Depth Over Diversity: Same-Expert Iteration Outperforms Expert Communication in a Translation MoE

Anonymous authors
Paper under double-blind review

Abstract

Mixture-of-Experts (MoE) models achieve scalability through sparse expert routing, but experts process tokens independently. A natural hypothesis is that enabling expert communication—through learned topologies, message passing, or sequential chains—should improve performance. We investigate this hypothesis by testing ten communication approaches across seven experimental axes on WMT14 En-De translation. We find no clear evidence that any communication variant improves over standard MoE, though we note that sample sizes (up to five seeds for the main result, three for most communication variants) limit power for detecting small effects. Several variants degrade validation perplexity. We hypothesize that parallel experts lack information asymmetry: they all receive the same input, so inter-expert messages carry information that the routing layer already integrates through soft-ensembling. We then propose an alternative: rather than combining diverse expert outputs, we apply the *same* expert twice sequentially. This modification—**expert depth**—achieves 7.4% lower validation perplexity, an effect consistent across five seeds and growing over training. A capacity-matched ablation suggests the improvement stems from iterative refinement rather than increased capacity, as doubling expert width provides no benefit. On this benchmark, our findings suggest that the value of sparse expert routing may lie in selecting the right expert and giving it deeper processing, rather than in combining multiple expert opinions.

1 Introduction

Sparse Mixture-of-Experts (MoE) models have emerged as a powerful approach for scaling model capacity without proportional increases in compute (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022; Lepikhin et al., 2021; Jiang et al., 2024). In a typical MoE layer, a learned router selects a small subset of expert modules to process each token, and their outputs are combined via a weighted sum. Significant efforts have been directed toward improving routing mechanisms, load balancing, and expert specialization, but the experts themselves remain independent: each processes its assigned tokens in isolation, with no mechanism for inter-expert coordination.

A natural question is whether this independence is a limitation. Several recent works have explored mechanisms for **expert communication**—enabling experts to exchange information through message passing, learned topologies, or sequential composition (Wang et al., 2025). The intuition is compelling: if experts could coordinate, they might avoid redundant computation, share specialized knowledge, or produce more coherent outputs. However, this intuition has not been systematically tested. Prior work typically proposes a single communication mechanism and evaluates it in isolation, without ablating across the many design axes that could affect the outcome.

In this work, we evaluate expert communication in MoE on a translation benchmark. We test ten approaches across seven experimental axes—including communication timing (before vs. after expert processing), topology structure (learnable, fixed, random, fully-connected), expert count (8, 16, 32), training duration, domain diversity, and sequential composition—on WMT14 English-German translation with a 93M-parameter Transformer and 32 experts per layer.

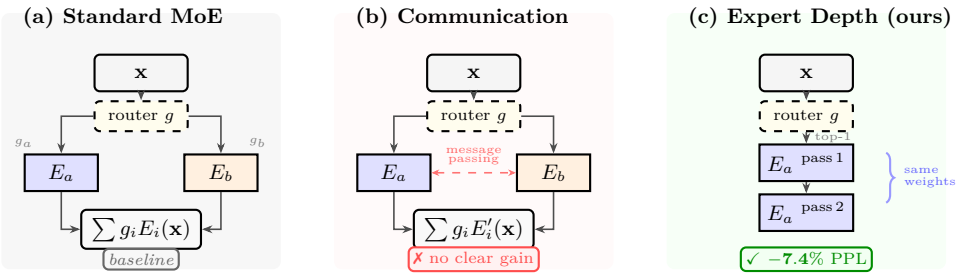

Figure 1: Three paradigms for expert computation in MoE. **(a)** Standard parallel routing. **(b)** Expert communication via message passing—10 variants tested across 1–3 seeds each; among the four configurations tested with 3+ seeds, none shows a statistically significant improvement, and six screening experiments are directionally consistent (§3). **(c)** Expert depth: the same expert processes its own output, the only approach showing a statistically significant improvement in this study (5 seeds, p=0.0025; §5).

We find no clear evidence that any communication variant improves over the baseline: none of the four tested with 3+ seeds reaches significance ($p > 0.6$), and six screening experiments (1–2 seeds) are consistent with this pattern. We note that these sample sizes provide limited power to detect small effects ($< 1\%$), so we frame these as absence of clear evidence rather than evidence of absence. Several communication variants increase validation perplexity. Notably, models *learn to use* the communication mechanisms—gates open from their suppressive initialization, learned topologies grow in density, and gradients flow through message-passing parameters—yet validation perplexity does not improve. We hypothesize that a structural property of standard MoE contributes: parallel experts operating on the same input lack **information asymmetry**, so inter-expert messages may be largely redundant with what the routing layer already integrates through its weighted output combination. Even sequential chains with different experts (Wang et al., 2025), which create genuine asymmetry by letting each expert see its predecessor's output, show no significant gain in our setting.

These negative results led us to an unexpected discovery. While investigating sequential expert chains, we found that applying the *same* expert twice—iterative refinement rather than diverse composition—produces the only statistically significant improvement in the entire study. We call this approach **expert depth**: the router selects a single expert, which processes the token and then processes its own output. Expert depth achieves 7.4% lower validation perplexity than standard MoE, an effect that is consistent across all five random seeds and grows monotonically over training, accompanied by markedly better expert load balance.

To understand whether this improvement reflects genuine iterative refinement or simply increased per-token computation, we conduct a four-way ablation (§6). A standard single expert and a width-doubled expert with twice the parameters produce identical perplexity, suggesting that added capacity alone does not explain the improvement. The observed gain appears attributable to the second pass through the same expert.

Overall, our contributions are three-fold:

- We evaluate ten approaches to expert communication in MoE across seven experimental axes, with four configurations tested at 3+ seeds; none shows statistically significant improvement (§3). We propose a structural hypothesis based on information asymmetry (§4).

- We introduce **expert depth**, a simple modification that applies the same router-selected expert twice, achieving 7.4% lower perplexity (p=0.0025, 5 seeds) and showing preliminary single-run downstream gains over a compute-matched baseline (§5).

- Through a four-way ablation, we isolate the source of improvement: added width provides no benefit, while the second pass through the same expert accounts for the observed gain (§6).

2 Background and Experimental Setup

In a standard MoE layer (Shazeer et al., 2017; Vaswani et al., 2017), each token $\mathbf{x} \in \mathbb{R}^d$ is processed by a subset of N expert modules $\{E_1, \ldots, E_N\}$ selected by a learned router g:

$$\mathbf{y} = \sum_{i \in \mathcal{S}(\mathbf{x})} g_i(\mathbf{x}) \cdot E_i(\mathbf{x}), \quad \mathcal{S}(\mathbf{x}) = \text{top-}k\big(g(\mathbf{x}), k\big) \tag{1}$$

where each expert E_i is a two-layer FFN (linear projection $\rightarrow$ activation $\rightarrow$ linear projection), $g(\mathbf{x})$ produces routing scores via a learned projection followed by softmax, and $\mathcal{S}(\mathbf{x})$ selects the top-k experts by score. The output is a weighted combination of the selected experts' outputs, providing implicit coordination through soft-ensembling.

All experiments use an identical Transformer decoder (Vaswani et al., 2017) with ∼93M total parameters, 4 layers, hidden dimension 512, and 8 attention heads. Each layer contains 32 FFN experts with hidden dimension 2048 and top-k=2 routing. We train on WMT14 English-German translation using commoncrawl and europarl as training data (4.3M sentence pairs) and the news-commentary-v9 corpus as a held-out validation set (201K pairs). This is a custom split derived from WMT14 data: news-commentary is a distinct corpus from the training sources, though it is not the standard WMT14 benchmark test set (newstest2014, which we use separately for BLEU evaluation). We use shared BPE tokenization (Sennrich et al., 2016) and train for 5 epochs using AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019) (lr=10^{-3}, weight decay 0.05) with cosine scheduling, 500-step warmup, and label smoothing of 0.05. We report validation perplexity and use paired t-tests over matched seeds with Cohen's d and 95% confidence intervals. For the main depth result, we use 5 seeds; for the capacity ablation and most communication experiments, 3 seeds; several early screening experiments use 2 seeds, which we note explicitly in Table 1.

For the depth result and capacity ablation (§6), three of the four conditions are matched in expert FLOPs per token per layer at ≈4M: the standard baseline activates k=2 experts of dimension 2048; expert depth activates 1 expert twice through a 2048-dim FFN; and the width control activates 1 expert of dimension 4096. A fourth condition—a single-expert control (k=1, dimension 2048, ≈2M FLOPs)—uses half the compute budget to isolate the specific contribution of top-k=2 ensembling.

3 Expert Communication Shows No Clear Gain

Overview. We evaluate ten approaches to expert communication across seven axes. Table 1 summarizes all results; among the four configurations evaluated with 3+ seeds, none shows a statistically significant improvement over standard independent-expert MoE, and the screening-level results are directionally consistent.

Parallel communication (rows 1–8). We implemented a learnable communication topology parameterized as $\sigma(\mathbf{W})$ where $\mathbf{W} \in \mathbb{R}^{N \times N}$ are learned logits and σ is the sigmoid function. Messages are exchanged through a learned projection and gating mechanism: expert states are projected, aggregated via topology-weighted attention, and gated before combining with the original state. We evaluated this at multiple expert counts (8, 16, 32), training durations (5, 15 epochs), communication timings (before FFN, after FFN, both), and with multi-domain data (En-De + En-Fr with top-k=1 to force expert isolation). At 32 experts, post-FFN communication increases perplexity by 1.26%; combining pre- and post-FFN yields the largest degradation at +2.41%. Multi-domain training does not induce domain specialization: expert domain purity remains near-uniform with shared BPE tokenization (see Appendix A for details), so there is no asymmetric information to exchange.

Interestingly, across all parallel communication experiments, models *actively learn to use* the communication mechanisms. Pre-FFN gate biases move from their suppressive initialization toward zero, topology density grows organically, and gradients flow through the message-passing parameters (quantitative details in Appendix A). Yet validation perplexity does not improve. This suggests that diagnostic evidence of mechanism usage—open gates, growing topology, non-zero gradient norms—is not sufficient evidence that a mechanism is beneficial, a point we believe is methodologically important for evaluating architectural innovations more broadly.

Table 1: Summary of expert communication experiments. $\Delta\%$ is relative change in validation perplexity vs. standard MoE (lower is better). Rows with 2 or 1 seed are screening-level evidence; rows with 3 seeds report p-values from paired t-tests where shown. Among the four configurations evaluated with 3 seeds, none achieves a statistically significant improvement; the screening-level rows are directionally consistent, and several variants increase perplexity. Per-seed details are in Appendix A.

#	Approach	Experts	Timing	Seeds	$\Delta\%$	Verdict
1	Post-FFN learnable topology	8	Post	2	-0.06	No improv.†
2	Post-FFN learnable (15 epochs)	8	Post	3	~ 0	Not sig. (p>0.9)
3	Post-FFN learnable (16 experts)	16	Post	2	$+0.53$	No improv.†
4	Post-FFN learnable (32 experts)	32	Post	2	-0.34	No improv.†
5	Pre-FFN communication	32	Pre	3	-0.26	Not sig. (p=0.63)
6	Post-FFN (32 experts, high spec.)	32	Post	2	$+1.26$	Worse†
7	Pre-FFN + Post-FFN combined	32	Both	2	$+2.41$	Worse†
8	Multi-domain (En-De + En-Fr)	32	Pre	1	$+0.45$	Worse†
9	Sequential chain (diff. experts)	32	Seq.	3	-0.46	Not sig. (p=0.81)
10	Topology-governed chain	32	Seq.	3	-0.56	Not sig. (p=0.84)

†Screening-level (1–2 seeds); not formally tested.

Sequential composition (rows 9–10). Motivated by Chain-of-Experts (Wang et al., 2025), we also implemented sequential expert processing where tokens flow through a chain of experts, with each expert seeing the previous expert's output. In the "free" mode (row 9), per-step routing is unconstrained; in "topology-governed" mode (row 10), a learned adjacency matrix constrains allowed expert transitions. Neither achieves significant improvement (-0.46%, p=0.81 and -0.56%, p=0.84 respectively, both over 3 seeds). The learned topology converges to near-full density (effectively unconstrained; see Appendix A), suggesting the transition structure provides no useful inductive bias in this setting.

4 Analysis: A Hypothesis on Information Asymmetry

The consistency of the negative results across the tested approaches suggests a structural explanation rather than implementation-specific failures. We hypothesize that a key contributing factor is a property of standard MoE: parallel experts operating on the same input lack *information asymmetry*.

In Eq. 1, all selected experts receive the same input $\mathbf{x}$. Expert E_i's hidden state $f_i(\mathbf{x})$ is a deterministic function of $\mathbf{x}$ that any other expert could in principle reconstruct given E_i's weights. No expert possesses information that is unavailable to others, so explicit inter-expert messages carry redundant information—they communicate functions of a shared input that the routing layer already integrates through its weighted output combination $\mathbf{y} = \sum_i g_i \cdot E_i(\mathbf{x})$. In this view, the routing layer's soft-ensembling is itself an implicit coordination mechanism, and adding explicit message-passing channels on top of it is second-order.

Sequential composition with different experts does create genuine information asymmetry: expert E_B in a chain $E_B(E_A(\mathbf{x}))$ sees a transformed representation rather than the raw input. However, this comes at a cost. The model must split its compute budget across two expert functions whose weights were independently optimized, and on our translation task, composing through an unrelated expert does not produce representations that are more useful than those obtained by the parallel baseline.

If this hypothesis is correct, communication should help primarily when experts hold genuinely different information (e.g., from different training distributions) and the communication channel provides information that the routing mechanism cannot replicate. Neither condition is met in standard MoE with shared training data and top-k routing. We emphasize that this is a plausible hypothesis consistent with our observations, not a validated mechanism—a rigorous test would require settings where expert specialization can be independently verified.

Algorithm 1 Expert Depth Forward Pass (per MoE layer)

Require: Token representations $\mathbf{X} \in \mathbb{R}^{B \times d}$, experts $\{E_1, \ldots, E_N\}$, router g, number of passes T

```
logits ← g(X)                                          ▷ Router scores: B × N
H ← X                                                  ▷ Current hidden states
for t = 1 to T do
    p ← softmax(logits)                                ▷ Routing probabilities
    idx, s ← top-1(p)                                  ▷ Expert index and score per token
    for each expert i = 1, ..., N do
        M_i ← {b : idx_b = i}                          ▷ Tokens assigned to expert i
        if M_i ≠ ∅ then
            H[M_i] ← E_i(H[M_i]) · s[M_i] / sg(s[M_i])   ▷ Straight-through
        end if
    end for
end for
return H
```

Table 2: Expert depth main result (5 seeds, 5 epochs). S_depth applies the same expert twice; all seeds favor it with 4× lower cross-seed variance.

	s42	s123	s456	s789	s1024	Mean	Std	
A_none (k=2, 1 pass)	15.22	14.36	15.51	15.04	15.28	15.08	0.44	
S_depth (k=1, 2 pass)	13.99	13.77	13.95	14.11	14.01	13.97	0.12	
Difference	−1.22	−0.59	−1.57	−0.93	−1.27	−1.12		**−7.4%**

Paired t-test: $t = -6.73$, $p = 0.0025$, 95% CI = $[-1.58, -0.66]$, Cohen's $d = -3.01$

5 Expert Depth: The Same Expert Twice

Method. While investigating sequential expert chains, we discovered that applying the **same** router-selected expert twice produces a significant improvement of 7.4% in validation perplexity. Given token $\mathbf{x}$ and router-selected expert E_a, standard MoE computes:

$$\mathbf{y}_{\text{parallel}} = g_a \cdot E_a(\mathbf{x}) + g_b \cdot E_b(\mathbf{x}) \quad \text{(two different experts, one pass each)} \tag{2}$$

Expert depth instead applies the same expert twice:

$$\mathbf{y}_{\text{depth}} = E_a\big(E_a(\mathbf{x})\big) \quad \text{(one expert, two passes)} \tag{3}$$

The router selects a single expert via top-1, which processes the token, then processes its own output. Both Eq. 2 and Eq. 3 perform two expert forward passes per token, matching FLOPs. Implementation uses straight-through score estimation—routing scores are multiplied as $s/\text{sg}(s)$ (where sg denotes stop-gradient) to maintain unit forward magnitude while enabling gradient flow through the router. Algorithm 1 gives the pseudocode.

Results. Table 2 presents the main result across five random seeds. All five seeds favor expert depth, with a highly significant paired difference ($p = 0.0025$, Cohen's $d = -3.01$).

The depth benefit grows over training. Figure 2 shows the epoch-by-epoch trajectory. Expert depth starts behind at epoch 1, reflecting the initial disadvantage of top-1 routing, but crosses over at epoch 2–3 and the gap widens monotonically through epoch 5. This crossover is noteworthy: it indicates that iterative refinement is a *learned* capability, not an automatic architectural benefit. The expert must discover how to productively process its own output, and this takes several epochs to develop. The accelerating gap suggests that further training may yield even larger improvements.

Improved stability and load balance. Beyond perplexity, expert depth exhibits qualitatively different training dynamics (Figure 2b). Cross-seed variance drops by 4× (std of 0.12 vs. 0.44), suggesting a smoother

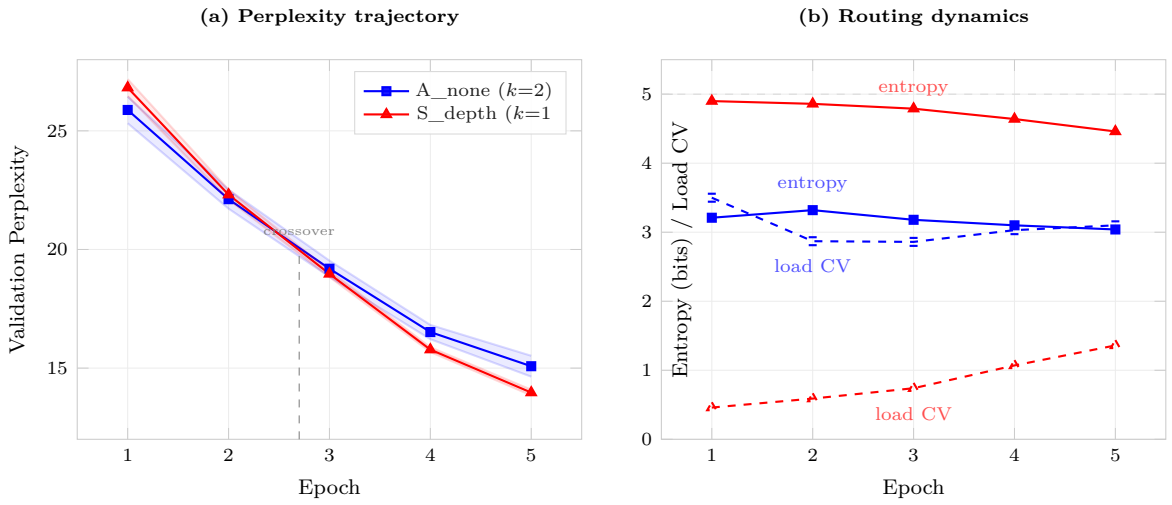

Figure 2: Expert depth training dynamics (mean ± std of 5 seeds). **(a)** Perplexity: depth starts behind due to top-1 routing but overtakes the baseline as the expert learns to refine its own output. **(b)** Routing entropy (solid, higher = more uniform) and expert load CV (dashed, lower = more balanced), layer-averaged. Depth maintains near-maximal entropy and 2–3× better load balance.

optimization landscape. Load balance also improves substantially. We observe empirically that expert depth learns more uniform routing than the baseline, with load CV falling from 3.1 to 1.4. One possible explanation is that top-k=2 creates competitive dynamics where popular experts attract disproportionate load, whereas top-1 selection reduces this effect. Routing entropy remains near-maximal throughout training, indicating far more uniform expert utilization than the baseline.

6 Capacity Ablation: Depth, Not Width

A natural objection is that expert depth improves simply because it provides more computation per token. To test this, we evaluate four conditions (3 seeds each) that decompose the contributions of routing diversity (k), expert width, and iterative depth. An implementation note: A_single and D_wide use the standard parallel routing path with top-k=1, while S_depth uses the sequential forward path (Algorithm 1) which reuses the same router logits and applies straight-through score estimation. The routing scores are applied identically (softmax + top-1 selection) across all three top-1 conditions; the difference is that S_depth passes the expert output through the same expert a second time:

Table 3: Capacity ablation conditions. Expert FLOPs are matched at ∼4M per token per layer across Standard, S_depth, and D_wide. A_single (∼2M FLOPs) isolates the top-k contribution.

Condition	Configuration	s42	s123	s456	Mean	Std	Params
A_none	k=2, dim=2048, 1 pass	15.22	14.36	15.51	15.03	0.60	64M
A_single	k=1, dim=2048, 1 pass	16.36	16.38	16.36	16.37	0.01	64M
D_wide	k=1, dim=4096, 1 pass	16.38	16.40	16.34	16.37	0.03	128M
S_depth	k=1, dim=2048, 2 pass	13.99	13.77	13.95	13.90	0.12	64M

Figure 3 visualizes these results.

Results. The ablation reveals three findings. First, the single expert and the width-doubled expert achieve *identical* perplexity (16.37 vs. 16.37, p=0.69), despite the latter having 2× the expert parameters (128M vs. 64M) and 2× the per-token FLOPs. Doubling expert width provides no benefit over a standard-width single expert. Second, dropping from top-k=2 to top-k=1 costs +8.9% perplexity (Standard → Single), indicating that soft-ensembling of two expert outputs contributes substantially to perplexity. Yet expert

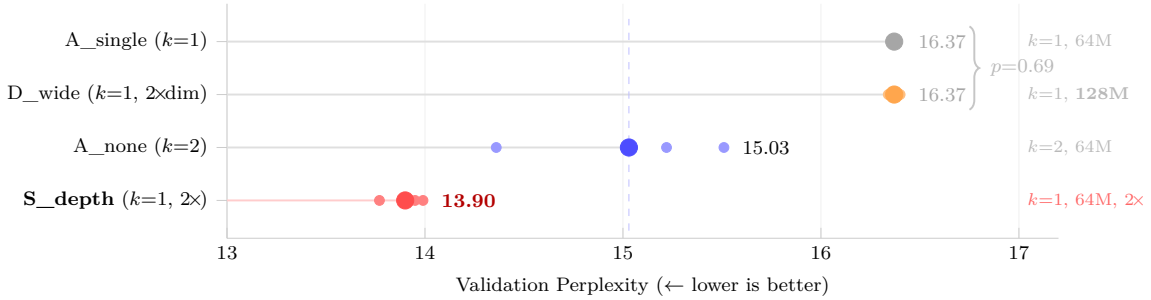

Figure 3: Capacity ablation (3 seeds; large dots are means, small dots are individual seeds). D_wide (128M params) matches A_single (64M) exactly despite double the parameters. The observed improvement is attributable to the second pass, not added capacity.

depth overcomes this disadvantage: despite using top-1 routing, it surpasses the top-k=2 baseline by -7.5%. Third, expert depth outperforms both single-pass baselines by -15.1% ($p = 0.001$). Since all three top-1 conditions share the same routing, the only difference is that expert depth applies the selected expert twice. This 15% improvement is consistent with the second pass through the same expert being the active factor, rather than increased capacity or architectural complexity.

Preliminary downstream evaluation. As a preliminary assessment of downstream quality, we evaluate BLEU (Papineni et al., 2002) on the WMT14 newstest2014 test set (2,737 sentences) using greedy decoding with repetition penalty (single seed; Table 4).

Table 4: Preliminary BLEU evaluation on WMT14 newstest2014 (single seed, greedy decoding with repetition penalty 1.2 and no-repeat 3-gram blocking). A_none is trained for 12 epochs to approximately match S_depth's total wall-clock compute. These are single-run results without variance estimates and should be interpreted with caution.

Condition	Epochs	Val PPL	BLEU	Length Ratio
A_none (k=2, compute-matched)	12	10.46	4.9	1.11
S_depth (k=1, 2 pass)	5	13.99	**11.8**	1.07

Expert depth achieves higher BLEU (11.8 vs. 4.9) despite the baseline reaching lower validation perplexity with more training. We note two important caveats: first, absolute BLEU scores are low for both conditions, reflecting the decoder-only prefix-LM architecture rather than a translation-optimized encoder-decoder; second, these are single-run results using non-standard decoding heuristics, so the BLEU comparison should be treated as preliminary rather than conclusive. The divergence between perplexity and BLEU is nonetheless suggestive—it is consistent with expert depth producing more coherent outputs, as indicated by the length ratio closer to 1.0.

Compute-matched observation. S_depth uses approximately 2.5× more wall-clock time per epoch than A_none (645 vs. 262 minutes on one H200 GPU). When A_none is given approximately equivalent total wall-clock time (12 epochs), it achieves lower validation perplexity (10.46 vs. 13.99) but lower BLEU in this single-run comparison. We present this as an observation: the compute matching is approximate (wall-clock on one H200), the BLEU comparison is single-run, and alternative training schedules or decoding strategies could alter the result.

7 Discussion: Why Does Depth Work?

Each expert E_i is a two-layer FFN: $E_i(\mathbf{x}) = W_2 \cdot \mathrm{GELU}(W_1\mathbf{x} + b_1) + b_2$. Applying it twice creates an effective four-layer transformation with shared weights, analogous to weight-tied iteration in Universal Transformers (Dehghani et al., 2019) and Deep Equilibrium Models (Bai et al., 2019), but applied *within* the sparse expert routing framework rather than across layers. The key difference from these prior approaches is that expert

depth combines sparse routing (selecting *which* function to iterate) with weight-tied iteration (applying it multiple times), whereas Universal Transformers and DEQs iterate the same transformation for all inputs.

The training trajectory (Figure 2) offers insight into how this iterative behavior develops. In early training, the expert is optimized for single-pass processing, and applying it twice initially degrades perplexity. Over training, the expert gradually adapts to process both raw inputs and its own outputs, learning a dual-purpose function that refines on the second pass. This learned refinement behavior is reminiscent of iterative inference in DEQs, where the model converges toward a fixed-point representation through repeated application of the same function. The crossover appears gradual rather than sudden, suggesting continuous adaptation of the expert weights rather than a sharp phase transition.

More broadly, our results suggest a reframing of the purpose of sparse routing in MoE. Standard top-k MoE can be viewed as an *ensemble*: it combines k different expert opinions about each token. Expert depth instead operates as an *iterative solver*: it selects the most relevant expert and allocates more processing time to it. Our finding that, at least on translation, deeper processing by the right expert outperforms shallow processing by multiple experts suggests that the router's primary value may lie in selecting *quality*—the right expert for each token—rather than *quantity*—many expert perspectives.

8 Related Work

Mixture-of-Experts. MoE dates to Jacobs et al. (1991) and Jordan & Jacobs (1994), with the modern sparse gating formulation by Shazeer et al. (2017). Switch Transformer (Fedus et al., 2022) simplified routing to top-1; GShard (Lepikhin et al., 2021) demonstrated distributed MoE training. Expert Choice routing (Zhou et al., 2022) inverts the assignment: experts select tokens rather than tokens selecting experts. Recent large-scale deployments include Mixtral (Jiang et al., 2024) and DeepSeekMoE (Dai et al., 2024); the latter introduced shared experts alongside routed experts as a form of implicit coordination, an approach extended by DeepSeek-V3 (DeepSeek-AI, 2024). ST-MoE (Zoph et al., 2022) studied stability and transferability; GLaM (Du et al., 2022) demonstrated efficient scaling; and sparse upcycling (Komatsuzaki et al., 2023) converts dense models to MoE. All of these treat routed experts as independent processors—our negative results on explicit inter-expert communication are consistent with this design choice. Scaling laws for MoE (Clark et al., 2022; Kaplan et al., 2020; Hoffmann et al., 2022) and differentiable expert selection (Hazimeh et al., 2021) have been studied but do not address inter-expert interaction.

Expert communication and interaction. Chain-of-Experts (Wang et al., 2025) proposes sequential expert composition with different experts at each step, reporting gains on math reasoning with DeepSeek-V2-Lite. Our sequential-different-expert condition (row 9 in Table 1) replicates this approach and finds no significant gain on translation, suggesting task dependence. Our key finding—that same-expert iteration outperforms different-expert chains—is novel relative to CoE, which explicitly routes to different experts per step. Soft MoE (Puigcerver et al., 2024) replaces discrete routing with continuous token-expert mixing. V-MoE (Riquelme et al., 2021) scales vision transformers with MoE layers. OLMoE (Muennighoff et al., 2024) provides open MoE language models. SwitchHead (Csordás et al., 2024) applies MoE to attention heads. Chen et al. (2022) analyze the theoretical properties of MoE layers. None of these works proposes or evaluates iterative expert application. Our learned communication topology uses graph-based parameterization related to GCNs (Kipf & Welling, 2017) and GATs (Veličković et al., 2018), applied to expert adjacency.

Weight-tied iterative processing. Universal Transformers (Dehghani et al., 2019) share weights across transformer layers, applying the same transformation iteratively with halting mechanisms. Deep Equilibrium Models (Bai et al., 2019; 2020) solve for fixed-point representations, finding that implicit-depth models can match explicit-depth ones. Adaptive Computation Time (Graves, 2016) learns to apply variable numbers of iterations per input. ALBERT (Lan et al., 2020) shares parameters across transformer layers to reduce model size. Our expert depth applies weight-tied iteration *within* the sparse expert routing framework: the router selects *which* expert iterates, while the architecture determines *how many* times. This combines the benefits of sparse routing (efficiency) with iterative refinement (depth).

Negative results and reproducibility. Systematic negative findings prevent wasted research effort. Dacrema et al. (2019) showed that many deep learning recommendation models fail to outperform well-

tuned baselines. Lipton & Steinhardt (2019) identified troubling trends in ML scholarship, including failure to beat baselines. Musgrave et al. (2020) found similar issues in metric learning. Bouthillier et al. (2021) emphasized the importance of accounting for variance in benchmarks—a concern we address through multi-seed evaluation. Taken together, these results provide a broad empirical look at expert communication in MoE on a standard translation benchmark, while leaving several configurations supported only by screening-level evidence.

9 Limitations and Future Work

Our models are modest in size compared to current state-of-the-art MoE systems. All experiments use WMT14 En-De translation with a ∼93M parameter model and 32 experts; the depth benefit may differ on other tasks or at larger scale. Notably, Wang et al. (2025) report gains from sequential (different-expert) composition on math reasoning, suggesting task dependence that warrants investigation. Scaling to 1B+ parameters with 256+ experts (Lepikhin et al., 2021; Fedus et al., 2022; Rajbhandari et al., 2022) is an important direction. We evaluate only T=2 sequential passes; more iterations may show continued improvement or diminishing returns. Whether depth and diversity are complementary—e.g., applying each of top-k=2 experts twice—is also an open question. Adaptive computation (Graves, 2016), where the router learns how many iterations each token needs, is a natural extension. S_depth is ∼2.5× slower per epoch than the baseline (645 vs. 262 minutes) due to the sequential nature of the two passes; optimized implementations with fused kernels could reduce this overhead. While we report BLEU on newstest2014, absolute scores are low for both conditions due to the decoder-only prefix-LM architecture; evaluation with an encoder-decoder model or on language modeling benchmarks would provide a more complete picture.

10 Conclusion

This work began with a natural hypothesis—that expert communication should improve MoE performance—and arrived at a surprising conclusion. Across ten approaches and seven experimental axes, we find no statistically significant gain among the four communication configurations tested with 3+ seeds, and six additional screening experiments are consistent with that pattern, while applying the same expert twice yields a significant improvement. The capacity ablation strengthens the case that this gain is not recoverable through increased width alone, while the BLEU evaluation provides only preliminary single-run downstream evidence.

On WMT14 En-De at the scale we evaluate, these results point toward concentrating computation on the most relevant expert through iterative application, rather than spreading it across diverse experts. Whether this finding extends beyond translation, to larger scales, or to settings where experts hold genuinely asymmetric information remains an open question that we hope will motivate further investigation.

References

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Shaojie Bai, Vladlen Koltun, and J Zico Kolter. Multiscale deep equilibrium models. *Advances in Neural Information Processing Systems*, 33, 2020.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, et al. Accounting for variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems*, 2021.

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding the mixture-of-experts layer in deep learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Aidan Clark, Diego de Las Casas, Aurelia Guy, Arthur Sherburn, Ballal Saber, Lisa Anne Hendricks, et al. Unified scaling laws for routed language models. *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

Róbert Csordás, Piotr Piękos, Kazuki Irie, and Jürgen Schmidhuber. SwitchHead: Accelerating transformers with mixture-of-experts attention. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 101–109, 2019.

Damai Dai, Chengqi Deng, Chenggang Zhao, R X Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y Wu, et al. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the ACL*, 2024.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. GLaM: Efficient scaling of language models with mixture-of-experts. *International Conference on Machine Learning*, 2022.

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.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sez, Kai Liu, Kai Chen, John Langford, and Sanjiv Gupta. DSelect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35, 2022.

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

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations*, 2017.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *International Conference on Learning Representations*, 2023.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *International Conference on Learning Representations*, 2020.

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. *arXiv preprint arXiv:2006.16668*, 2021.

Zachary C Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship. *Queue*, 17(1): 45–77, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Niklas Muennighoff, Luca Yang, Weijia Shi, Jungo Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Noah A Smith, Dirk Groeneveld, et al. OLMoE: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. *European Conference on Computer Vision*, 2020.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pp. 311–318, 2002.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. *International Conference on Learning Representations*, 2024.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. DeepSpeed-MoE: Advancing mixture-of-experts inference and training to power next-generation AI scale. *International Conference on Machine Learning*, 2022.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. *Advances in Neural Information Processing Systems*, 34, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the ACL*, 2016.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Zihan Wang, Deli Dong, Hailin Zheng, Yifan Zhu, Xingjian Xie, Jinyang Wang, and Lei Chen. Chain-of-experts: Unlocking the communication power of mixture-of-experts models. *arXiv preprint arXiv:2506.18945*, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35, 2022.

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

A Detailed Negative Results

Table 5 provides per-seed validation perplexity for three communication experiments evaluated with 3 seeds (rows 5, 9, 10 in Table 1). For all remaining rows, we report per-seed values inline below, organized by seed count.

Table 5: Per-seed validation perplexity for experiments with 3 seeds.

Experiment	Condition	s42	s123	s456	Mean
Pre-FFN (32 exp)	A_none	15.16	15.01	14.95	15.04
	E_prefn	14.98	15.05	14.98	15.00
Sequential chain	A_none	15.22	14.36	15.51	15.03
	S_free	14.83	14.80	15.26	14.96
Topology chain	A_none	15.22	14.36	15.51	15.03
	S_topology	14.95	14.98	14.90	14.94

3-seed result not in table above. Post-FFN learnable topology (8 experts, 15 epochs, 3 seeds): A_none s42=15.43, s123=15.38, s456=15.41 (mean 15.41); C_learn s42=15.45, s123=15.36, s456=15.39 (mean 15.40); $\Delta \approx 0\%$, paired t-test $p > 0.9$. Extended training does not create separation.

2-seed screening results. Post-FFN learnable topology (8 experts, 5 epochs): A_none s42=15.97, s123=15.85 (mean 15.91); C_learn s42=16.00, s123=15.81 (mean 15.90); $\Delta = -0.06\%$. Post-FFN (16 experts): A_none s42=15.54, s123=15.66 (mean 15.60); C_learn s42=15.64, s123=15.72 (mean 15.68); $\Delta = +0.53\%$. Post-FFN (32 experts): A_none s42=15.16, s123=15.32 (mean 15.24); C_learn s42=15.12, s123=15.26 (mean 15.19); $\Delta = -0.34\%$. Post-FFN at 32 experts with high specialization (2 seeds): C_postfn s42=15.12, s123=15.43 (mean 15.28) vs. A_none s42=15.16, s123=15.01 (mean 15.09); $\Delta = +1.26\%$. Pre-FFN + Post-FFN combined (2 seeds): F_both s42=15.39, s123=15.51 (mean 15.45) vs. A_none s42=15.16, s123=15.01 (mean 15.09); $\Delta = +2.41\%$.

1-seed screening result. In the multi-domain setting (En-De + En-Fr, top-k=1, seed 42), A_none achieves 22.39 and E_prefn achieves 22.49, for $\Delta = +0.45\%$. This is a single-seed screening result and should be interpreted accordingly.

Mechanism usage diagnostics. Across pre-FFN experiments, gate biases moved from their suppressive initialization of -2.0 to near-zero values (-0.03 to -0.17 across layers at epoch 5), indicating the model actively opened the communication channel. Topology density grew organically from 3.1% (identity initialization) to 15–26% across layers, with deeper layers developing denser connectivity. In the topology-governed sequential experiments, the learned transition graph converged to 97–100% density across all layers and seeds, effectively eliminating the structural constraint.

Domain specialization. In the multi-domain experiment (En-De + En-Fr, top-k=1), expert domain purity was 0.13 (A_none) and 0.20 (E_prefn) at epoch 5, both near the uniform baseline of 1/num_domains = 0.5. Experts assigned tokens from both language pairs with near-equal frequency, indicating that the shared BPE tokenizer produces sufficiently similar subword distributions across languages to prevent domain-specific routing.

B Implementation Details

Model architecture. The model is a Transformer decoder with causal self-attention. Each of the 4 layers contains multi-head self-attention (8 heads, dimension 64 per head) followed by the MoE-FFN sublayer. The

MoE-FFN contains 32 expert modules, each a 2-layer FFN: Linear(512 → 2048) → GELU → Linear(2048 → 512), with layer normalization applied before the MoE sublayer (pre-norm architecture). The output projection and a residual connection follow the MoE output. Total parameters: ∼93M.

Training hyperparameters.

Hyperparameter	Value
Optimizer	AdamW (β_1=0.9, β_2=0.999)
Learning rate	10^{-3}
Weight decay	0.05
LR schedule	Cosine decay (min ratio 0.1)
Warmup steps	500
Batch size	64
Epochs	5
Max sequence length	128 tokens
Label smoothing	0.05
Gradient clipping	1.0
Dropout	0.1
Tokenizer	Shared BPE (SentencePiece)
Training data	4.3M pairs (WMT14 En-De)
Validation data	201K pairs (news-commentary)

Expert depth implementation. In depth mode, the sequential forward pass (Algorithm 1 in the main text) reuses the same router logits at both steps. Since the same expert is selected at both steps, the routing is deterministic: the top-1 expert at step 0 is guaranteed to be reselected at step 1. Gradient flow through the routing scores is maintained via straight-through estimation: scores are multiplied as $s/\text{stopgrad}(s)$, which has unit forward value but allows gradients to propagate to the router parameters. This is implemented in approximately 30 lines of PyTorch within the `ExpertGraph.forward_sequential()` method.

FLOP computation. Each expert FFN performs two matrix multiplications: in_proj $\in \mathbb{R}^{512\times2048}$ and out_proj $\in \mathbb{R}^{2048\times512}$, for a total of $2 \times 512 \times 2048 \approx 2.1$M multiply-accumulate operations per expert activation. With top-k=2, the baseline uses ≈4.2M FLOPs per token per layer. Expert depth (2 passes through one expert) uses the same ≈4.2M FLOPs. The width-doubled control (dim=4096) uses $2 \times 512 \times 4096 \approx 4.2$M FLOPs in a single pass.

C Additional Depth Diagnostics

Per-epoch training dynamics. Table 6 reports per-epoch validation perplexity, training perplexity, routing entropy, and expert load CV for both conditions, averaged across 5 seeds.

Step overlap verification. In depth mode, step overlap (fraction of tokens selecting the same expert at both sequential steps) is exactly 1.0 across all 5 seeds, all 4 layers, and all 5 epochs, confirming that the same expert is applied twice as intended.

Runtime. A_none averages 262 minutes per epoch; S_depth averages 645 minutes per epoch (2.5× overhead) on a single H200 GPU. The overhead is due to the sequential nature of the two expert passes, which prevents parallelization across steps. D_wide averages 221 minutes per epoch; A_single averages 220 minutes per epoch.

Table 6: Per-epoch training dynamics (mean of 5 seeds, layer-averaged). Routing entropy is in bits (max = $\log_2 32 = 5.0$). Load CV is the coefficient of variation of expert token counts (lower = more balanced).

Epoch	Condition	Val PPL	Train PPL	Entropy	Load CV
1	A_none	25.88 ± 0.11	59.38	3.20	3.50
	S_depth	26.82 ± 0.39	67.54	4.90	0.46
2	A_none	22.12 ± 0.11	37.38	3.32	2.87
	S_depth	22.31 ± 0.19	38.34	4.86	0.59
3	A_none	19.19 ± 0.12	32.48	3.18	2.86
	S_depth	18.97 ± 0.16	33.23	4.78	0.74
4	A_none	16.52 ± 0.35	27.52	3.10	3.03
	S_depth	15.78 ± 0.10	28.27	4.64	1.07
5	A_none	15.08 ± 0.44	23.24	3.04	3.10
	S_depth	13.97 ± 0.12	24.04	4.46	1.36