepikhin et al. (2021) (token-level MoE with top-2 gating), **SCoMoE** Zeng & Xiong (2023) (structured communication for efficient sparse training) and **Lingual-SMoE** Zhao et al. (2024) (language-level routing, but full-parameter training from random initialization) as external reference points for performance and parameter scale, cited from their papers and therefore not sharing our backbone.

4.3 Implementation Details

We implement all models in Fairseq Ott et al. (2019). Dense baselines are trained from scratch on OPUS-100, and L-MoL is initialized from the corresponding dense checkpoint with the backbone frozen. During adaptation, only the router, language-aware routing embeddings, and low-rank experts are updated. Detailed model configurations, training schedules, hardware, and routing mechanics are reported in Appendix D.

4.4 Ablation Setup on Language-Guidance Routing

To isolate the effect of language-guided expert selection, we train a hidden-state-only routing variant in the Base setting, denoted **w/o Lang. Guide**. This variant removes the language-aware routing embedding $\mathbf{e}_l$ from Equation (4) and computes router logits only from $\mathbf{h}$. All other factors, including the frozen dense backbone, number of experts, low-rank expert rank, top-k, optimizer, and stopping rule, are kept unchanged. This variant also serves as a PEFT-style capacity control.

5 Results and Analysis

5.1 Main Result

Table 1 compares L-MoL with dense baselines and reported MoE systems across Base and Large scales.

Superiority over Strong Baselines. Dense-Base and Dense-Large achieve 23.10 and 27.75 BLEU. Adaptation raises these scores to 25.79 and 29.29, respectively: gains of **2.69** and **1.54 BLEU**. These seed-to-adapted comparisons include the additional adaptation stage; the matched adaptation controls below are needed to assess the contribution of routing.

Contextual Comparison with Reported MoEs. L-MoL-Large's reported 29.29 BLEU is numerically 0.31 below SCoMoE-Seq-β (29.60), with approximately 0.66B versus 4.1B total parameters. This contextual comparison does not establish matched-compute superiority: the systems differ in training, preprocessing, and potentially evaluation. L-MoL reuses a dense seed, whose pretraining cost must be included when comparing end-to-end training budgets.

Consistent Gains Across All Resource Settings. L-MoL improves across resource settings: at Large scale, High (+0.86), Medium (+2.21), and Low (+2.08), suggesting that language-guided routing allocates language-specific capacity to under-represented languages while still allowing selective sharing among related languages.

Table 1: **OPUS-100 results.** †: published reference scores under different training/evaluation setups (Zeng & Xiong, 2023; Zhao et al., 2024), not controlled comparisons. Other rows use our protocol. Δ is relative to the corresponding dense seed (the source paper's seed for cited rows). Bold marks the best value in each column among the rows trained under our protocol (rows without †); published † rows, which differ in setup and parameter scale, are excluded from bolding, and bold in the Δ column marks L-MoL's gain over its seed. High/Med/Low cover 44/27/23 of the 94 evaluated languages (88/54/46 directions) and average both translation directions within the group. The Overall column is the unweighted mean over all 188 directions and therefore equals the direction-count-weighted mean of the three group columns, e.g. $23.10 = (44 \cdot 23.46 + 27 \cdot 25.02 + 23 \cdot 20.15)/94$ for Dense-base; the same identity holds for every row. α/β identify SCoMoE communication configurations. Counts denote total parameters.

Model	# Params (Billion)	Avg BLEU by Resource High	Med	Low	Overall Avg	Δ
Base Scale						
Dense (Lingual-SMoE)†	0.17 B	19.67	24.92	24.22	22.21	-
Dense (SCoMoE)†	∼0.2 B	-	-	-	23.43	-
Gshard†	∼0.2 B	-	-	-	24.41	+0.98
SCoMoE-Seq-α†	∼0.2 B	-	-	-	24.45	+1.02
SCoMoE-Feat-α†	∼0.2 B	-	-	-	23.47	+0.04
Lingual-SMoE†	∼1.77 B	25.58	31.57	27.08	27.67	+5.46
Dense-base (Ours)	0.17 B	23.46	25.02	20.15	23.10	-
L-MoL-Base w/o Lang. Guide	0.18 B	24.10	26.08	20.72	23.84	+0.74
L-MoL-Base (Ours)	**0.18 B**	26.57	27.61	22.17	25.79	**+2.69**
Large Scale						
Gshard†	∼4.1 B	-	-	-	26.78	-
SCoMoE-Seq-α†	∼4.1 B	-	-	-	27.19	-
SCoMoE-Feat-α†	∼4.1 B	-	-	-	27.99	-
SCoMoE-Seq-β†	∼4.1 B	-	-	-	29.60	-
SCoMoE-Feat-β†	∼4.1 B	-	-	-	28.19	-
Dense-large (Ours)	∼0.61 B	27.39	30.37	25.35	27.75	-
L-MoL-Large (Ours)	**∼0.66 B**	**28.25**	**32.58**	**27.43**	**29.29**	**+1.54**

The Margin Is Smaller at Large Scale. Adaptation gains $+1.54$ BLEU at Large scale against $+2.69$ at Base. The two settings differ at once in seed strength (27.75 vs. 23.10), pool size (N=32 vs. 64) and rank (r=64 vs. 16) under the same 100k-update budget; we do not attribute the difference to any single factor.

5.2 Controlled Comparison with MoPE-Style Routing Methods

We adapt routing rules from AdaMix (Wang et al., 2022), MoLE (Wu et al., 2024), MixLoRA (Li et al., 2024), and LoRA-Mixer (Li et al., 2025) to the same encoder–decoder framework. The expert pool is fixed (N=64, r=16 for Base; N=32, r=64 for Large), as are the dense seed, optimizer, and stopping rule. These are routing adaptations rather than complete reproductions; differences in routing modules may still change parameter counts or computation. The hidden-state-only version of our own router provides the closest ablation of language conditioning.

The tested language-agnostic adaptations span 23.72–23.95 BLEU at Base and 28.10–28.30 at Large. L-MoL exceeds their best scores by **1.84** and **0.99 BLEU**. This supports language conditioning in our adaptation setting; it does not establish a general performance ceiling for the original methods (implementation notes: Appendix H).

5.3 Ablation on Language-Guided Expert Selection

Controlled Removal of Language Guidance. We compare L-MoL-Base against the **w/o Lang. Guide** variant (Table 1), which removes the language-aware routing embedding so that expert selection depends only on decoder hidden states, with the expert pool, rank, top-k and optimization

Table 2: **Routing adaptations and controls.** Adapted methods share the seed and expert pool, not necessarily identical routing-module cost. C has one expert; D uses the full pool with fixed random routing (Base only). L-MoL has approximately 0.18B/0.66B *total* parameters and 6.5M/50.8M added parameters at Base/Large scale. Parentheses show gains over the respective dense seed.

Model	Routing Mechanism	Base BLEU	Large BLEU
Dense Baseline (Seed)	–	23.10	27.75
C: single shared LoRA (N=1)	no routing	23.28 (+0.18)	–
D: 64 experts, frozen random	$\mathbf{h}^\top R$, untrained	23.34 (+0.24)	–
AdaMix (Adapted)	Stochastic adapter	23.72 (+0.62)	28.10 (+0.35)
MoLE (Adapted)	Token top-k	23.81 (+0.71)	28.15 (+0.40)
MixLoRA (Adapted)	Token softmax	23.88 (+0.78)	28.22 (+0.47)
LoRA-Mixer (Adapted)	Serial attention	23.95 (+0.85)	28.30 (+0.55)
L-MoL w/o Lang. Guide	Token top-k (proxy)	23.84 (+0.74)	28.25 (+0.50)
L-MoL (Ours)	Language-guided bias	**25.79 (+2.69)**	**29.29 (+1.54)**

setup unchanged. This is both a capacity control and a controlled proxy for language-agnostic MoE+PEFT adaptation (Section 5.2).

Routing Guidance Adds Beyond Expert Capacity. Adding the same low-rank expert pool with hidden-state-only routing improves Dense-Base from 23.10 to 23.84 BLEU, so additional low-rank capacity is useful on its own. To ask how much of the remaining gap is *routing* rather than capacity, we train two further arms at the main scale, both upcycled from the same dense seed and trained under identical data, seed, rank (r=16) and schedule: arm C adds a single shared low-rank expert (top-1), and arm D adds the full 64-expert pool behind a *frozen random* router (logits $= \mathbf{h}^\top R$, with R drawn once and never updated). Table 2 places them alongside the routing variants of Section 5.2 and gives the resulting ladder.

The observed increments are 0.06 BLEU from C to D, 0.50 from D to hidden-state-only routing, and 1.95 from hidden-state-only to language-guided routing. The last contrast is the largest, but these increments are not a causal decomposition of independent effects. In particular, a fixed random projection can still partition token states nonuniformly; its small gain does not imply that all experts receive the same tokens. Gains over hidden-state-only routing are 2.47/1.53/1.45 BLEU in High/Med/Low groups.

Directional Asymmetry in Low-resource Generation. In X-En translation, w/o Lang. Guide achieves 27.31 / 29.61 / 26.58 BLEU (High / Med / Low), but in the En-X direction the scores drop to 20.89 / 22.56 / 14.86. The asymmetry is most severe for low-resource *target* generation, where hidden states alone are unreliable for selecting target-language experts; L-MoL recovers En-X performance to **24.61** and **17.07** BLEU for Med and Low groups.

Off-Target Reading of the Asymmetry. Explicit target conditioning could help with off-target generation (Johnson et al., 2017; Chen et al., 2023; Yang et al., 2021), but we do not measure off-target rates. The directional imbalance is already visible at the main scale, without any auxiliary run: for the hidden-state-only router, Med-resource BLEU is 29.61 on X→En but 22.56 on En→X—a 7.05 BLEU directional gap—and language guidance lifts the En→X side to 24.61. A separate, shorter-budget replication with an unfrozen backbone shows the same ordering (gain over a shared expert: 0.446 BLEU on En→X against 0.174 on X→En; Appendix G); its absolute level lies far below the main scale and its contrasts are *not* estimates of the main-scale effects. Neither observation establishes the mechanism, and the significance of each directional gain on its own is not a test of the difference between them.

5.4 Consistency Across Scales

Directions involving Telugu, Kannada, and Kinyarwanda appear among the five largest improvements at both scales, while yi–en regresses at both (Appendix B). The causes are not established: decoder states contain source information through cross-attention, and tokenizer sensitivity alone does not explain a regression. Character-based evaluation would help assess robustness on unsegmented scripts.

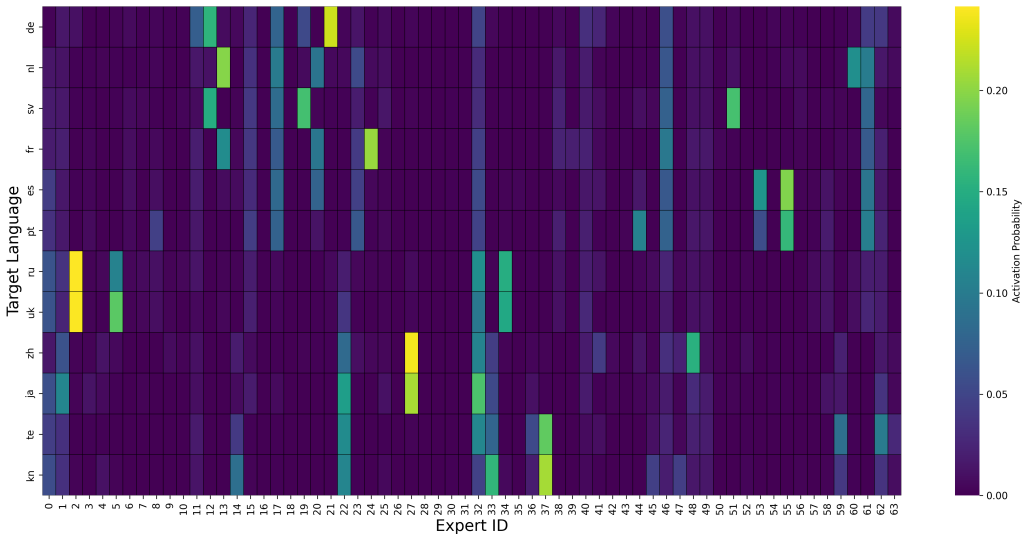

Figure 2: **Expert Specialization Heatmap** (L-MoL, Layer 5). Rows: target languages; columns: experts; color: activation probability. Specialized expert groups emerge by language family and script without phylogenetic supervision (cluster details in text).

Table 3: **Routing overlap on five selected pairs.** Lower Ω means less similar mean gate distributions, not necessarily fewer shared experts. Red.: relative reduction. Control: average of the nine per-language split-half means (Table A5), a sampling-stability diagnostic rather than uncertainty in model differences.

Type	Pair	w/o Guide	L-MoL	Red.
Distant	zh–en	0.482	**0.094**	**80.5%**
Distant	te–fr	0.435	**0.076**	**82.5%**
Distant	ja–de	0.467	**0.088**	**81.2%**
Related	es–pt	0.584	0.542	7.2%
Related	de–nl	0.562	0.518	7.8%
Control	split-half means	0.953	0.978	–

5.5 Language-Dependent Expert Usage

We visualize the Layer-5 routing probability in Figure 2.

Emergent Clustering. The router groups languages by family without any family supervision: Germanic (de, nl, sv), Slavic (ru, uk), and Iberian-Romance (es, pt) clusters each co-activate dedicated experts.

Within-family sharing. Below family level, es/pt share Experts #53 and #55, distinct from fr, while the Dravidian pair (te, kn) stays separate from European languages.

Quantifying Routing Overlap. The heatmap is qualitative; we also measure separation directly. Let $\bar{g}_k(l)$ be the expected activation probability of expert k on target language l, and define the routing-overlap index as the cosine similarity of expected activation vectors,

$$\Omega_{AB} = \frac{\sum_{k=1}^{N} \bar{g}_k(A)\,\bar{g}_k(B)}{\left(\sum_k \bar{g}_k(A)^2\right)^{1/2}\left(\sum_k \bar{g}_k(B)^2\right)^{1/2}} \in [0,1], \tag{6}$$

This statistic describes mean gate distributions, not task-gradient signs or magnitudes. Identical normalized distributions give $\Omega_{AB} = 1$; disjoint support gives $\Omega_{AB} = 0$.

Selected-pair pattern. Mean relative reductions are approximately 81% on the three distant pairs and 7.5% on the two related pairs. Mean related-minus-distant overlap increases from 0.112 to 0.444.

Table 4: **Adaptation overhead.** "Added" counts the low-rank experts, router and language embedding table; with the backbone frozen these are also the trainable parameters, as a share of the *adapted* total. "Added MACs" is the sparse branch's multiply–accumulate count relative to the frozen FFN, from tensor shapes; it is not a measured speedup.

Scale	Backbone	Added	% of backbone	% trainable	Active/token	Added MACs
Base	173M	6.53M	3.77%	3.64%	4 / 64	4.9%
Large	610M	50.77M	8.32%	7.68%	4 / 32	6.7%

These selected pairs suggest differentiated sharing, but do not establish a benchmark-wide family effect or reduced gradient conflict. Split-half consistency does not justify a reliability correction without an explicit measurement model.

5.6 Parameter and Adaptation Cost

Each expert is low-rank and only a subset is evaluated per token, so the overhead is small. At Base scale one expert costs $2dr = 1.64 \times 10^4$ parameters, the complete 64-expert pool of one decoder layer ≈ 1.05M—about half of one dense FFN ($2 \cdot d \cdot 4d \approx 2.1$M)—and the router ≈ 0.04M; over six decoder layers this is 6.53M added parameters, 3.77% of the 173M seed. At Large scale (N=32, r=64) the count is 50.77M, 8.32% of 610M.

Two consequences follow. First, adaptation updates only 3.6% (Base) or 7.7% (Large) of the adapted model, so gradient and optimizer-state memory scale with the expert branch rather than the backbone. Second, only four of the 64 (Base) or 32 (Large) experts are evaluated per token, so by tensor-shape arithmetic the sparse branch adds about 4.9% and 6.7% of the frozen FFN's multiply–accumulate operations (a full-rank MoE with the same 64 experts would multiply FFN parameters by 64). Neither figure is a measured speedup.

6 Conclusion

L-MoL attaches language-conditioned low-rank expert branches to the decoder FFNs of a dense multilingual checkpoint and trains only those branches. *First, under a fixed expert pool, language conditioning is the largest single increment we measure*: at Base scale a 64-expert pool behind an untrained random router adds 0.24 BLEU, training the router adds 0.50, and conditioning it on the target language adds 1.95. We do not claim that capacity is never binding: this holds only over the configurations tested here. *Second, the benefit concentrates where target-language generation is weakest*: the hidden-state-only router loses 7.05 BLEU from X→En to En→X on Med-resource languages, and language guidance raises the En→X side by 2.05 BLEU—a substantial but partial recovery, and the two directions are not a controlled baseline for one another. *Third, the overhead is small*: 3.8%/8.3% added parameters and four experts per token.

7 Limitations and Broader Impacts

Inherited corpus biases and translation errors require the usual caution before deployment.

- **Scope of evaluation.** We evaluate the English-centric directions of OPUS-100 with BLEU, one run per arm; non-English-centric pairs and run-to-run variation lie outside this scope.
- **Comparisons.** The adapted routing rules are re-implementations in our framework, not complete reproductions, and cited MoE scores come from different setups; L-MoL reuses a converged seed whose pretraining cost is excluded.
- **What the diagnostics establish.** The increments of Section 5.3 are nested-arm differences, not a causal partition, and overlap is measured on selected pairs; gradients and off-target rates are not measured directly.
- **Operating assumptions.** The router takes the requested target language as an ID, so unseen target languages and code-switched output are untested; Section 5.6 reports counts and arithmetic, not measured speedups.

AI USE STATEMENT

In this work, we used generative AI tools for editing and polishing the manuscript text, for checking the internal consistency of reported numbers against the result tables, for formatting LaTeX tables and figures, and for assistance in writing analysis and experiment scripts. We have not used generative AI tools to formulate the research idea or the mathematical claims, to design the method, to generate data or results, or to run experiments; these were carried out by the authors. All experiments were designed, executed, and interpreted by the authors, and every number reported in this paper was produced by our own training and evaluation pipeline. AI-assisted text and code were reviewed by the authors, who take responsibility for the final content of this work, including text, code, and claims produced with the aid of generative AI.

ETHICS STATEMENT

All experiments use the publicly available OPUS-100 benchmark. Public availability does not establish the absence of personal information or bias. The method is a training-efficiency technique for machine translation and does not introduce new data collection or human-subject involvement. We note that machine-translation systems can produce fluent but incorrect output, and we recommend human verification before deployment in high-stakes domains such as medical or legal translation.

REPRODUCIBILITY STATEMENT

Model architecture, routing mechanism, and the two-stage training procedure are described in Section 3; full hyperparameters, hardware, and training budgets are given in Appendix D, and the parameter breakdown is in Appendix E. The routing-overlap index reported in Section 5.5 is defined by an explicit formula together with the aggregation protocol. All experiments are implemented in Fairseq on the public OPUS-100 corpus. Anonymous source code is provided in the supplementary material.

REFERENCES

Liang Chen, Shuming Ma, Dongdong Zhang, Furu Wei, and Baobao Chang. On the off-target problem of zero-shot multilingual neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 9542–9558, 2023. doi: 10.18653/v1/2023.findings-acl.608.

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. *Advances in Neural Information Processing Systems*, 35:34600–34613, 2022.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 8440–8451, 2020.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, et al. LoRAMoE: Alleviating world knowledge forgetting in large language models via moe-style plugin. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1932–1945, 2024.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48, 2021.

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

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065.

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 *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=T5nUQDrM4u.

Jung Lee, Taero Kim, and Nikhil Verma. MLM: Multi-linguistic LoRA merging. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

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 *International Conference on Learning Representations*, 2021.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, et al. MixLoRA: Enhancing large language models fine-tuning with LoRA-based mixture of experts. *arXiv preprint arXiv:2404.15159*, 2024.

Wenbing Li, Zikai Song, Hang Zhou, Yunyao Zhang, Junqing Yu, and Wei Yang. LoRA-Mixer: Coordinate modular LoRA experts through serial attention routing. *arXiv preprint arXiv:2507.00029*, 2025.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, 2019.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7654–7673, 2020.

Matt Post. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, 2018.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

NLLB Team. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846, 2024.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan, and Jianfeng Gao. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5744–5760, 2022.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *International Conference on Learning Representations*, 2021.

Chenxi Whitehouse et al. Low-rank adaptation for multilingual summarization: An empirical study. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

Xun Wu, Shaohan Huang, and Furu Wei. Mixture of LoRA experts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=uWvKBCYh4S.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. URL https://aclanthology.org/2021.emnlp-main.578/.

Zhiyuan Zeng and Deyi Xiong. SCoMoE: Efficient mixtures of experts with structured communication. In *The Eleventh International Conference on Learning Representations*, 2023.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1628–1639, 2020.

Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. Sparse MoE with language guided routing for multilingual machine translation. In *The Twelfth International Conference on Learning Representations*, 2024.

Yiran Zhao et al. AdamergeX: Cross-lingual transfer with large language models via adaptive adapter merging. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025.

A Data Detail

Table A1: **Statistics of the OPUS-100 dataset.** The dataset is English-centric, containing **99 non-English languages** which, combined with English, total **100 languages**; five low-resource languages have no validation/test split and are therefore never evaluated. The *# Pairs (Eval)* column gives the grouping actually used in Table 1: 44 High, 27 Medium and 23 Low languages (88/54/46 directions), summing to 94 languages and 188 directions. Because the Overall column there is an unweighted mean over directions, it equals the direction-count-weighted mean of the three group columns.

Category	# Langs (all)	# Pairs (Eval)	Eval Size/Dir
High	45	44 (88 directions)	2000
Medium	28	27 (54 directions)	2000
Low	26	23 (46 directions)	2000
Total (non-English)	**99**	**94 (188 directions)**	**Up to 2000**

B Direction-Level Gains and Regressions

Table A2 lists the directions with the largest positive and negative transfer relative to the corresponding dense baseline, at both scales. **Bold** marks directions that appear at both scales.

Table A2: **Fine-grained Analysis: Top-5 Gains and Regressions across Scales.** We compare the transfer performance relative to the respective dense baselines for both Base and Large models. **Bold** indicates directions appearing in both scales. Δ denotes the BLEU improvement over dense baseline.

Type	L-MoL Base Scale		L-MoL Large Scale	
	Direction	Δ BLEU	Direction	Δ BLEU
Top-5 Gains	**en-rw (Kinyarwanda)**	+9.3	**te-en (Telugu)**	+6.6
	te-en (Telugu)	+7.6	en-cy (Welsh)	+4.7
	en-ps (Pashto)	+6.7	or-en (Oriya)	+4.6
	en-oc (Occitan)	+6.4	**kn-en (Kannada)**	+4.5
	kn-en (Kannada)	+6.2	**en-rw (Kinyarwanda)**	+4.4
Bottom-5 Regression	en-my (Burmese)	+0.1	en-km (Khmer)	-0.4
	yi-en (Yiddish)	-0.1	ky-en (Kyrgyz)	-0.4
	se-en (Northern Sami)	-0.5	et-en (Estonian)	-0.6
	oc-en (Occitan)	-0.6	li-en (Limburgish)	-0.6
	en-th (Thai)	-1.0	**yi-en (Yiddish)**	-1.0

C Local Analysis of Routing-Weighted Gradient Interaction

We analyze direct gradients on the expert parameters of a single decoder layer, conditional on its incoming hidden state. This describes an interaction permitted by routing; it is not a proof that optimization learns separated experts or that the measured overlap determines gradient conflict.

C.1 Problem Setup: Multilingual Gradient Interference

Consider a multilingual model training on a set of languages $\mathcal{S}$. Let θ be the trainable parameters. The total gradient update $\Delta\theta$ is an aggregate of gradients from different languages. For any two languages $A, B \in \mathcal{S}$, **gradient interference** occurs when their optimization directions conflict. Formally, the interference term $\mathcal{I}_{AB}$ is defined as the inner product of their gradients:

$$\mathcal{I}_{AB} = \langle \nabla_\theta \mathcal{L}_A, \nabla_\theta \mathcal{L}_B \rangle \tag{7}$$

For a sufficiently small gradient step on A, $\mathcal{I}_{AB} < 0$ predicts an increase in the loss on B to first order. Linguistic distance alone does not determine this sign; shared parameters can also support aligned gradients.

C.2 Gradient Isolation in L-MoL

Under the frozen-backbone protocol, updates to θ_{dense} are disabled. The experts, routers, and routing embeddings remain trainable. We analyze only the expert parameters of one layer; freezing weights does not eliminate differentiation through the dense computation.

The forward pass for expert k is $\text{Expert}_k(\mathbf{h}) = \phi(\mathbf{h}\mathbf{A}_k)\mathbf{B}_k$, where ϕ is the non-linear activation. The contribution of this expert to the final output is scaled by the routing gate $g_k(\mathbf{h}, \mathbf{e}_l)$. By the chain rule, the expected gradient received by the first low-rank matrix $\mathbf{A}_k$ from a batch of target language l is:

$$\mathbf{g}_{k,l} = \alpha\, \mathbb{E}_{\mathbf{h}\sim\mathcal{H}_l}\left[g_k(\mathbf{h}, \mathbf{e}_l)\, \mathbf{U}_{k,l}(\mathbf{h})\right], \qquad \mathbf{U}_{k,l} = \mathbf{h}^\top \left[(\mathbf{q}_l \mathbf{B}_k^\top) \odot \phi'(\mathbf{h}\mathbf{A}_k)\right]. \tag{8}$$

Here $\mathbf{q}_l = \partial\mathcal{L}_l/\partial\mathbf{y}$ is the downstream gradient at the actual forward pass; $\mathbf{U}_{k,l}$ excludes the local gate and scale factors. It is not obtained by setting the gate to one and rerunning the model. The expression assumes separate parameters for each layer and holds away from top-k boundary ties.

Now, consider the interference between Language A and Language B on this specific expert k:

$$\mathcal{I}_{AB}^{(k)} = \alpha^2 \left\langle \mathbb{E}_A[g_k \mathbf{U}_{k,A}], \mathbb{E}_B[g_k \mathbf{U}_{k,B}] \right\rangle \tag{9}$$

The factorization into mean gates and mean task gradients additionally requires negligible gate–gradient covariance. We do not assume that condition. If the local gradient factors are bounded by $\|\mathbf{U}_{k,l}\| \le M_{k,l}$, nonnegative gates instead give the conditional bound

$$|\mathcal{I}_{AB}^{(k)}| \le \alpha^2 M_{k,A} M_{k,B}\, \bar{g}_k(A)\bar{g}_k(B). \tag{10}$$

The unknown gradient bounds and gate-vector norms prevent converting cosine overlap alone into a measured amount of interference.

C.3 Routing-Based Disentanglement

For the concatenated expert parameter blocks, interaction is the sum of their inner products. This excludes the shared router and language-embedding parameters.

Our **Language-Guided Router** uses decoder hidden states together with the language-aware routing embedding $\mathbf{e}_l$ to partition the expert space.

- **Case 1: Linguistically Distant Languages (e.g., Zh vs. En).** If the router assigns disjoint sets of experts, as suggested by the final heatmap in Figure 2, then if Expert k is active for Chinese ($g_k(Zh) \approx 1$), it is deactivated for English ($g_k(En) \approx 0$). Therefore, the routing overlap term vanishes:
$$g_k(A) \cdot g_k(B) \approx 0 \implies \mathcal{I}_{AB}^{(k)} \approx 0 \tag{11}$$
Under this routing condition, the expert-level interference term is approximately zero: parameters updated by Chinese receive little or no gradient contribution from English through the same expert.
- **Case 2: Linguistically Related Languages (e.g., De vs. Nl).** The router may assign shared experts to related languages when their decoder hidden states induce similar routing patterns. Here, $g_k(De) \cdot g_k(Nl) > 0$. Positive transfer additionally requires the task-gradient term to be aligned ($\langle \mathbf{q}_{k,De}, \mathbf{q}_{k,Nl} \rangle > 0$). This condition is plausible for related languages that share syntactic or lexical regularities, and should be treated as an empirical assumption on the task gradients.

Scope. Exactly disjoint support eliminates direct interaction on the corresponding local expert blocks. Partial overlap permits both positive and negative interaction. Shared router parameters and downstream layers may still interact, so routing diagnostics cannot establish a whole-model gradient guarantee.

D Hyperparameter Settings and Implementation Details

Table A3 reports the main-scale configurations, using a single node with **8 NVIDIA H200 GPUs**. The reported effective batch size is approximately 500k tokens in both stages; adaptation uses a smaller learning rate. The independent replication uses a different budget and an unfrozen backbone (Appendix G).

Training Pipeline. We implement models using **Fairseq** Ott et al. (2019) with FP16 mixed-precision. The full pipeline consists of two stages: (1) Dense Pre-training, which takes approximately 24 hours (Base) and 48 hours (Large); (2) Sparse Adaptation, which converges in under 8 hours (Base) and 24 hours (Large). During adaptation, the dense backbone remains frozen and optimization is restricted to the low-rank experts, router, and language-aware routing embeddings.

Model Configuration. We investigate L-MoL across two scales: **Base** and **Large**, constructed by adapting the corresponding dense backbones. L-MoL-Base uses 64 low-rank experts with rank $r = 16$ and $d_{\text{aux}} = 64$. L-MoL-Large uses 32 experts with rank $r = 64$ and $d_{\text{aux}} = 64$. In both configurations, we use top-k gating with $k = 4$, set the low-rank scaling factor α to 32, and initialize the low-rank up-projection matrices $\mathbf{B}$ to zero so that the initial L-MoL state matches the pre-trained dense backbone.

Stopping Criteria. The maximum update budgets are listed in Table A3. Within those budgets, validation occurs every **2,000 steps**, with early stopping after **7 consecutive checkpoints** without improved validation loss. Equal budget caps and stopping rules need not imply equal realized update counts.

Routing Mechanics. To ensure precise expert selection, we clarify the order of operations in our gating function. For a given token, the router first computes the raw logits for all experts and selects the top-k indices. The final gating probabilities are computed by applying **Softmax only to these top-k logits**, rather than over the entire expert distribution. This prevents probability dilution and ensures that the selected experts receive significant gradient signals.

Load Balancing Strategy. Standard MoE training recipes often use auxiliary load balancing losses to prevent a small number of experts from dominating the token load. This objective is complementary to our design goal: load balancing encourages utilization, while L-MoL focuses on language-aware specialization under a frozen dense backbone. In our experiments, we set $\lambda_{\text{aux}} = 0$ and found that the model trains stably, likely because the frozen dense checkpoint provides a stable optimization landscape and the language-guided router supplies an additional signal for separating translation directions. Future work can combine L-MoL with explicit load balancing when expert utilization itself is the primary concern.

Impact of Hyperparameters. We investigate the sensitivity of L-MoL to two critical hyperparameters: the LoRA rank r and the number of experts N. Our current experiments provide only limited sensitivity evidence rather than a full hyperparameter sweep. Comparing our preliminary experiments with the final configuration, increasing the rank from 16 to 64 yielded a modest improvement in BLEU scores (+0.3 on the large model), particularly for high-resource languages that require greater capacity to model complex syntactic variations. Similarly, reducing the number of experts from 64 to 32 on the Large model while increasing the rank did not degrade performance in our setting. These observations suggest that expert capacity may be at least as important as expert count in the sparse adaptation regime, but a complete rank–expert grid is left for future diagnostic study.

E Parameter Efficiency Breakdown

We conducted a precise breakdown of the parameter overhead based on our model checkpoints. The baseline Transformer-Base model contains approximately 173M parameters, with each position-wise FFN in the decoder ($d_{\text{model}} = 512$, $d_{\text{ffn}} = 2048$) accounting for $\approx 2.1 \times 10^6$ parameters.

In L-MoL, we freeze these dense FFNs and introduce trainable components exclusively to the 6 decoder layers. For our configuration with $N = 64$ experts and rank $r = 16$, the additional parameters per layer are:

- **LoRA Experts:** The parameterized matrices $\mathbf{A}$ and $\mathbf{B}$ have shapes $[64, 512, 16]$ and $[64, 16, 512]$ respectively. The total parameter count is $2 \times 64 \times 512 \times 16 \approx 1.05 \times 10^6$.
- **Router:** The gating layer maps the concatenated hidden state and language-aware routing embedding ($512 + 64 = 576$) to expert logits, adding a negligible $64 \times 576 \approx 0.04 \times 10^6$ parameters.

For L decoder layers, N experts, hidden width d, rank r, routing width d_{aux}, and a shared table with V_{lang} entries, the added count (including router bias) is

$$P_{\text{add}} = L\left[2Ndr + N(d + d_{\text{aux}} + 1)\right] + V_{\text{lang}} d_{\text{aux}}. \tag{12}$$

With $V_{\text{lang}} = 256$ and $d_{\text{aux}} = 64$, Base adds 6,529,408 parameters (3.77% of 173M), and Large adds 50,766,208 (8.32% of 610M). Under full backbone freezing these are also the trainable counts, representing approximately 3.64% and 7.68% of the *adapted* totals. These ratios describe parameters, not measured runtime or FLOPs. Replacing one dense FFN with 64 copies gives 64 times its parameters, a 6300% increase, before routing overhead.

Table A3: Hyperparameter configurations for Dense Pre-training and L-MoL Adaptation.

	Dense Baseline		**L-MoL Adaptation**	
Hyperparameter	**Base**	**Large**	**Base**	**Large**
Encoder Layers	6	12	6	12
Decoder Layers	6	12	6	12
Hidden Dimension (d)	512	1024	512	1024
FFN Dimension ($4d$)	2048	4096	2048	4096
Attention Heads	8	16	8	16
MoE Settings	-	-		
Number of Experts (N)	-	-	64	32
LoRA Rank (r)	-	-	16	64
LoRA Alpha (α)	-	-	32	32
Top-k	-	-	4	4
Lang.-Aware Routing Emb. Dim	-	-	64	64
Auxiliary Loss Coeff.	-	-	0.0	0.0
Training				
Optimizer	Adam	Adam	Adam	Adam
Adam Betas	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)	(0.9, 0.98)
Learning Rate	$7e-4$	$7e-4$	$3e-4$	$3e-4$
LR Scheduler	inv_sqrt	inv_sqrt	inv_sqrt	inv_sqrt
Warmup Steps	4000	4000	2000	2000
Max Updates	500k	500k	100k	100k
Effective Batch Size	$\sim$500k tok	$\sim$500k tok	$\sim$500k tok	$\sim$500k tok
Dropout	0.1	0.1	0.1	0.1
Label Smoothing	0.1	0.1	0.1	0.1

Table A4: **List of all languages in OPUS-100.** We report the language name, ISO code (Abbr.), Language Family (Group), and Resource Level (Res.). Resource levels are categorized as **High** ($>$ 0.9M), **Mid** (0.1M-0.9M), and **Low** ($<$ 0.1M).

Language	Abbr.	Group	Res.	Language	Abbr.	Group	Res.
Hebrew	he	Afro-Asiatic	H	Portuguese	pt	Romance	H
Arabic	ar	Afro-Asiatic	H	Romanian	ro	Romance	H
Maltese	mt	Afro-Asiatic	M	Spanish	es	Romance	H
Hausa	ha	Afro-Asiatic	L	French	fr	Romance	H
Amharic	am	Afro-Asiatic	L	Italian	it	Romance	H
Vietnamese	vi	Austro-Asiatic	H	Catalan	ca	Romance	H
Khmer	km	Austro-Asiatic	M	Galician	gl	Romance	M
Malay	ms	Austronesian	H	Walloon	wa	Romance	M
Indonesian	id	Austronesian	H	Occitan	oc	Romance	L
Malagasy	mg	Austronesian	M	Aragonese	an	Romance	L
Mongolian	mn	Mongolic	L	Bulgarian	bg	Slavic	H
Sinhala	si	Indo-Iranian	H	Slovak	sk	Slavic	H
Malayalam	ml	Dravidian	M	Slovenian	sl	Slavic	H
Tamil	ta	Dravidian	M	Croatian	hr	Slavic	H
Telugu	te	Dravidian	L	Polish	pl	Slavic	H
Kannada	kn	Dravidian	L	Ukrainian	uk	Slavic	H
Lithuanian	lt	Baltic	H	Russian	ru	Slavic	H
Latvian	lv	Baltic	H	Bosnian	bs	Slavic	H
Irish	ga	Celtic	M	Serbian	sr	Slavic	H
Welsh	cy	Celtic	M	Czech	cs	Slavic	H
Breton	br	Celtic	M	Macedonian	mk	Slavic	H
Scottish Gaelic	gd	Celtic	L	Serbo-Croatian	sh	Slavic	M
German	de	Germanic	H	Belarusian	be	Slavic	L
Danish	da	Germanic	H	Basque	eu	Isolate	H
Dutch	nl	Germanic	H	Japanese	ja	Japonic	H
English	en	Germanic	-	Georgian	ka	Kartvelian	M
Swedish	sv	Germanic	H	Korean	ko	Koreanic	H
Icelandic	is	Germanic	H	Kinyarwanda	rw	Niger-Congo	M
Norwegian	no	Germanic	H	Xhosa	xh	Niger-Congo	M
Bokmal	nb	Germanic	M	Igbo	ig	Niger-Congo	L
Afrikaans	af	Germanic	M	Zulu	zu	Niger-Congo	L
Nynorsk	nn	Germanic	M	Yoruba	yo	Niger-Congo	L
Frisian	fy	Germanic	L	Chinese	zh	Sino-Tibetan	H
Yiddish	yi	Germanic	L	Burmese	my	Sino-Tibetan	L
Limburgish	li	Germanic	L	Thai	th	Tai-Kadai	H
Dzongkha	dz	Sino-Tibetan	L	Turkish	tr	Turkic	H
Persian	fa	Indo-Iranian	H	Azerbaijani	az	Turkic	M
Bangla	bn	Indo-Iranian	H	Uzbek	uz	Turkic	M
Assamese	as	Indo-Iranian	M	Uyghur	ug	Turkic	L
Gujarati	gu	Indo-Iranian	M	Kyrgyz	ky	Turkic	L
Tajik	tg	Indo-Iranian	M	Kazakh	kk	Turkic	L
Nepali	ne	Indo-Iranian	M	Tatar	tt	Turkic	L
Punjabi	pa	Indo-Iranian	M	Turkmen	tk	Turkic	L
Urdu	ur	Indo-Iranian	M	Hungarian	hu	Uralic	H
Hindi	hi	Indo-Iranian	M	Estonian	et	Uralic	H
Marathi	mr	Indo-Iranian	L	Finnish	fi	Uralic	H
Pashto	ps	Indo-Iranian	L	N. Sami	se	Uralic	L
Kurdish	ku	Indo-Iranian	L	Esperanto	eo	Constructed	M
Odia	or	Indo-Iranian	L	Greek	el	Hellenic	H
Armenian	hy	Armenian	L	Albanian	sq	Albanian	H

F Per-Language Split-Half Reliability

Table A5 reports the supplied per-language split-half statistics. Held-out tokens are partitioned into two disjoint 15k-token halves, and cosine similarity is computed between their mean gate vectors over five random splits shared by the models. These are sampling splits, not independently trained

model seeds. High within-language similarity indicates stable averages for this diagnostic; it does not provide uncertainty estimates for cross-language or between-model contrasts. Aggregation across languages must be distinguished from variation across splits.

Table A5: **Per-language split-half reliability** (same language, disjoint 15k-token halves, five random splits, mean $\pm$ s.d.). All values bound the cross-lingual Ω in Table 3 from above.

Language	ISO	w/o Guide	L-MoL
Chinese	zh	0.956 ± 0.004	0.981 ± 0.002
English	en	0.962 ± 0.003	0.984 ± 0.002
Telugu	te	0.938 ± 0.007	0.969 ± 0.003
French	fr	0.954 ± 0.004	0.979 ± 0.002
Japanese	ja	0.949 ± 0.005	0.975 ± 0.003
German	de	0.955 ± 0.004	0.980 ± 0.002
Spanish	es	0.958 ± 0.003	0.982 ± 0.002
Portuguese	pt	0.952 ± 0.004	0.978 ± 0.003
Dutch	nl	0.951 ± 0.004	0.976 ± 0.003
Mean of displayed language means	–	0.953	0.978

G Independent Three-Arm Replication: Setup and Within-Protocol Contrasts

This appendix documents a supplementary replication run carried out under a shorter budget and a different adaptation protocol, and reports it in full. **It is not a substitute for the main-scale experiments, and its absolute scores must not be compared with them.** Its dense seed is a separate checkpoint trained from scratch for 100k updates only, and the arms are upcycled from that seed with the backbone *unfrozen*. The main-scale protocol instead trains the dense seed to convergence and keeps the backbone frozen during adaptation. Both differences lower the absolute level: the replication seed scores 17.56 BLEU against 23.10 for the Dense-base of Table 1. We nevertheless report every absolute score (Table A6) rather than only the differences, because withholding them invites more doubt than disclosing them; the main text draws on the *within-replication* contrasts only, never on its absolute level.

Setup. A dense seed is trained from scratch for 100k updates. Three arms are then upcycled from *that same* seed via the expert-splitting initialization and trained for a further 100k updates each, with identical data, seed, effective batch (32,768 tokens/step), optimizer, and a freshly reset inverse-square-root schedule. Arm C adds a single shared low-rank expert (0.118M parameters). Arm D adds the 64-expert pool with a *frozen random* router, logits $= \mathbf{h}^\top R$, where R is drawn once at seed 1234 and never updated. Arm B adds the same pool with language-guided routing (Eq. 5). All arms are decoded and scored on the same 188 directions (94 languages $\times$ En$\leftrightarrow$X) under the same SacreBLEU configuration used elsewhere in the paper, with the target-language tag stripped before tokenization.

Run configuration. Training uses the same binarized OPUS-100 data, language dictionary and language-pair list as the main experiments. Effective batch is 32,768 tokens per update; the optimizer is Adam (β_1=0.9, β_2=0.98) with peak learning rate 3×10^{-4}, 2,000 warm-up updates and an inverse-square-root schedule that is reset at upcycling; precision is FP16 with memory-efficient FP16 and activation checkpointing. Each arm is a single run: no seed repetition and no checkpoint averaging. Checkpoints are read at 65k and 100k further updates and decoded with beam 5 (otherwise fairseq defaults). Scoring is SacreBLEU (`13a` tokenization, `zh` for Chinese) over the same 188 directions, with the target-language tag removed before tokenization. **The backbone is not frozen in any replication arm**—the single largest difference from the main-scale protocol, in which only the low-rank experts and the router receive updates.

Scope of this replication. Its shorter-trained seed and unfrozen backbone differ from the main-scale protocol. The contrasts in Table A7 compare arms within this separate setting; they do not validate the main-scale effect sizes. Holding the update count fixed controls duration but does not remove interactions between optimization and architecture.

Table A6: **Absolute BLEU of every replication arm** (SacreBLEU over 188 directions: 94 languages $\times$ En$\leftrightarrow$X; En$\rightarrow$X and X$\rightarrow$En are 94 directions each). Arm A is the shared dense seed, trained 100k updates from scratch with no adaptation; the remaining arms are upcycled from A and trained for the stated number of further updates with the backbone unfrozen. **These absolute scores are not comparable with Tables 1–2**, which use a converged seed and a frozen backbone; they are given only so that the contrasts of Table A7 can be inspected against the level they were measured at.

Arm	Further updates	All 188	En→X	X→En
A: dense seed	–	17.56	14.20	20.91
C: single shared LoRA	65k	19.13	15.13	23.12
D: 64 experts, fixed random	65k	19.26	15.39	23.14
B: language-guided routing	65k	19.39	15.60	23.18
C: single shared LoRA	100k	19.49	15.48	23.49
D: 64 experts, fixed random	100k	19.67	15.77	23.56
B: language-guided routing	100k	19.80	15.92	23.67

Within-protocol contrasts. Table A7 reports direction-level differences. Two-sided paired t-tests treat 188 directions as observations; 124, 137, and 149 differences are positive in the respective contrasts. Trimming the five largest and smallest differences changes each mean by less than 0.02 BLEU. Directions share languages and training data, so independence is an approximation: these nominal intervals and p-values do not quantify variation over training runs.

Table A7: **Within-protocol contrasts in the shorter-budget replication.** C: one shared expert; D: 64 experts with fixed random routing; B: language-guided routing. The seed and update budget are shared, with the backbone unfrozen. Tests summarize direction-level differences; they do not test the difference between the two directional gains.

Contrast	Mean	95% CI	p	Cohen's d
B$-$D: language-guided vs. fixed random routing	$+0.131$	$[+0.067, +0.194]$	5.3×10^{-5}	0.295
D$-$C: 64-expert pool vs. single shared LoRA	$+0.179$	$[+0.095, +0.264]$	3.3×10^{-5}	0.303
B$-$C: full module vs. single shared LoRA	$+0.310$	$[+0.220, +0.400]$	1.4×10^{-11}	0.493
Directional decomposition of B$-$C (94 directions each)				
X→En	$+0.174$	$[+0.050, +0.297]$	5.8×10^{-3}	–
En→X	$+0.446$	$[+0.321, +0.572]$	2.9×10^{-12}	–

The off-target literature includes zero-shot settings, whereas these 188 directions are supervised English-centric pairs. Their BLEU asymmetry does not establish an off-target rate or a target-specification mechanism.

We do not report measured off-target rates. In a diagnostic check, fastText `lid.176` mislabeled two thirds of en–ms reference translations as Indonesian and confused other closely related languages. Systematic classifier errors need not cancel between model outputs; such comparisons require a separately validated detector.

What the contrasts identify. All arms receive the same number of additional updates under the same protocol, avoiding unequal training duration in their comparisons. However, C$-$A ($+1.930$ BLEU) combines further backbone training *and* the shared adapter; it is not a continued-dense-only control. Although B$-$A equals $2.240 = 1.930 + 0.310$ arithmetically, these numbers do not identify an 86% training / 14% module causal attribution. B$-$C estimates the difference between the two adaptation recipes in this replication, not the isolated effect of every component or an effect transferable to the main-scale setting.

Relation to the main-scale ladder. Table 2 gives the main-scale comparisons. B$-$C is 0.310 BLEU here versus 2.51 at the main scale; D$-$C is 0.179 versus 0.060. Both freezing and training budget differ, so these differences do not identify the effect of freezing alone. This appendix additionally reports the directional split within its own protocol.

At 65k additional updates, the reported identity is $1.836 = 1.573 + 0.263$. B−D changes from 0.127 to 0.131 by 100k updates (0.004 at displayed precision), while D−C changes from 0.136 to 0.179 (0.043). These checkpoint comparisons are descriptive; they do not establish how longer training would change the contrasts or why they differ from the main-scale results.

H Implementation Notes for the Adapted Routing Baselines

Table 2 compares L-MoL with four routing rules taken from AdaMix (Wang et al., 2022), MoLE (Wu et al., 2024), MixLoRA (Li et al., 2024) and LoRA-Mixer (Li et al., 2025). All four are implemented inside our own code base on top of the same L-MoL module, so that the comparison isolates the rule that produces the gate.

Shared settings. Every adapted variant keeps the dense seed, the frozen backbone, the expert pool (N=64, r=16 at Base; N=32, r=64 at Large), top-k=4, the residual scaling α=32, the optimizer, the learning-rate schedule and the 100k-update budget recorded in Appendix D. Decoding and scoring are identical to every other row of Table 2: beam 5, SacreBLEU with `13a` tokenization (`zh` for Chinese), over the same 188 directions, with the target-language tag removed before scoring.

What differs. Only the module that maps a token state to gate weights is replaced. The residual fusion is identical in every row, $\mathbf{y} = \mathbf{h} + \mathrm{FFN}_{\mathrm{fixed}}(\mathbf{h}) + \alpha \sum_{i \in \mathcal{T}} g_i \, \mathrm{Expert}_i(\mathbf{h})$, with the same renormalisation over the selected experts described in Section 3. Table A8 records the rule used in each variant.

Table A8: **Routing rule implemented in each adapted variant.** "Gate input" is what the rule conditions on; "selection" is how the active experts are chosen; the last column records whether the rule contains trainable gate parameters. Everything else (pool size, rank, top-k, fusion, optimizer, budget) is shared with L-MoL.

Variant	Gate input	Selection	Learned gate
AdaMix (Adapted)	none	one adapter sampled at random per forward pass	no
MoLE (Adapted)	$\mathbf{h}$	top-k over learned expert logits	yes
MixLoRA (Adapted)	$\mathbf{h}$	softmax over all N experts	yes
LoRA-Mixer (Adapted)	$\mathbf{h}$	serial attention module, layer-by-layer	yes
L-MoL w/o Lang. Guide	$\mathbf{h}$	top-k over $[\mathbf{h}] \cdot \mathbf{w}_i$	yes
L-MoL (Ours)	$\mathbf{h} \Vert \mathbf{e}_l$	top-k over $[\mathbf{h}; \mathbf{e}_l] \cdot \mathbf{w}_i$	yes

Relation to the original methods. The four methods were proposed for decoder-only LLM backbones, and some of them are defined with additional objectives (for example load-balancing losses) or inference-time procedures (for example averaging over adapters). We port the routing rule into an encoder–decoder NMT model and attach the experts as a parallel branch at the decoder FFN; components that would change the training objective or the inference procedure beyond the routing rule itself are not reproduced. These variants are therefore routing adaptations rather than complete reproductions, as stated in Section 5.2, and the routing module itself can differ in parameter count, which we do not neutralise. Full configurations are included in the anonymous source code.