Transformer-FSQ

Discrete Latents for Extreme KV-Cache Compression

Andras Szecsenyi Thomas L. Lee Vaisakh Shaj
Elliot J. Crowley Amos Storkey

The University of Edinburgh

arXiv:2611.09598v1 10 November 2026

Publication not found as of 10 Nov 2026 | Code forthcoming: author-linked repository is empty

The KV cache can exceed the model on local hardware

- KV grows linearly; Qwen3-8B cache uses 144 KiB/token across all layers.
- Qwen3-8B-Base weights occupy 15.3 GiB in half precision.
- On a 16 GiB GPU, about 0.74 GiB remains for cache and runtime.
- That fits only about 5.4k uncompressed tokens before activations.

Paper, §1, PDF p. 1, Fig. 1; Table 3, PDF p. 22.

The target

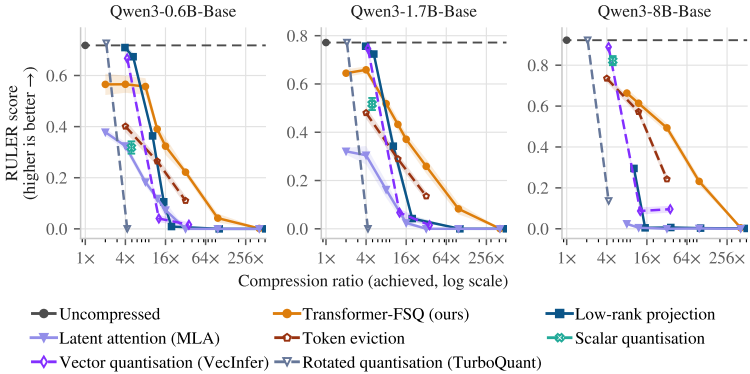
Extend context capacity by shrinking the *per-token state*, while keeping standard Transformer attention.

256× compression

1.39M estimated tokens

This is a capacity estimate, not a million-token quality test.

FSQ stays useful after other cache methods collapse



- At $\leq 5\times$, VecInfer and Palu are often stronger.
- At $12\times$–$97.5\times$, Transformer-FSQ has the highest evaluated RULER score across these Qwen3 sizes.

Paper, Fig. 2, PDF p. 2. RULER: 13 tasks, 50 items each, 8,192-token context.

Compress once, before K/V expansion

Standard attention

Residual state x_t

$$x_t \to W_k x_t, \;\; W_v x_t \to \text{store K and V}$$

Each KV head's floating-point keys and values remain in the cache.

Transformer-FSQ

Residual state x_t

$$x_t \to E x_t \to Q_{\mathrm{FSQ}} \to \text{one packed code } z_t$$

All KV heads reconstruct their K/V from the same code.

The bottleneck represents the shared state behind K and V, not separate post-hoc quantizers.

Paper, §4, PDF p. 5; architecture cross-check: Fig. 4, PDF p. 6.

A wide product code uses four bits per coordinate

For $x_t \in \mathbb{R}^C$, encoder $E \in \mathbb{R}^{G \times C}$ produces G latent coordinates:

$$z_t = Q_{\mathrm{FSQ}}(Ex_t) \in \mathcal{L}_1 \times \cdots \times \mathcal{L}_G.$$

With L levels per coordinate:

$$\tilde{u} = h \tanh(u + \operatorname{atanh}(\delta/h)) - \delta, \qquad \hat{u} = \tilde{u} + \mathrm{sg}(\lfloor \tilde{u} \rceil - \tilde{u}).$$

- Main experiments: $L = 16$, hence $b = \lceil \log_2 L \rceil = 4$ bits.
- Cache size: $\lceil Gb/8 \rceil$ bytes per token per layer.
- The L^G grid is implicit; no codebook table or nearest-neighbour lookup.
- Straight-through gradients make rounding trainable; the forward pass still uses rounded codes.

Paper, §3 and §4.1, PDF pp. 4–5, Eq. (2)–(4). For $L = 16$: $h = 7.5075$, $\delta = 0.5$.

Attention pools codes before decoding values

$$\hat{k}_t = \mathrm{RoPE}_t(N_k(W_k z_t)), \qquad \hat{v}_t = W_v z_t$$

Values

$$\sum_t a_t^{(h)} W_v z_t = W_v \left(\sum_t a_t^{(h)} z_t \right)$$

Pool the compact codes, then apply W_v once.

Keys

RoPE depends on each cached position t.

$$q^\top \mathrm{RoPE}_t(N_k(W_k z_t))$$

Reconstruct keys at each position before scoring.

This asymmetry saves value-side work, but key reconstruction is the main added arithmetic.

Paper, §4.1 and §4.3, PDF pp. 6–7, Eq. (5)–(6).

PCA and ridge regression convert a pretrained model

1. Capture 32,768 ClimbMix residual tokens from the original model.
2. Initialize encoder rows from PCA directions:

$$E_{i,:} = \frac{u_{i \bmod C}^{\top}}{3\sigma_{i \bmod C}}.$$

3. Quantize projected activations; fit a ridge decoder:

$$D = (Z^{\top}Z + \lambda I_G)^{-1} Z^{\top} X, \quad \lambda = 1.$$

4. Compose D with original K/V projections, then fine-tune only the inserted encoder/decoder.

Training budget

Backbone frozen; ClimbMix; 0.25 tokens per original parameter for reported checkpoints.

Paper, §4.2 and Appendix C.2–C.3, PDF pp. 6–7, 22–23, Eq. (7)–(8).

The comparison prices each cache by actual bytes

Models and baselines

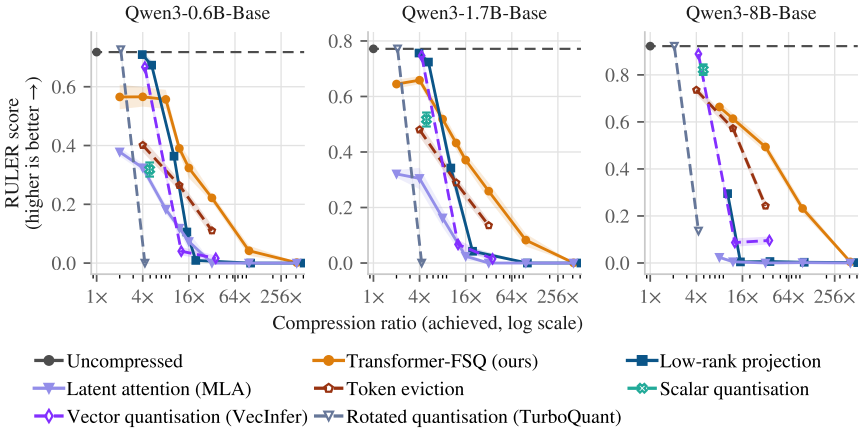
- Qwen3-0.6B, 1.7B, 8B Base
- PyramidKV: token eviction
- KIVI: scalar quantization
- VecInfer: vector quantization
- Palu: low-rank projection
- TransMLA: latent attention

Evidence

- RULER: 13 tasks, 50 items, 8,192 tokens
- LongBench: narrativeqa, gov_report, trec
- Decode: full-model tokens/s vs native SDPA
- Charge metadata, windows, rotary state, and auxiliary cache bytes

Paper, §4.4–5.1 and Appendix C.4–C.5, PDF pp. 7–8, 22–24.

RULER: FSQ leads where compression gets extreme



Paper, Fig. 2 and §5.2 RQ1, PDF pp. 2, 9–10. Scores use up to 3 seeds; Qwen3-8B uses 1.

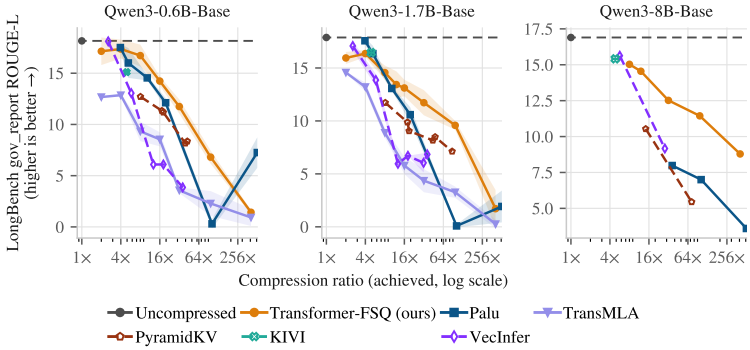
Retrieval quality still falls as the cache gets smaller

Model	12×	32×	97.5×
Qwen3-0.6B	0.39	0.22	0.04
Qwen3-1.7B	0.43	0.26	0.08
Qwen3-8B	0.61	0.49	0.23

- Uncompressed means: 0.72, 0.77, and 0.92, respectively.
- Single-key retrieval survives deepest; multi-key distractors and variable tracking decline earlier.

Paper, Tables 9–11, PDF pp. 37–39. Scores are RULER means across 13 tasks.

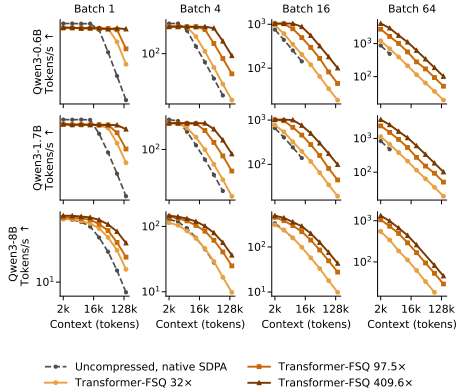
LongBench transfers the gain to summaries, not every task



- Qwen3-8B gov_report: 12.5 ROUGE-L at 32×, 11.4 at 97.5×; Palu scores 7.0 at 102.4×.
- On Qwen3-8B narrativeqa, PyramidKV reaches 18.8 F1 at 26.6×; FSQ scores 8.5 at 32×.

Paper, Fig. 5, PDF p. 9; Table 14, p. 42. Qwen3-8B uses one seed.

Fused decoding offsets key reconstruction at 32×



- Qwen3-8B: 1.01× native SDPA at 32× (95% interval 0.98–1.05).
- At 97.5×: 1.54× (1.16–1.88); at 409.6×: 1.63× (1.20–2.26).

Paper, Fig. 6 and Appendix D.5, PDF pp. 11, 29–30. Full-model decode on NVIDIA L40S.

The kernel reads codes and never writes reconstructed K/V

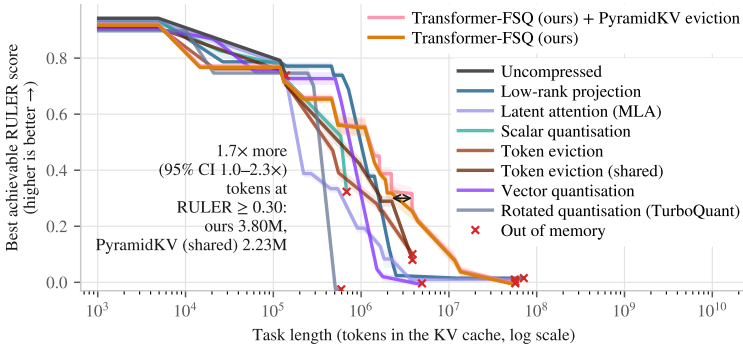
1. Read packed codes from HBM and unpack to registers.
2. Reconstruct keys with W_k on tensor cores; apply key norm and RoPE on chip.
3. Compute scores and online softmax; accumulate attention-weighted codes.
4. Merge partials and apply W_v once, after pooling.

At 32×

128 B/token/layer of packed codes replace 4,096 B/token/layer of half-precision K/V.

Paper, Appendix D.4, PDF pp. 27–28, Fig. 13. The figure's stored bytes do not equal all repeated head reads.

Eviction and FSQ can extend projected cache capacity



- At RULER ≥ 0.30: 3.80M projected tokens vs 2.23M for shared-head PyramidKV (1.7×, 95% interval 1.0–2.3).
- Uses a shared-head PyramidKV variant and extrapolates quality measured at 8,192 tokens.

Paper, Appendix D.3, PDF pp. 26–29, Fig. 11. Capacity is projected from 8K RULER.

Training through the grid beats post-hoc quantization

Method	Width	RULER
Continuous latent	40	0.001
Post-hoc FSQ	160	0.136
Transformer-FSQ (STE)	160	**0.177**
Trained min–max 4-bit	128	0.151

80 B/token/layer (51×), Qwen3-0.6B; mean over 3 seeds.

Paper, Appendix E.3, PDF pp. 32, 34–35, Tables 7–8. Qwen3-0.6B, 3 seeds.

- A wide low-bit latent carries more coordinates than a narrow FP16 latent.
- At 16 levels, RULER also exceeds 4- and 256-level grids at both tested budgets.

FSQ adapts to a 4-bit weight backbone

Qwen3-8B setting	RULER	Change
Uncompressed, 16-bit weights	0.922	–
Uncompressed, NF4 weights	0.913	−0.009
FSQ 32×, then quantize weights to NF4	0.382	−0.111
FSQ 32×, fine-tuned on frozen NF4	0.464	−0.029 vs 16-bit FSQ

Fine-tuning the discrete cache bottleneck on the quantized backbone recovers 74% of the post-hoc loss.

Paper, §5.2 RQ5, PDF pp. 10–11. NF4 quantizes decoder linear layers in groups of 64.

The evidence supports a regime-specific conclusion

Where it works

- RULER 8K: strongest evaluated method at 12×–97.5×.
- gov_report: competitive at 32×–97.5×.
- 32× Qwen3-8B decode: matches native SDPA.

What remains open

- VecInfer/Palu can be better at moderate compression.
- PyramidKV is stronger on 8B narrativeqa.
- No code/math/many-shot reasoning test; RULER uses 8K.
- 8B quality results use one training seed.

Paper, §5–7, PDF pp. 8–12; Appendix D and G, pp. 25–29, 37–42.

Takeaways

1. Cache a shared discrete state before K/V expansion; use the same code for all KV heads.
2. Train through the low-bit grid so its coordinates fit the inference representation.
3. The strongest evidence is at extreme compression, with an attention-aware fused decoder.

Paper

Szecsenyi et al., arXiv:2611.09598v1 (2026)

Author-linked repository, currently empty: transformer-fsq.

Backup: the grid size is an allocation choice

Levels per coordinate	144 B/token	80 B/token	Bits/coordinate
4	0.178	0.104	2
16	**0.232**	**0.177**	4
256	0.161	0.035	8

Each cache budget is fixed; changing level count changes the available latent width.

Paper, Appendix E.3, PDF p. 35, Table 8.

Backup: capacity and accuracy are counted by different protocols

- RULER accuracy is measured at a fixed 8,192-token context.
- Cache bytes include all per-token data and metadata; ratios are priced at 8,192 tokens.
- The 16 GiB frontier maps those fixed-context scores to an estimated number of retained tokens.
- For the stacked capacity curve, prefill peak memory sets the ceiling.

Paper, §4.4 and Appendix D.3, PDF pp. 7, 26–29, Fig. 11.

Backup: main positioning choices

- Post-projection FSQ keeps the pretrained model's key normalization and RoPE path.
- FSQ-pre moves RoPE before the encoder, but query-key normalization breaks relative-position preservation in the reported probe.
- Per-head codes and nonlinear encoders worsened validation BPB; a shared code and linear encoder performed better.
- At moderate ratios, repeating PCA directions can create redundant code coordinates and parameter overhead.

Paper, §4.3 and Appendix E.1–E.2, E.7–E.8, PDF pp. 7, 30–35.

References

- Andras Szecsenyi, Thomas L. Lee, Vaisakh Shaj, Elliot J. Crowley, and Amos Storkey. *Transformer-FSQ: Discrete Latents for Extreme KV-Cache Compression*. arXiv:2611.09598v1, 2026. arXiv page.

- Qwen3-0.6B-Base, Qwen3-1.7B-Base, and Qwen3-8B-Base are evaluated using the paper's reported model configurations.
- RULER and LongBench scores follow the benchmark protocols described in the paper, §5.1 and Appendix C.5.

Publication not found as of 10 Nov 2026. Code forthcoming: author-linked repository.