

Peer Review

Review of: "Union of Experts: Adapting Hierarchical Routing to Equivalently Decomposed Transformer"

Evgenii Arsentev¹

1. Independent researcher

1. What the work claims

The paper takes the Megatron-LM idea of splitting a layer's weight matrices across devices without changing what the layer computes, and turns each slice into an expert. The first MLP matrix is cut by columns and the second by rows (§III.A, Fig. 2a); attention is cut by heads. Because the slices sum back to the original layer, the activated experts form a "union" that behaves like one smaller dense model rather than a committee of independent networks. Two routing modes sit on top of this decomposition: data selection, where each expert receives only a subset of patches of the sequence (§III.B.1), and expert selection, where whole samples are sent to the top-k experts (§III.B.2). The resulting blocks are called Selective Multi-Head Attention (SMHA) and Union-of-MLP-Experts (UoME). A section on efficiency (§III.D) replaces the usual per-expert loop with one batched index-select, one scatter-add, and `baddbmm` calls.

The empirical claims are broad. On WikiText-103 and One Billion Word, the model reaches 24.09 and 24.52 perplexity (Table I, row "Ours", PPL columns), below every MoE baseline and slightly below the dense Transformer. On Long Range Arena, it averages 61.72 at 0.37× the Transformer's FLOPs (Table II, row "UoE (ours)"). On ViT-style CIFAR classification, it reaches 82.70 and 55.47 (Table III, row "UoE (ours)"). Ablations on expert count and activation ratio (Table IV, Fig. 5) and component-level comparisons (Table V) follow, plus runtime and memory curves up to sequence length 4096 on one A100 (Fig. 6). Code is released.

I liked the core idea. Using an exact tensor-parallel decomposition as the definition of an expert is simple, it explains in one sentence why the experts should cooperate, and it extends MoE-style sparsity to

attention in a principled way. The causal-mask observation in §III.C.3 (Fig. 4) is neat too: because queries and keys share the same routing indices, the selected mask is always the leading principal submatrix of the original, so no per-expert mask has to be built.

2. *Whether the conclusion holds on the data*

Several things are done well here. The paper tests one mechanism across three quite different task families, reports FLOPs alongside quality in every main table, and includes both an internal ablation (Table IV) and a component-swap ablation (Table V). I also recomputed every "Average" cell in Table II from the five task columns, and all twelve rows are correct; for example, UoE gives $(38.91 + 65.61 + 82.82 + 46.09 + 75.17) / 5 = 61.72$, and DeepSeek-V3 gives $305.20 / 5 = 61.04$. The abstract's headline numbers also check out: the language-modeling gap to DeepSeek-V3 is $(26.96 - 24.09 + 26.41 - 24.52) / 2 = (2.87 + 1.89) / 2 = 2.38$ (Table I), the FLOPs ratio is $(1.74/2.65 + 4.53/5.26) / 2 \approx (0.657 + 0.861) / 2 \approx 0.76$, and the image gain is $(1.19 + 2.31) / 2 = 1.75$ (Table III).

Where I think the conclusions run ahead of the evidence is in four places.

No variance anywhere. Every table reports a single number per cell, and I could not find the number of runs or seeds. That matters most where the margins are small. Against the dense Transformer, UoE's perplexity advantage is $24.23 - 24.09 = 0.14$ on WikiText-103 and $24.70 - 24.52 = 0.18$ on One Billion Word (Table I). On LRA, the lead over DeepSeek-V3 is 0.68 points of average accuracy (Table II). In Table IV, the whole spread of the SMHA data-selection column is $38.44 - 37.89 = 0.55$ points on ListOps, yet the text calls the $n = 8$ result "a considerable degradation in performance." ListOps accuracy is known to move by more than half a point between seeds. I would not call any of these differences established until they are reported as mean and standard deviation over at least three seeds.

The ViT comparison is confounded by RoPE. §IV.C says directly that "by introducing Rotary Position Embedding (RoPE) to transformer model design, our UoE achieves a significant accuracy improvement of 8.25% in CIFAR-10 and 12.87% in CIFAR-100 compared to the standard transformer". I confirm the arithmetic ($82.70 - 74.45 = 8.25$; $55.47 - 42.60 = 12.87$, Table III), but the sentence itself attributes the gain partly to RoPE. The next paragraph notes that Skywork-MoE also uses RoPE, and it is the baseline that jumps to $80.87 / 49.25$, while DeepSeek-MoE and XMoE, which do not, stay at $72.31 / 40.82$ and $70.90 / 41.27$. That pattern suggests positional encoding, not the union of experts, explains much of the 10 to 15 point gaps over those two models. A Transformer + RoPE row, and RoPE added to every MoE baseline,

would separate the two effects. The same question applies, more mildly, to Tables I and II, since §III.C.1 adds RoPE to SMHA as well.

The MoE baselines are unusually weak relative to the dense model. In Table I, every MoE baseline is worse than the dense Transformer by 2.7 to 6.4 perplexity on WikiText-103 (e.g., XMoE 30.60 vs 24.23). In the usual MoE setting, where experts add capacity at matched active compute, the opposite is expected. The implementation details explain part of it: each MoE baseline activates half of its MLP experts, and models were matched on "approximate model size and floating-point operations" (§IV.A), so these MoEs are, in effect, dense models with half the MLP switched off. That is a legitimate comparison, but it is a different claim from "surpasses state-of-the-art MoEs". I would report total and active parameter counts for every row and describe the setting plainly as equal-size sparsification of a dense model. Given that framing, the more informative headline is the one against the dense Transformer: equal quality (a 0.14 to 0.18 PPL difference) at 1.74 vs 2.67 TFLOPs on WikiText-103, which is a genuinely useful result on its own.

Some FLOPs and efficiency claims need tightening. In Table II, Flash Attention is listed at 0.17× FLOPs and later described (§IV.B, Results) as "a linear complexity attention mechanism". FlashAttention computes exact softmax attention; it reduces memory traffic, not arithmetic, so its FLOPs should be close to 1.00×. Its accuracy row is also copied from Dao et al. [11] (Table II footnote), together with Performer and Linear Attention, so those three rows come from a different training pipeline than the rest of the table. Separately, the Introduction says the optimizations reduce "FLOPs by at least 30%" relative to the original version, while §III.D.2 says they give "a speedup of over 30%". Batching with baddbmm changes speed, not operation count, so I think the Introduction should say speedup.

Finally, the runtime and memory claims (a 2.26× average speedup at length 4096 and a 2.68× average memory efficiency improvement) rest on Fig. 6. As rendered on Qeios, I could not read Fig. 6, so I could not check the curves or the averaging. The Introduction states these figures are "over the state-of-the-art MoE method", but §IV.D describes them as an average "relative to the comparison models". Those are different quantities.

3. Reproducibility

The public GitHub repository is the strongest point here, and it is more than many architecture papers offer. The methods are written as equations (Eq. 1 to 44), and the routing, masking, and load-balancing steps can be followed from them. The runtime experiment names its hardware (one A100 80 GB) and tools (PyTorch profiler, nvidia-smi), and FLOPs come from a named tool, calcflops [34].

What is missing from the paper itself:

- No hyperparameter table. For none of the three task families could I find the number of layers, embedding dimension, heads, number of MLP experts, patch count m or cap c , learning rate, schedule, batch size, sequence length, or number of training steps. §IV.A says only that the base configuration follows the Transformer setup of [15] and that all experiments share "the same training settings".
- The balance factor α of the load-balancing loss (§III.C.4, Eq. 42 to 44) is defined, but its value is not given.
- No parameter counts (total or active) for any model in Tables I, III, or V.
- No seeds, no number of runs, no training hardware, or wall-clock time for the main experiments.
- Baselines were reimplemented inside one framework ("with reference to the code implementation of Dai et al. [15]"), but the paper does not say which code or commit each baseline came from, or how DeepSeek-V3's MLA and routing were ported.
- For One Billion Word, the vocabulary and tokenization are not stated, and perplexity is defined per word (Eq. 45); a reader cannot tell whether the comparison is word-level for every model.
- calflops is cited without a version, and the FLOPs figures in Table I are said to exceed the theoretical value because of load imbalance, without saying how that measured value was obtained.

I also want to note that, as rendered on Qeios, most equations appear twice (a plain-text line and a symbol-by-symbol line). I could follow them, but I could not verify every index in Eq. 12 to 18 and Eq. 42 to 44 from this rendering.

4. What specifically to fix

1. §IV.C, Results, first paragraph. The text says UoE surpasses DeepSeek-V3 "by 1.19%/1.83% in CIFAR-10 and 1.93%/6.22% in CIFAR-100". From Table III, UoE minus DeepSeek-V3 on CIFAR-100 is $55.47 - 53.16 = 2.31$, not 1.93. The value 1.93 is the UoE-DD row ($55.09 - 53.16$). The abstract's 1.75 already uses 2.31, so only this sentence needs correcting.
2. §IV.D, External Ablation, last paragraph. XMoE is quoted as "(38.00/32.60)", while Table V, row XMoE, column Wikitext-103 PPL gives 30.60 (as does Table I).
3. Table III: add a Transformer + RoPE row, and ideally RoPE for every MoE baseline, so that the 8.25 and 12.87 point gains can be split between positional encoding and the UoE mechanism.
4. Report mean $\pm$ standard deviation over three or more seeds in Tables I to V. Until then, I would soften "noticeably outperforms" (§IV.A) for the 0.14/0.18 PPL gap to the Transformer, and

"considerable degradation" for the 0.55-point spread in Table IV.

5. **Table V vs Table I.** SMHA alone (standard MLP) reaches 22.68 PPL on WikiText-103 (Table V), better than the full UoE model at 24.09 (Table I) and the Transformer at 24.23. That is either a very interesting finding (the UoME half costs perplexity) or a configuration difference. Please explain it. It also bears on the conclusion that both components help.
6. Give a hyperparameter table per task, including m , c , n , k , r , and α , with total and active parameters for every baseline.
7. **Table II.** Revisit the $0.17\times$ FLOPs entry for Flash Attention and the phrase "linear complexity attention mechanism such as Flash Attention" in §IV.B. Mark in the table body (not only the footnote) which rows are copied from Dao et al. [11].
8. **Table I, row XMoE.** TFLOPs on WikiText-103 is 2.67, identical to the dense Transformer, although the text says MoE baselines activate half their MLP experts; on One Billion Word it is 4.69 against 6.27. A sentence explaining this would help.
9. §IV.C, Comparison Baseline lists "DeepSeek-V2", but Table III reports DeepSeek-V3 [31]. Please make these match.
10. Introduction, the paragraph on efficiency optimizations: change "reduces FLOPs by at least 30%" to a speedup statement, consistent with §III.D.2, and say whether $2.26\times$ and $2.68\times$ are averages over all comparison models or a comparison with one model.
11. Fig. 6: provide the underlying numbers in a table (runtime and peak memory per model per length). It would also help to state what the 8192-length out-of-memory result in §IV.D looks like for UoE: its memory and runtime at that length.
12. §IV.B, Results: "our model achieved an accuracy of 82.82, ranking among the top contenders of transformer variants (best: 82.20 from Performer)". This undersells the result: 82.82 is the highest value in the Retrieval column of Table II, above the full Transformer (82.30) as well as Performer. Saying so directly would be clearer.

5. Recommendation

Minor revisions. The idea is original and clearly explained, the code is public, and the numbers in the main tables are internally consistent apart from the two misquotes above. What the paper needs is modest in compute at these model sizes: seeds, a RoPE-controlled baseline for Table III, a hyperparameter table, and a more careful framing of the MoE comparison. With those, the efficiency result against the dense Transformer would stand on firmer ground than the current wording suggests.

Declarations

Potential competing interests: No potential competing interests to declare.