Mixture-of-Top-k Attention

Efficient Attention via Scalable Fast Weights

Paper by Qishuai Wen, Zhiyuan Huang, Wei He,
Xianghan Meng, and Chun-Guang Li

arXiv:2602.01219v5 (9 May 2026)
Research discussion — 21 September 2026

Build a few sparse memories; reuse them across queries

Central idea

A small set of landmark queries builds reusable top-k key–value sets, plus a compressed global memory.

- Each query reads the shared memory and one routed set.
- LRA: 58.19% $\rightarrow$ 58.91% mean accuracy; 10.6 $\rightarrow$ 2.4 training hours.
- Evidence supports an efficiency–accuracy tradeoff, with task-dependent losses.

Secs. 3.2 and 4.3; PDF pp. 5–6, 9; Table 5.

The missing middle: flexible but reusable sparse patterns

Design	Pattern construction	Main constraint
Block routing	Contiguous K/V blocks	Rigid membership
Per-query top-k	A private set per query	N sets to construct
Compression	A small global memory	Fine details lost
MiTA	m reusable top-k sets	Shared pattern budget

Here an expert is an input-dependent K/V set, not an added FFN with persistent parameters.

Secs. 1–2; PDF pp. 1–4; Table 1.

Attention can be viewed as an input-dependent wide MLP

$$\mathrm{Atten}(q, K, V) = V\,\mathrm{softmax}(K^\top q/\sqrt{d}) = \frac{V\exp(\hat{K}^\top q)}{Z(q)}$$

$$\hat{K} = K/\sqrt{d}, \qquad Z(q) = \sum_{j=1}^{N} \exp(\hat{k}_j^\top q).$$

- $K, V \in \mathbb{R}^{d\times N}$: width grows with sequence length N.
- N queries reading N entries give quadratic arithmetic.
- Caveat: the paper's $\hat{V} = V/Z(q)$ depends on the query. This is an algebraic view, not an ordinary fixed-weight MLP.

Sec. 3.1; PDF pp. 4–5; Eqs. (1)–(4). Caveat: reading analysis.

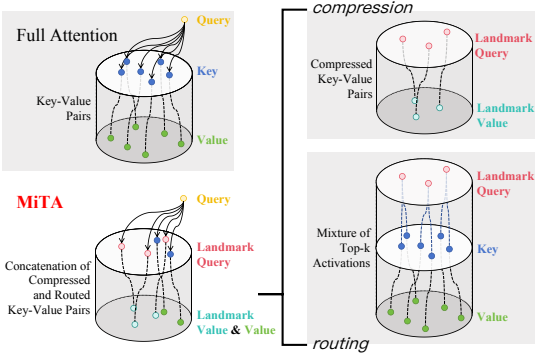
Five design axes separate compression from routing

1. **Strategy**: compress, route, or combine both.
2. **Count**: one shared expert, m experts, or N private experts.
3. **Type**: linear layer, attention/MLP, or a more general module.
4. **Construction**: locality, learned projection, landmark probing, etc.
5. **Topology**: how queries are assigned to experts.

MiTA: m routed experts plus one shared expert of width m.

Table 1, Sec. 3.1, Appendix B; PDF pp. 2, 5, 15.

MiTA combines a global summary with original entries



Original Figure 2, PDF p. 3; Sec. 3.2, pp. 5–6.

1. Pool Q into m landmarks.
2. Probe all keys once per landmark.
3. Keep top-k original K/V and form a weighted summary.
4. Route queries and jointly normalize the two memories.

One landmark builds one deformable expert

$$\tilde{Q} = \mathrm{AvgPool}_{2D}(Q) \in \mathbb{R}^{d \times m}, \qquad S^{\mathrm{kv}} = K^{\top} \tilde{Q} / \sqrt{d} \in \mathbb{R}^{N \times m}$$

$$\mathcal{I}_i = \mathrm{Top}_k(K^{\top} \tilde{q}_i), \qquad K^{(i)} = K_{:,\mathcal{I}_i}, \quad V^{(i)} = V_{:,\mathcal{I}_i}.$$

- Selection depends on both pooled queries and key content.
- Membership can cross spatial boundaries and overlap across experts.
- The retrieved K/V remain original entries, not pooled values.

Sec. 3.2; PDF pp. 5–6; Eqs. (5)–(7).

The same landmarks also form the global memory

$$\tilde{V} = V \,\mathrm{softmax}_N(S^{\mathrm{kv}}) \in \mathbb{R}^{d\times m}, \qquad \tilde{\mathcal{E}}(q) = \mathrm{Atten}(q, \tilde{Q}, \tilde{V}).$$

- Each landmark value summarizes all N original values.
- Landmark *queries* now serve as compressed *keys*.
- $K^\top \tilde{Q}$ supports both top-k selection and compression.
- $\tilde{Q}^\top Q$ supports both routing and shared-memory lookup.

Compression and routing share computations rather than acting as unrelated add-ons.

Sec. 3.2; PDF p. 6; Eqs. (8)–(9).

A single softmax couples detail and global context

$$K^\star = [\tilde{Q}, K^{(e_1(q))}, \ldots, K^{(e_s(q))}], \quad V^\star = [\tilde{V}, V^{(e_1(q))}, \ldots, V^{(e_s(q))}]$$

$$\mathrm{MiTA}(q) = V^\star \,\mathrm{softmax}((K^\star)^\top q/\sqrt{d}).$$

- Each query reads $m + ks$ entries; all experiments use $s = 1$.
- Sharing an expert does not force queries to share attention weights.
- Separate branch outputs must be combined with partition functions:

$$o = \frac{Z_c o_c + Z_r o_r}{Z_c + Z_r} \quad \text{(not simply } o_c + o_r\text{)}.$$

Sec. 3.2; PDF p. 6; Eqs. (10)–(12). Last equation: direct derivation.

Linear scaling requires a fixed memory budget

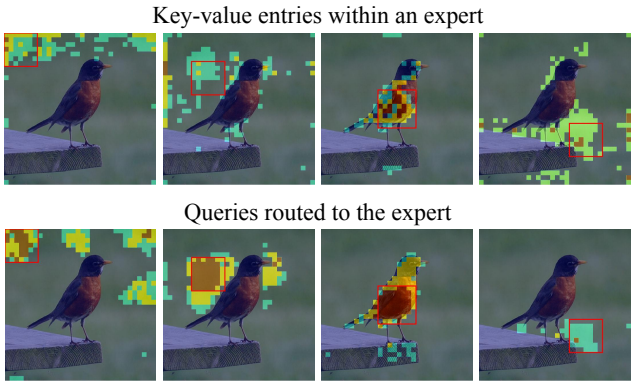
- Main attention arithmetic: $O(Nmd + Nksd)$, including landmark probing.
- Linear in N only if m, k, s, d are held fixed.
- Top-k, query grouping, random gather, and fusion add real overhead.
- Random K/V access can make the pipeline memory-bound.

Changing the budget changes the claim

If $m \approx k \propto \sqrt{N}$, the main arithmetic becomes $O(N^{3/2}d)$ for fixed s (our analysis).

Sec. 3.2, PDF p. 6; Algorithm 1, p. 16.

Local landmarks can retrieve nonlocal information



Original Figure 3; PDF p. 5. Heatmaps average attention heads.

- Red box: landmark pooling region.
- Top: selected K/V.
- Bottom: queries routed to that expert.
- Neither set is confined to the box.

Evaluation uses different budgets for different tasks

Evaluation	Input	(m, k, s)	Key metric
ImageNet-1K	224^2, patch 16	$(25, 25, 1)$	Accuracy
ADE20K	512^2 / 640^2	$(49, 49, 1)$	mIoU
LRA	1K–4K tokens	$(128, 128, 1)$	Acc., steps/s
Speed curve	1K–1024K	$mk \approx N/4$	Samples/s

- LRA and speed measurements: RTX 4090, 24 GB.
- Standard attention uses a fused/FlashAttention implementation.
- Speed curve: 3 layers, embedding 128; batch tuned per run.

Secs. 4.1–4.3; PDF pp. 8–9; Figure 5 legend distinguishes the speed budget.

Basic MiTA nearly matches full attention on DeiT-S

Attention / extras	Tiny acc.	Small acc.	Small FLOPs
Standard	72.2	79.9	4.6G
Agent	70.3	78.5	4.4G
Agent + bias	71.1	79.1	4.4G
MiTA	71.1	79.8	4.4G
MiTA + DWC	73.4	80.6	4.4G
MiTA + DWC + gate	—	81.2	4.7G

Base MiTA: -1.1 / -0.1 accuracy points vs. full attention.
The gains above full attention include extra architectural components.

Table 2, PDF p. 7; Sec. 4.1, p. 8. Accuracies in percent.

A stronger architecture improves MiTA, but preserves a gap

Model	Accuracy (%)	FLOPs
ViT-5-S	82.2	4.7G
DeiT-III-S	81.4	4.6G
MiTA-ViT-5-S	81.3	4.5G
MiTA-ViT-5-S + DWC	81.7	4.5G

- Best listed MiTA variant is 0.5 points below ViT-5-S.
- Fewer FLOPs, but no universal accuracy dominance.
- This is a comparison of complete models, not an isolated attention ablation.

Selected rows of Table 3, PDF p. 7.

Segmentation reduces FLOPs at a measurable accuracy cost

Backbone	FLOPs: full → MiTA	mIoU: full → MiTA	Δ points
Tiny, 512^2	12G → 7G	39.1 → 36.5	−2.6
Small, 512^2	33G → 25G	45.8 → 44.5	−1.3
Base, 512^2	111G → 95G	49.3 → 48.1	−1.2
Large, 640^2	622G → 508G	53.3 → 50.5	−2.8

- Matched comparisons use the same CBT segmentation head.
- 14–42% fewer reported FLOPs; 1.2–2.8 mIoU points lost.
- Backbones were pretrained with standard attention, not native MiTA.

Table 4, PDF p. 8; Sec. 4.2, pp. 8–9.

LRA combines near-full accuracy with faster training

Task	Full acc.	MiTA acc.	Steps/s speedup
ListOps (2K)	37.35	37.20	4.1×
Text (4K)	63.63	63.47	9.9×
Retrieval (4K)	79.90	79.88	10.3×
Image (1K)	40.38	40.72	2.4×
Pathfinder (1K)	69.68	73.26	2.7×
Mean accuracy	58.19	58.91	

Total training: 10.6 $\rightarrow$ 2.4 hours (77.4% less).

Accuracy gains are concentrated in Pathfinder, not uniform across tasks.

Table 5 and Sec. 4.3, PDF p. 9. Accuracy in percent; same training recipe.

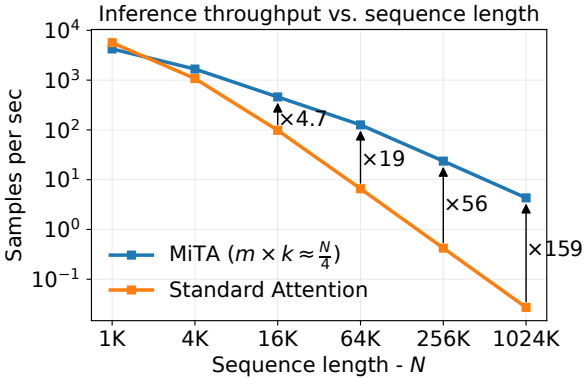
Compression helps MiTA, while Agent remains faster overall

Method	Mean accuracy (%)	Total hours
Full attention	58.19	10.6
Agent Attention	58.22	1.9
MoBA, route-only	56.74	4.6
MiTA, route-only	58.81	5.5
MiTA, full	58.91	2.4

- Route-only variants preserve the attended-entry budget by increasing k.
- More gathered entries can cost more than a compact shared summary.
- MiTA is more accurate than Agent here, but takes longer to train.

Table 5, PDF p. 9. No multi-seed uncertainty is reported.

Long-context throughput gains are setting-specific



- At 1024K: about 159× throughput.
- 3-layer, width-128 model on RTX 4090.
- Batch size optimized independently per run.
- Legend: $mk \approx N/4$.

This does not establish fixed-batch latency or million-token task accuracy.

Original Figure 5 and Sec. 4.3; PDF p. 9.

Ablations favor reusable wider experts

Landmark extraction (accuracy)

- 2D average pooling: 71.1
- Learned parameters: 66.3
- Convolution: 67.0
- Token merging: 71.3

Equal mk budgets

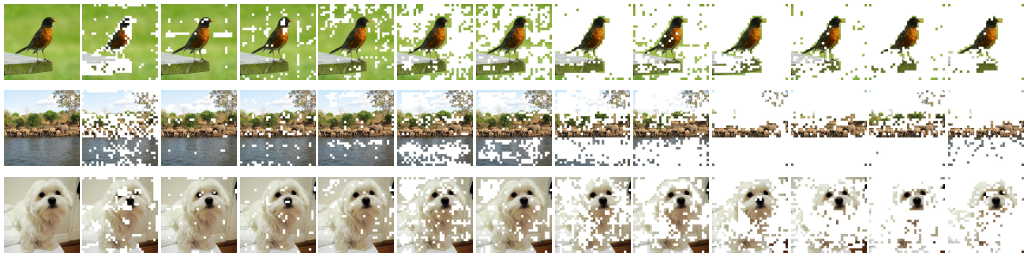
- $(16, 25)$: 71.0
- $(25, 16)$: 70.5
- $(25, 36)$: 71.7
- $(36, 25)$: 71.2

Full: 71.1; compression-only: 70.3; routing-only: 70.8.

Gradient conflict is an author hypothesis, not directly measured evidence.

Table 6, Appendix A; PDF pp. 14–15. Values are accuracy percentages.

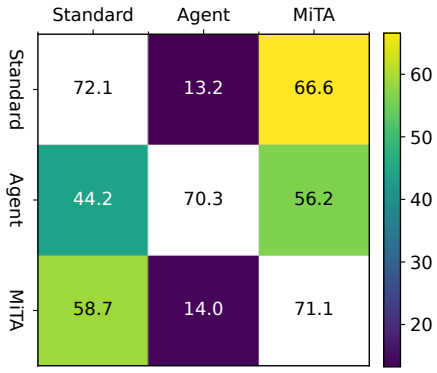
The pruning effect is selective retrieval, not token deletion



- Rows: training images; columns: input followed by successive layers.
- Later experts select class-relevant regions more strongly.
- Unselected tokens can still contribute through the global summary.

Original Figure 4; PDF p. 7. Qualitative examples, aggregated over heads.

Fixed-weight swapping: the plotted data contradict the text



Standard-trained models

MiTA at inference

- Tiny: $58.7/72.1 = 81.4\%$ retention.
- Small: $71.3/80.0 = 89.1\%$ retention.
- Appendix C claims over 95% in this direction.

The figure and text disagree.

Figure 9 (Tiny panel shown), PDF p. 16; conflicting statement in Appendix C, p. 17.

Finetuning transfers better to MiTA than to Agent

Attention	Tiny	Small	Base	Large
Standard	76.9	81.2	84.4	85.9
Linear	73.2	78.6	79.8	80.7
Agent (49)	74.6	79.7	81.4	83.5
Agent (64)	74.7	79.9	81.5	83.8
MiTA	75.6	80.9	82.8	85.5

- ImageNet-21K initialization; 50 epochs of ImageNet-1K finetuning.
- Standard-to-MiTA gaps remain 0.3–1.6 accuracy points.
- This is adaptation with training, distinct from frozen-weight swapping.

Table 7 and Appendix C, PDF p. 17. Accuracy in percent; AdamW optimizer.

What is established, and what remains open?

1. **Established design**: reusable, content-adaptive K/V experts coupled to compressed global memory.
2. **Established tradeoff**: strong LRA training gains; classification and segmentation retain measurable accuracy gaps.
3. **Open extension**: causal construction, incremental caching, $s > 1$, and LLM decoding.
4. **Reading caution**: fixed-budget scaling, pretrained-vs-native comparisons, and internal figure/text inconsistencies.

Secs. 3–4; Appendices A–D; PDF pp. 5–9, 14–17.

Reference and discussion prompts

Qishuai Wen, Zhiyuan Huang, Wei He, Xianghan Meng, and Chun-Guang Li.
Mixture-of-Top-k Attention: Efficient Attention via Scalable Fast Weights.
arXiv:2602.01219v5, 9 May 2026.

arxiv.org/abs/2602.01219v5 github.com/QishuaiWen/MiTA

- What budget allocation best preserves rare details?
- Can experts and summaries be updated causally at low cost?
- How much of the measured gain survives at matched quality and batch size?

Reading based on the version-matched PDF, main.tex, and original figures. No code reproduction.

Backup: larger inference budgets can help or hurt

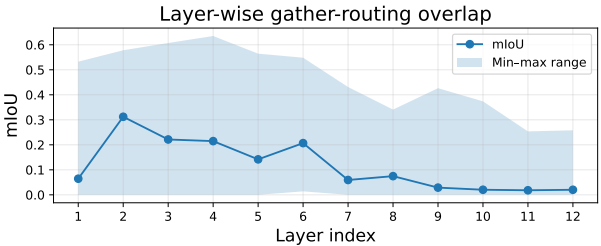
Fixed weights: MiTA-ViT-T at 512^2 resolution

m	k	Accuracy (%)	Configuration
25	25	77.1	Training baseline
25	36	77.6	Wider experts
36	36	77.7	More and wider experts
100	100	75.2	Much larger inference budget

- Only inference hyperparameters change; no weight updates.
- A modest increase helps in these selected cells.
- A larger memory budget does not guarantee higher accuracy.

Selected cells transcribed from Figure 10(a), PDF p. 16; Appendix C, p. 17.

Backup: routing is not identical to hard spatial clustering



Original Figure 8; Appendix B; PDF p. 15.

- Compare gathered K/V positions with routed-query positions.
- Vertical axis: mIoU, averaged over experts and heads.
- Modest overlap supports nonlocal information flow.
- It does not prove retrieval completeness or semantic correctness.

Backup: two implementation details need explicit handling

1. Compute expert labels $a = \arg\max(\tilde{Q}^\top Q)$.
2. Sort queries by a; retain the inverse permutation.
3. Build cumulative lengths from counts of expert labels, not counts of the query permutation.
4. Run grouped attention; restore the original query order.
5. Combine branches using their log-sum-exp normalizers.

Algorithm 1 conflates the label-count and permutation variables and omits an explicit inverse permutation. This is a pseudocode observation, not a repository audit.

Algorithm 1, PDF p. 16; corrections above are reading analysis.

Backup: $m = k = N$ does not generally recover full attention

Appendix A makes this claim, but Eq. (10) still includes the shared branch.

$$d = 1, \quad Q = K = [1, 0], \quad V = [1, 0], \quad m = k = N = 2, \quad s = 1.$$

For $q = 1$, let $a = e/(e + 1) \approx 0.7311$. Then

$$\tilde{V} = [a, 0.5], \qquad \mathrm{Atten}(q, K, V) = a, \qquad \mathrm{MiTA}(q) = \frac{ea + 0.5 + e}{2e + 2} \approx 0.7000.$$

Additional conditions or removal of the shared branch are needed.

This is a constructed formula check, not a reported paper experiment.

Claim: Appendix A, PDF p. 14. Formula: Eqs. (10)–(12), p. 6. Counterexample: our analysis.