LASER: LOW-RANK ACTIVATION SVD FOR EFFICIENT RECURSION

Ege Çakar, Ketan Raghu & Lia Zheng
Harvard University
Cambridge, MA, USA
{ecakar, karaghu, liazheng}@college.harvard.edu

ABSTRACT

Recursive architectures such as Tiny Recursive Models (TRMs) perform implicit reasoning through iterative latent computation, yet the geometric structure of these reasoning trajectories remains poorly understood. We investigate the activation manifold of TRMs during recursive unrolling and find that activations occupy an effectively linear, low-dimensional subspace whose principal directions can be tracked dynamically with cheap power iterations. This suggests that weight-sharing concentrates iterative computation along a small number of dominant eigendirections, and we find that this concentration varies sharply across computational sites. We exploit this structure through LASER (Low-Rank Activation SVD for Efficient Recursion), a dynamic compression framework that maintains an evolving low-rank basis via matrix-free subspace tracking with a fidelity-triggered reset mechanism, achieving $\sim 60\%$ activation memory savings with no statistically significant accuracy degradation. Our analysis raises questions about how recursive architectures allocate representational capacity during implicit reasoning, and whether this concentration can be exploited to improve the efficiency and stability of latent computation.

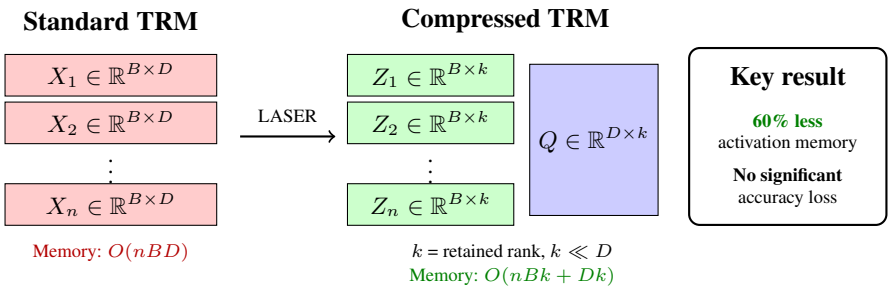

Figure 1: **LASER overview.** Standard TRM training stores full activations $X_1, \ldots, X_n$, with each $X_i \in \mathbb{R}^{B\times D}$, across recursive steps. LASER instead stores compressed coefficients $Z_1, \ldots, Z_n$, where $Z_i = X_i Q \in \mathbb{R}^{B\times k}$, together with a shared low-rank basis $Q \in \mathbb{R}^{D\times k}$, where $k \ll D$ is the retained rank.

1 INTRODUCTION

Implicit reasoning, the ability to perform multi-step computation through iterative latent processing rather than explicit chain-of-thought generation, has emerged as a promising paradigm for both parameter-efficient and capable reasoning models. Tiny Recursive Models (TRMs) (Jolicoeur-Martineau, 2025) exemplify this approach: a single transformer block is unrolled over many recursive time-steps, enabling deep computation with minimal parameters. While TRMs achieve strong performance on structural reasoning tasks, their training requires storing activations across all unrolled steps for backpropagation through time, creating a memory bottleneck that limits practical reasoning depth.

This bottleneck raises a natural question: what is the intrinsic dimensionality of these stored reasoning trajectories? If activations across 24 recursive steps span a high-dimensional, nonlinearly curved manifold, then reducing memory requires powerful but expensive methods. However, if the manifold is low-rank and approximately linear, then the iterative reasoning process is far more constrained than the architecture nominally permits, and the memory problem becomes tractable.

We find the latter. TRM activations occupy an effectively linear, low-dimensional subspace whose principal directions can be tracked dynamically with cheap power iterations. This finding has both interpretive and practical significance. Interpretively, it suggests that weight-sharing concentrates iterative computation along a small number of dominant eigendirections, raising questions about the effective utilization of reasoning depth. Practically, it motivates **LASER** (Low-Rank Activation SVD for Efficient Recursion), a dynamic compression framework that exploits this structure to achieve $\sim 60\%$ activation memory savings with negligible accuracy loss.

This structure also suggests that recursive models are especially well-suited to activation compression. In standard backpropagation through time, activation memory grows directly with the number of recursive steps because a full hidden state must be retained at each step. LASER instead stores a compressed representation at each step together with a shared low-rank basis, so its advantage becomes larger as recursion depth increases. We evaluate this effect in the 24-step TRM setting used in prior work, both to remain comparable to the original architecture and to stay within our training budget.

Our contributions are:

- **Low-rank geometry:** TRM activations exhibit strong spectral decay and are well-approximated by a linear low-rank subspace.
- **Adaptive tracking:** LASER tracks this evolving subspace with matrix-free power iteration and fidelity-triggered fallback.
- **Practical compression:** LASER reduces activation memory by about 60% with no statistically significant loss in performance.
- **Quantization compatibility:** TRM activations remain stable under INT8 quantization after low-rank projection.

2 Related Work

2.1 Activation Checkpointing and Quantization

Standard memory-reduction techniques generally fall into two categories: *Rematerialization* and *Quantization*. Gradient Checkpointing (or Rematerialization) reduces memory by discarding intermediate activations and recomputing them during the backward pass. While effective, it imposes significant computational overhead when done aggressively, and we propose that LASER can be applied in parallel with non-aggressive checkpointing. Scalar quantization methods like ActNN (Chen et al., 2021) and GIST (Jain et al., 2018) compress activations by reducing precision (e.g., to 2-bit or 4-bit integers). While efficient, pure quantization treats activations as statistically independent scalars, ignoring the strong correlations between features. In this work, we demonstrate that quantization can complement subspace methods: LASER applies quantization to its low-rank *residuals*, yielding higher compression rates than either method alone.

2.2 Low-Rank Activation Compression

A more structurally aware approach to compression exploits the low-rank nature of neural activations. If the activation matrix $X \in \mathbb{R}^{N \times d}$ (where $N = B \times L$) resides in a subspace of rank $k \ll d$, it can be stored as two smaller factors $U \in \mathbb{R}^{N \times k}$ and $V \in \mathbb{R}^{d \times k}$.

2.2.1 Static and Randomized Projections

Recent work has attempted to apply this principle to training. LANCE (Apolinario & Roy, 2025) employs a "one-shot" Higher-Order SVD (HOSVD) to fix a compression basis at initialization. This eliminates decomposition overhead, but assumes the activation subspace at epoch 0 is the same as

at epoch 100. Our experiments show this stationarity assumption fails for recursive models, where the manifold evolves as the model learns to reason. GALE (Muhsin et al., 2026) uses randomized projections (sketching) to compress states. However, applying randomized sketching to activations at every layer introduces significant computational overhead.

2.2.2 SUBSPACE TRACKING

Our method draws inspiration from signal processing techniques for subspace tracking, such as the Projection Approximation Subspace Tracking (PAST) algorithm (Yang, 1995). Unlike LoRAct (Shi et al., 2025), which uses sampling-based orthogonal decomposition, LASER applies a power-iteration scheme directly to the activation stream. Furthermore, we address the primary weakness of iterative tracking, error accumulation, by implementing an adaptive fallback: when the tracked subspace fails to explain the variance, we trigger a sparse, exact SVD update to realign the basis.

While the above methods focus on compression as an engineering goal, our work additionally treats compressibility as a *diagnostic*: the degree to which activations admit low-rank approximation reveals structural properties of the implicit reasoning process itself.

3 APPROACH

3.1 THE GEOMETRY OF IMPLICIT REASONING TRAJECTORIES

We perform principal component analysis on activations collected from trained TRMs across all recursive time-steps. The spectral decay is rapid: at the MLP intermediate site (dimension 3072), the top 128 components capture the vast majority of activation variance (Figure 4 in the Appendix). This implies that out of 3072 available dimensions, the model's reasoning trajectory is effectively confined to a subspace roughly $24\times$ smaller.

To rule out the possibility that a nonlinear manifold merely *appears* linear under PCA, we compared linear projection against nonlinear autoencoder baselines. The autoencoders provided no meaningful improvement in reconstruction quality, confirming that the low-rank structure is genuinely linear rather than a curved manifold that happens to be locally flat.

This linearity is not coincidental—it arises from architectural induction biases. Smooth activations (GELU, SiLU) are locally linear over the operating range; LayerNorm constrains activations to a bounded manifold; and critically, recursive weight-sharing forces the same linear operator to be applied repeatedly, concentrating the activation distribution along the dominant eigendirections of the shared weight matrices. The result is that implicit reasoning in TRMs, despite unrolling over many steps, proceeds along a low-dimensional linear corridor.

3.2 ACTIVATION STATISTICS AND QUANTIZATION FEASIBILITY

To evaluate whether TRM activations are suitable for low-precision compression, we analyzed their empirical distribution at both MLP and attention sites. Across 2.1M samples, activations were approximately zero-mean with near-Gaussian value histograms and tightly concentrated per-channel variances. Crucially, we observed no heavy-tailed behavior or frequent outliers.

These properties imply that TRM activations are highly compressible. Simulated INT8 quantization using symmetric scaling introduced negligible distortion (relative MSE $< 0.03\%$), and applying quantization *after* low-rank projection preserved means and standard deviations within 2% across a range of ranks.

Overall, TRM activations inhabit a well-behaved statistical regime—low-rank, nearly isotropic, and free of significant outliers—making them well-suited for LASER's low-rank–plus–quantization design. Full distributional analyses and quantization metrics are provided in Appendix A.3, A.4.

3.3 LASER ALGORITHM

LASER approximates the activation matrix $X \in \mathbb{R}^{B\times D}$ using a low-rank basis $Q \in \mathbb{R}^{D\times k}$, such that $X \approx XQQ^T$. We store the compressed representation $Z = XQ \in \mathbb{R}^{B\times k}$ and the basis Q, reducing memory complexity from $O(BD)$ to $O(Bk + Dk)$.

Algorithm 1 LASER: Low-Rank Activation SVD for Efficient Recursion

Require: Streams $\{X_t^{(s)}\}_{s\in\mathcal{S}}$, rank k_0, threshold ε, patience p, expansion m

Init: $\forall s$: $Q_0^{(s)} \leftarrow$ top-k_0 SVD$(X_0^{(s)})$; $c^{(s)} \leftarrow 0$
for each training step t **do**
  for each site s **do**
    $Z \leftarrow X_t^{(s)} Q_{t-1}^{(s)}$ *// Compress*
    $F \leftarrow \|Z\|_F / \|X_t^{(s)}\|_F$ *// Fidelity*
    if $F \geq \varepsilon$ **then**
      $Q_t^{(s)} \leftarrow \text{Orth}(X_t^{(s)T} Z)$ *// Power iteration*
      $c^{(s)} \leftarrow 0$
    else
      $c^{(s)} \leftarrow c^{(s)} + 1$
      if $c^{(s)} \geq p$ **then**
        $Q_t^{(s)} \leftarrow$ top-k_0 SVD$(X_t^{(s)})$ *// Reset*
        $c^{(s)} \leftarrow 0$
      else
        Sample m rows from $X_t^{(s)}$ as R *// Expand*
        $R_\perp \leftarrow R - R Q_{t-1}^{(s)} Q_{t-1}^{(s)T}$
        $Q_t^{(s)} \leftarrow \text{Orth}([Q_{t-1}^{(s)} | R_\perp^T])$
      end if
    end if
  end for
  Backward: $\forall s$: use $\hat{X}^{(s)} = Z^{(s)} Q_t^{(s)T}$
end for

Instead of computing the expensive covariance matrix $X^T X$ for every batch (an $O(D^2)$ operation), LASER updates the basis Q using a matrix-free power iteration step. The new subspace direction is estimated via $M = X^T(XQ)$. This is computed in $O(BDk)$, significantly faster than full SVD. M is then orthonormalized to produce the updated basis Q_{new}.

A major failure mode of iterative tracking is "subspace drift," where the basis Q slowly diverges from the true principal components during rapid distribution shifts or poor conditioning. LASER mitigates this via a fidelity-based response mechanism.

We define a fidelity metric $F_t = ||Z_t||_F / ||X_t||_F$. If $F_t < \varepsilon$ (a pre-defined threshold), LASER first attempts a subspace expansion. m activation vectors are sampled, orthogonalized against the current basis, and appended to Q, immediately broadening the subspace. If fidelity remains low for p consecutive batches, LASER triggers a "Hard Reset." We compute the exact SVD on the current micro-batch to perfectly realign Q with the current activation variance, but do not backpropagate over that batch to avoid a memory spike. This hybrid approach allows for the speed of power iterations with the stability of exact SVD.

LASER maintains separate bases $Q^{(s)}$ for each compression site s (e.g., MLP intermediate, MLP output, attention output). Each site tracks its own fidelity $F_t^{(s)}$ and adapts rank independently. This allows aggressive compression where possible while preserving capacity where needed (in particular, later-stage activations often require higher k).

LASER's key advantage over prior work is *matrix-free adaptive tracking with graduated fallback*. Unlike methods outlined in Section 2, LASER tracks the evolving subspace via $O(BDk)$ power iteration, only invoking exact SVD when fidelity drops persistently. This yields the efficiency of static methods with the robustness of dynamic ones.

3.4 Memory Scaling with Recursion Depth

LASER is especially well-suited to recursive architectures because its memory advantage grows with recursion depth. For recursion depth n, batch size B, activation width D, and retained rank

$k \ll D$, standard backpropagation stores activations with memory

$$M_{\text{full}}(n) = O(nBD),$$

whereas LASER stores compressed per-step coefficients together with a shared basis:

$$M_{\text{LASER}}(n) = O(nBk + Dk).$$

Thus LASER replaces the dominant per-step memory term in width D with one in width k. As recursion depth increases, the one-time basis cost becomes negligible and the memory savings grow approximately linearly with n, making deeply recursive models a particularly favorable regime for low-rank activation compression.

3.5 Theoretical Guarantees

3.5.1 Fidelity Metric

The fidelity $F_t = \|Z\|_F / \|X\|_F$ exactly equals the cosine similarity between X and its reconstruction $\hat{X} = ZQ^T$ when Q is orthonormal (Proposition 1, Appendix A.1). This enables zero-cost quality monitoring without reconstructing $\hat{X}$.

3.5.2 Gradient Fidelity

Reconstruction accuracy alone is insufficient—we require that gradients computed at $\hat{a} = QQ^T a$ approximate those at a. Under standard smoothness assumptions (Lipschitz Jacobian with constant L_J), we prove:

Theorem 1. *The gradient error satisfies $\|\tilde{g}_a - g_a\| \leq L_J \|\lambda\| \varepsilon$, where $\varepsilon = \|\hat{a} - a\|$ and λ is the upstream gradient.*

Corollary 1. *Gradient cosine similarity satisfies* $\cos(g_a, \tilde{g}_a) \geq 1 - 2L_J \|\lambda\| \varepsilon / \|g_a\|$.

The smoothness assumptions hold because TRM activations traverse a well-conditioned manifold: GELU/SiLU activations are C^∞, LayerNorm bounds activation norms, and recursive weight-sharing concentrates activations along stable eigendirections. Together, these architectural choices ensure small L_J, meaning gradient fidelity degrades gracefully with reconstruction error. Crucially, LASER uses the *full* Jacobian—only the evaluation point shifts. This avoids rank collapse that would occur if gradients flowed through a bottleneck layer. Proofs are in Appendix A.2.

4 Implementation

We implemented LASER in PyTorch using forward/backward hooks, allowing it to be slotted into other models as well, theoretically.

4.1 Experimental Setup

We evaluate LASER on 11×11 maze pathfinding, where a TRM must predict the solution path connecting start to goal. The dataset contains 10,000 procedurally generated mazes (9,000 train / 1,000 validation). Our TRM uses hidden dimension 512, 8 attention heads, SwiGLU MLP with 4× expansion, and 24 recursive L-cycles with gradient. We use 24 recursions to match the standard TRM setting from prior work; this depth is also sufficient to expose the activation-memory bottleneck while remaining feasible within our compute budget. LASER compresses all activations saved for backpropagation. Adaptive rank growth allows smaller tensors to recover full rank when needed, while large tensors (`mlp_3072`, `attn_1536`) remain highly compressed (57–92%), providing the bulk of memory savings. We believe smaller tensors can also be compressed with tuning of LASER's hyperparameters. We compare baseline training against LASER with initial ranks $k \in \{128, 256\}$, reporting token accuracy and maze solve rate over 5 seeds. Full hyperparameters are provided in the Appendix, as well as seeds, for reproducibility.

5 RESULTS

Figure 2 shows validation performance during training. LASER closely tracks the baseline while using substantially less activation memory.

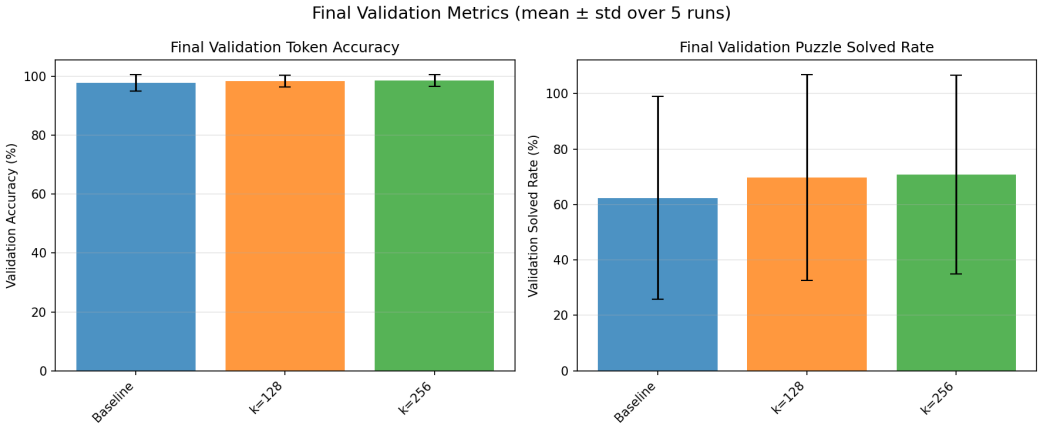

Figure 2: **Validation performance under LASER compression.** Across 24 recursive steps, LASER closely matches baseline validation behavior while reducing activation memory by approximately 60%.

Table 1 summarizes the final performance across five random seeds. LASER achieves **60% activation memory reduction** (2.85 GB $\rightarrow$ 1.14 GB) with no statistically significant performance degradation.

Table 1: Performance (mean $\pm$ std, 5 seeds).

Method	**Val Acc (%)**	**Val Solved (%)**	**Act. Mem.**
Baseline	97.82±2.78	62.34±36.55	2.85 GB
LASER k=256	98.54±2.00	70.74±35.88	1.23 GB
LASER k=128	98.46±2.01	69.71±37.09	1.14 GB

The primary compression target is `mlp_3072`, achieving 92.5% compression at k=128 and accounting for 65% of total memory savings.

6 DISCUSSION

Where does redundancy live in implicit reasoning? The most striking finding is not the aggregate compression ratio but its asymmetry across computational sites. The 3072-dimensional MLP intermediate activations compress to rank 128—a 92.5% reduction—while the 512-dimensional hidden state and smaller attention sites grow back toward full rank under LASER's adaptive mechanism.

Table 2: Per-site compression (final ranks at k=128)

Site	**Dim**	**Final Rank**	**Savings**
mlp_3072	3072	128	92.5%
attn_1536	1536	470	57.3%
mlp_1536	1536	510	53.6%
attn_512	512	510	0% (overhead)
mlp_512	512	500	0% (overhead)
attn_64	64	64	0%

This suggests that redundancy in implicit reasoning is highly localized: the expanded MLP bottleneck is massively over-parameterized relative to what each recursive step actually computes, while the core hidden state genuinely utilizes its full capacity. Weight-sharing appears to concentrate MLP computation along a few dominant eigendirections, but this concentration is site-specific rather than uniform.

Concentrated computation and reasoning efficiency. The rapid spectral decay at high-dimensional sites raises a natural architectural question: if MLP intermediate activations at each recursive step operate in a $\sim$128-dimensional corridor within a 3072-dimensional space, it may be possible to amplify computation along these directions and achieve comparable reasoning with fewer but more effective recursive steps. The high variance in maze solve rates (Table 1) hints that current TRM training may not reliably find these productive directions, and that methods which explicitly encourage signal concentration, whether through architectural changes or training objectives, could improve both efficiency and stability.

Smooth subspace evolution and reasoning complexity. LASER's success depends not only on the subspace being low-rank but on it evolving smoothly enough for power iteration to track during training. The fact that a single power iteration step per batch suffices, with hard resets rarely triggered, implies that the reasoning manifold at compressible sites changes gradually. This is consistent with TRMs learning smooth, stereotyped reasoning trajectories rather than chaotic or rapidly shifting computation patterns. Whether this smoothness is a feature (stable reasoning) or a limitation (inability to perform diverse reasoning strategies) is an open question.

The 11×11 maze pathfinding task provides a highly structured environment, which likely represents a best-case scenario for smooth subspace evolution.

If LASER were applied to more complex, open-ended tasks where implicit reasoning requires diverse and rapidly shifting strategies, the activation subspace may exhibit much higher variance. In volatile tasks, rapid distribution shifts would cause "subspace drift" where our iteratively tracked basis Q diverges from the true principal components. If the activation space is so chaotic that this expansion fails to restore fidelity for p consecutive batches, LASER abandons the power iteration and computes an exact SVD on the current micro-batch to realign the basis. Investigating how subspace dimensionality and drift correlate with task complexity remains a critical next step.

Task dependence and the maze pathfinding regime. The 11×11 maze pathfinding task exhibits several properties that are particularly favorable for low-rank activation compression: a small, fixed vocabulary (5 tokens), deterministic targets (unique shortest paths), a stationary input distribution, and a uniform reasoning strategy (path search applied identically to every input). Together, these properties mean the model converges to a single stable computational routine, and the activation subspace reflects this, settling into a low-dimensional corridor because the same computation is performed on every input with only spatial configuration varying.

On tasks requiring diverse reasoning strategies, such as mathematical problem-solving, natural language inference, or multi-step planning with branching, we would expect the activation subspace to be higher-dimensional (encoding multiple computational pathways), less stable across inputs (different problems activating different directions), and potentially non-stationary across training as the model acquires qualitatively new capabilities. LASER's adaptive mechanisms, namely rank expansion and fidelity-triggered resets, are designed to accommodate such variation, but whether power iteration can track a rapidly shifting subspace without frequent hard resets remains an open empirical question. We note that the overhead of hard resets is bounded (one exact SVD per micro-batch per site), so even in a volatile regime LASER degrades gracefully to periodic exact decomposition rather than failing silently.

Implications for other latent reasoning architectures. Our analysis is specific to TRMs, but the mechanism we believe drives compressibility (repeated application of shared weights concentrating activations along dominant eigendirections) should apply to any looped or weight-tied architecture. We conjecture that similar site-specific low-rank structure exists in other implicit reasoning systems (e.g., Universal Transformers), and that the effective dimensionality of activations at different computational sites could serve as a probe for where redundancy and capacity are allocated in latent reasoning.

Applicability to LoopLM and larger recursive models. LoopLM (Zhu et al., 2025) is a particularly relevant test case, as it applies the same weight-tied recursive principle as TRMs but at significantly larger scale (1.4B–2.6B parameters) and on natural language rather than a structured 5-token vocabulary. The core mechanism we identify, recursive weight-sharing concentrating activations along dominant eigendirections, does not depend on model scale; it depends on the spectral properties of the shared weights. MLP intermediate layers are widely observed to be over-parameterized in standard (non-recursive) transformers (Gromov et al., 2025), and weight-sharing should produce the same concentrating effect regardless of hidden dimension. The compression-relevant quantity is the ratio of effective rank to ambient dimension, which we expect to remain favorable.

However, two factors could reduce compressibility in this setting. First, models trained on natural language exhibit heavier-tailed activation distributions with more frequent outliers than we observe in the maze vocabulary (Section 3.2), which could degrade both low-rank approximation quality and post-projection quantization. Second, richer tasks may require higher effective rank at each computational site, narrowing the gap between k and D and reducing LASER's memory advantage. Validating whether the site-specific compression asymmetry we observe (Table 2), with MLP intermediates highly compressible and hidden states near full rank, persists in LoopLM and similar architectures remains to be tested empirically.

6.1 Comparisons with Alternative Methods in Literature

While computational and time constraints prevent us from benchmarking LASER directly against existing activation compression methods in the TRM setting, we identify principled structural differences that distinguish the recursive regime from the settings these methods were designed for. These differences partially motivated our development of LASER.

ActNN (Chen et al., 2021) achieves up to $12\times$ activation memory reduction through mixed-precision scalar quantization, treating each activation element independently. This approach is **orthogonal** to LASER's subspace compression: ActNN removes bit-width redundancy while LASER removes structural redundancy arising from weight-sharing. Our quantization analysis (Section 3.2, Appendix A.3, A.4) demonstrates that these two compression axes compose well: INT8 quantization applied after low-rank projection introduces negligible additional distortion ($<0.03\%$ relative MSE). We therefore view ActNN-style quantization as complementary to LASER rather than competitive.

LANCE (Apolinario & Roy, 2025) fixes a low-rank basis at initialization via one-shot HOSVD and reuses it throughout training, which is an efficient strategy for fine-tuning pre-trained models where the activation subspace is already near equilibrium. However, for training that proceeds from random initialization, such as our TRM training setup here, the activation subspace at epoch 0 bears little resemblance to the converged reasoning subspace. Our PCA analysis (Figure 4) reveals that even when the SVD basis is *recomputed exactly at every training step*, an oracle upper bound on any fixed-rank projection, lower-rank reconstructions exhibit a pronounced mid-training fidelity dip. Since the basis is optimal at each step, this dip cannot reflect directional misalignment; rather, it indicates that the effective dimensionality of the activation manifold expands as the model transitions between learning regimes. A static basis like LANCE's would face this same dimensionality expansion, which LASER accounts for, *and additionally* suffer from directional staleness: using step-0 principal components that are unlikely to span the relevant subspace at later training stages. This second source of error is invisible in the oracle curves, meaning Figure 4 represents a generous upper bound on LANCE's achievable fidelity. LASER addresses both failure modes: adaptive rank growth accommodates dimensionality expansion, while power iteration tracks directional drift.

LoRAct (Shi et al., 2025) is the most structurally similar method to LASER, performing online low-rank decomposition via sampling-based orthogonal factorization. However, LoRAct was designed for parameter-efficient fine-tuning of large pre-trained models, where each layer has distinct weights producing genuinely different activation distributions that warrant independent decomposition. In a 24-step TRM with shared weights, the same operator is applied at every recursive step, concentrating activations along the same dominant eigendirections. LoRAct would perform 24 independent decompositions per site that each converge to approximately the same subspace, while LASER amortizes a single shared basis Q across all steps, a distinction whose computational advantage grows linearly with recursion depth. Additionally, because LoRAct recomputes the decomposition

from scratch rather than tracking incrementally, it avoids subspace staleness entirely, at the cost of having to recompute fully at every step.

GALE (Muhsin et al., 2026) applies randomized sketching to compress activations, using data-independent random projections that preserve geometric structure. This data-independence means GALE cannot exploit the pronounced low-rank structure that weight-sharing induces: it treats the 128-dimensional effective subspace within a 3072-dimensional MLP identically to a full-rank distribution. However, this same property makes GALE immune to subspace drift by construction. On tasks where the activation manifold shifts rapidly and LASER's iterative tracking would require frequent hard resets, GALE's fixed random projections would maintain consistent compression quality. The per-step overhead of sketching is higher than LASER's power iteration and scales linearly with recursion depth, but this cost may be justified in volatile regimes. Investigating the tradeoff between structure-exploiting methods like LASER and structure-agnostic methods like GALE across tasks of varying complexity is a promising direction for future work.

7 Conclusion

We introduced LASER, a dynamic compression framework for training recursive models, and used it to reveal that TRM reasoning trajectories occupy a low-dimensional linear subspace that evolves smoothly during training. LASER achieves over 60% activation memory savings with accuracy statistically indistinguishable from baseline, supported by theoretical gradient fidelity guarantees.

Our analysis raises open questions about the effective reasoning depth of recursive architectures and the relationship between subspace dimensionality and reasoning complexity. Limitations include evaluation on a single task (maze pathfinding) and the need for manual tuning of adaptive rank growth. Future work includes extending this analysis to other looped architectures, investigating whether subspace dimensionality correlates with task difficulty, and combining LASER with gradient checkpointing for further memory reduction.

Author Contributions

Ege Çakar: LASER theory development, LASER implementation and updates, activation space linearity and autoencoder ablations, LASER ablations, additional ablations, literature comparison, transfer across architectures analysis, experimental debugging, and final manuscript formatting.

Lia Zheng: Quantization analysis (including outlier characterization), figure editing, theoretical scaling law edits, additional ablations, and experimental debugging.

Ketan Raghu: Activation quantization ablations, exploratory ablations on combined LASER and quantization, additional ablations, and experimental debugging.

REFERENCES

Marco Paul E. Apolinario and Kaushik Roy. Lance: Low rank activation compression for efficient on-device continual learning, 2025. URL https://arxiv.org/abs/2509.21617.

Jianfei Chen, Lianmin Zheng, Zhewei Yao, Dequan Wang, Ion Stoica, Michael W. Mahoney, and Joseph E. Gonzalez. Actnn: Reducing training memory footprint via 2-bit activation compressed training, 2021. URL https://arxiv.org/abs/2104.14129.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers, 2025. URL https://arxiv.org/abs/2403.17887.

Animesh Jain, Amar Phanishayee, Jason Mars, Lingjia Tang, and Gennady Pekhimenko. Gist: Efficient data encoding for deep neural network training. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pp. 776–789, 2018. doi: 10.1109/ISCA.2018.00070.

Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks, 2025. URL https://arxiv.org/abs/2510.04871.

Sayed Muhsin, Hao Zhang, and Seokbum Ko. Gale: Gradient activation low-rank extraction for fast memory efficient large language model training, 2026. URL https://openreview.net/forum?id=D9Oq3c5iHn.

Jiang-Xin Shi, Wen-Da Wei, Jin-Fei Qi, Xuanyu Chen, Tong Wei, and Yu-Feng Li. Memory-efficient fine-tuning via low-rank activation compression, 2025. URL https://arxiv.org/abs/2509.23472.

Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995. doi: 10.1109/78.365290.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, Ziniu Li, Haoran Que, Boyi Wei, Zixin Wen, Fan Yin, He Xing, Lu Li, Jiajun Shi, Kaijing Ma, Shanda Li, Taylor Kergan, Andrew Smith, Xingwei Qu, Mude Hui, Bohong Wu, Qiyang Min, Hongzhi Huang, Xun Zhou, Wei Ye, Jiaheng Liu, Jian Yang, Yunfeng Shi, Chenghua Lin, Enduo Zhao, Tianle Cai, Ge Zhang, Wenhao Huang, Yoshua Bengio, and Jason Eshraghian. Scaling latent reasoning via looped language models, 2025. URL https://arxiv.org/abs/2510.25741.

A APPENDIX

A.1 FIDELITY METRIC JUSTIFICATION

We formally demonstrate that F_t is exactly equivalent to the cosine similarity between the original activations and their low-rank reconstruction, provided the basis is orthonormal.

Proposition 1. *Let $X \in \mathbb{R}^{N\times D}$ be the activation matrix and $Q \in \mathbb{R}^{D\times k}$ be a semi-orthogonal basis such that $Q^TQ = I_k$. Let $Z = XQ$ be the compressed state and $\hat{X} = ZQ^T$ be the reconstruction. Then, the fidelity ratio equals the cosine similarity: $\frac{\|Z\|_F}{\|X\|_F} = sim(X, \hat{X})$.*

Proof. Recall the definition of cosine similarity for matrices: $\text{sim}(X, \hat{X}) = \frac{\langle X, \hat{X}\rangle_F}{\|X\|_F\|\hat{X}\|_F}$. First, we simplify the inner product (numerator) using the cyclic property of the trace:

$$\begin{aligned}\langle X, \hat{X}\rangle_F &= \text{Tr}(X^T(XQQ^T)) = \text{Tr}(Q^TX^TXQ) \\ &= \text{Tr}((XQ)^T(XQ)) = \|Z\|_F^2\end{aligned} \tag{1}$$

Next, we determine the norm of the reconstruction (denominator). Since Q is orthonormal, the projection preserves the norm of the coefficients Z:

$$\begin{aligned}\|\hat{X}\|_F^2 &= \text{Tr}((ZQ^T)^T(ZQ^T)) = \text{Tr}(QZ^TZQ^T) \\ &= \text{Tr}(Z^TZ(Q^TQ)) = \|Z\|_F^2 \implies \|\hat{X}\|_F = \|Z\|_F\end{aligned} \tag{2}$$

Substituting these results back into the similarity definition:

$$\text{sim}(X, \hat{X}) = \frac{\|Z\|_F^2}{\|X\|_F\|Z\|_F} = \frac{\|Z\|_F}{\|X\|_F} \equiv F_t \quad \blacksquare \tag{3}$$

This equivalence ensures that monitoring F_t provides a mathematically rigorous measure of angular alignment without requiring the expensive $O(ND)$ reconstruction of $\hat{X}$.

A.2 GRADIENT FIDELITY GUARANTEES

Reconstruction accuracy is necessary but not sufficient: we need gradients computed at reconstructed activations to approximate true gradients.

A.2.1 SETUP

Let $a \in \mathbb{R}^D$ be the activation at a compression site, and let $f(a, \theta) : \mathbb{R}^D \to \mathbb{R}^m$ represent the downstream computation. Let $\hat{a} = QQ^Ta$ be the reconstructed activation with error $\varepsilon = \|\hat{a} - a\|$. The true gradient is $g_a = J_af(a, \theta)^T\lambda$ where $\lambda = \nabla\mathcal{L}$ is the upstream gradient; LASER computes $\tilde{g}_a = J_af(\hat{a}, \theta)^T\lambda$.

A.2.2 ASSUMPTIONS

We require smoothness conditions satisfied by networks with GELU/SiLU activations and LayerNorm:

1. **Lipschitz Jacobian**: $\|J_af(a') - J_af(a)\| \le L_J\|a' - a\|$
2. **Bounded Jacobian**: $\|J_af(a)\| \le M_a$

A.2.3 MAIN RESULT

Theorem 1 (Gradient Error Bound). *Under the Lipschitz Jacobian assumption:*

$$\|\tilde{g}_a - g_a\| \le L_J\|\lambda\|\varepsilon \tag{4}$$

Proof. $\|\tilde{g}_a - g_a\| = \|(J_af(\hat{a}) - J_af(a))^T\lambda\| \le \|J_af(\hat{a}) - J_af(a)\|\|\lambda\| \le L_J\varepsilon\|\lambda\|$ $\blacksquare$

Corollary 1 (Cosine Alignment). *The gradient cosine similarity satisfies:*

$$\cos(g_a, \tilde{g}_a) \ge 1 - \frac{2L_J\|\lambda\|\varepsilon}{\|g_a\|} \tag{5}$$

A.2.4 INTERPRETATION

This result is crucial: LASER uses the *full* Jacobian $J_a f$, only the evaluation point shifts from a to $\hat{a}$. This avoids the rank collapse that would occur if gradients flowed through a k-dimensional bottleneck. Combined with the Eckart-Young-Mirsky theorem guaranteeing SVD optimality, this explains why LASER outperforms autoencoders despite their greater representational capacity.

A.3 ACTIVATION STATISTICS AND QUANTIZATION DETAILS

We characterized activation distributions at the MLP and attention compression sites using 2.1M activation samples collected over 425 batches. Both sites exhibited approximately Gaussian value histograms with near-zero means and per-dimension variances of 0.91 (MLP) and 1.46 (attention). Per-channel standard deviations were tightly concentrated in the ranges $[0.7, 1.1]$ for MLP and $[0.9, 1.4]$ for attention.

Outlier rates closely matched Gaussian predictions. For MLP activations we observed

$$\mathbb{P}(|x - \mu| > 2\sigma) = 4.1 \times 10^{-2}, \quad \mathbb{P}(|x - \mu| > 3\sigma) = 3.0 \times 10^{-3},$$

with only 1.4×10^{-4} and 6.7×10^{-6} samples exceeding 4σ and 5σ, respectively. Attention activations displayed similar behavior.

We simulated INT8 quantization using (i) max-based scaling and (ii) 4σ-based scaling. For both MLP and attention sites, max-based scaling yielded a relative MSE of approximately 1.4×10^{-4}, while 4σ scaling produced at most 2.9×10^{-4}. Thus, quantization noise accounts for less than 0.03% of activation energy.

Finally, applying INT8 quantization after PCA projection at ranks $k \in \{64, 128, 256\}$ produced reconstructed activation statistics nearly identical to the raw distributions: means and standard deviations changed by less than 2%, and overlaid histograms were visually indistinguishable.

A.4 QUANTIZATION ERROR

We evaluate post-hoc INT8 activation quantization at the MLP and attention sites using the two scaling schemes from Section 3.2. The resulting relative MSE is extremely small: $\approx 1.4 \times 10^{-4}$ with max-based scaling and at most 2.9×10^{-4} with 4σ scaling for both sites (Figure 3). Applying quantization after PCA projection with ranks $k \in \{64, 128, 256\}$ yields nearly identical marginal statistics to the raw activations (means and standard deviations change by $< 2\%$), and overlaid histograms are visually indistinguishable. Thus, in the TRM regime we study, low-rank projection plus INT8 quantization introduces only a tiny perturbation, supporting the use of LASER as a low-rank+quantization scheme for activation memory reduction.

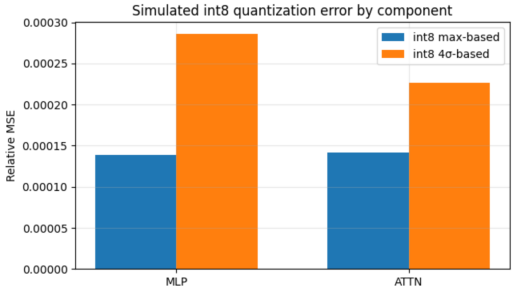

Figure 3: Simulated error by component

Table 3: Model configuration

Hidden size	512
Attention heads	8 (head dim 64)
MLP expansion	4.0× (SwiGLU, inter=1536)
H-cycles / L-cycles	1 / 24
Position encoding	RoPE (θ=10000)
Parameters	$\sim$3.4M

Table 4: Training configuration

Optimizer	AdamW (β=(0.9, 0.95), wd=10^{-2})
Learning rate	10^{-4}, cosine decay
Batch size	64
Epochs	16
Gradient clip	1.0
Precision	bfloat16 (AMP)
Seeds	100–104 (5 runs)

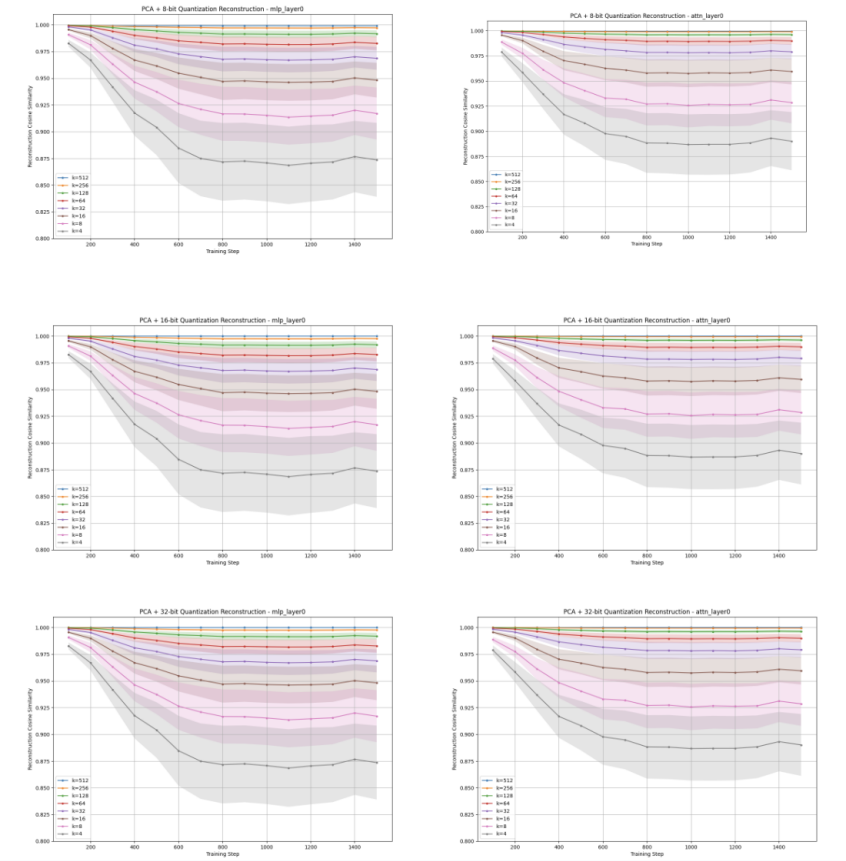

Figure 4: PCA reconstruction cosine similarity across training steps at the MLP (left) and attention (right) sites under 8-bit (top), 16-bit (middle), and 32-bit (bottom) quantization, with the SVD basis recomputed exactly each batch. Rank k=128 maintains >0.975 fidelity for the MLP activations throughout training, confirming rapid spectral decay. Quantization precision has minimal effect on reconstruction quality. The mid-training fidelity dip reflects the evolving activation subspace as the model learns, motivating LASER's adaptive tracking over static bases.

Table 5: LASER configuration

Initial rank (k)	128 or 256
Fidelity threshold (ε)	0.95
Growth size (m)	4
Max rank	512
Power iteration steps	1
Reset patience (p)	2

Table 6: Dataset: 11×11 maze pathfinding

Samples	10,000 (9k train / 1k val)
Grid size	11×11 (seq_len=121)
Vocabulary	5 tokens (wall, passage, start, goal, path)
Task	Predict shortest path from start to goal