The Gradient Does Not See Rank: Rank-Indifference in Matrix-CODI on ProsQA

Sam Larson
Pebble AI
sam@pebbleml.com

Abstract

Continuous chain-of-thought models compress reasoning into latent tokens. Matrix-valued variants introduce a new structural observable — the rank of the latent matrix Z. If matrix latents carry parallel reasoning paths via superposition, rank should track them, and truncating Z to low rank should hurt accuracy on tasks whose solutions plausibly require multiple components. Across four training regimes of a matrix-CODI model (three on ProsQA, one on GSM8K-Aug at a below-threshold operating point that we flag), the rank-k projection ablation curve is flat to within 0.6 percentage points. A three-seed replication yields accuracy at 81.5 ± 1.2pp while the final effective rank of Z spans $\{4, 12, 13\}$; the loss does not reward any particular rank. To test whether rank-blindness arises from the flatten-then-project readout alone, four readout variants are constructed: a bilinear reparametrization, a bilinear-plus-GELU readout explicitly nonlinear in Z, an SVD-augmented readout feeding singular values through an MLP, and a quadratic readout in $ZZ^\top$. All four rank-k curves remain flat (Spearman p-values $0.63, 0.14, 0.82, 0.46$). The readout Jacobian is not the sole culprit: even readouts nonlinear in Z do not produce rank-dependent curves, indicating the simple Jacobian-linearity mechanism is incomplete. We defer a refined mechanism (an effectively rank-1 active subspace of the trained readout's Jacobian, independent of its in-principle expressiveness) to future work. A linear probe on Z underperforms a raw pretrained hidden state at target prediction (AUC 0.673 vs 0.846). A negative control on vanilla GPT-2 SFT (no matrix bottleneck, no Z, three seeds, $n = 500$) reproduces a flat rank-k curve under the same intervention paradigm with pooled-mean range 0.20pp, and a random-h sensitivity floor lands at the same accuracy — the rank-k ablation, by itself, conflates rank-blindness with position-irrelevance. Rank-k ablation should not be used as a probe for superposition in CODI-trained matrix latents.

1 Introduction

Continuous chain-of-thought (CoT) models replace explicit textual reasoning steps with continuous latent tokens fed back into the transformer's residual stream. COCONUT [11] and CODI [20] are representative instances: both compress an explicit rationale into a small number of continuous latent positions and decode the answer from the resulting state. Theoretical work [8, 22] argues that these latents can hold multiple reasoning paths in *superposition*, so continuous CoT could explore a search tree in parallel.

Recent empirical work has pushed back. Rizvi-Martel et al. [17] showed that a fine-tuned COCONUT model reaches 96.6% on ProsQA *without* feeding back any latent tokens, against 99.0% with latents and 85.3% for explicit CoT. In their fine-tuned setting, the latent machinery is not doing work. They named this the *Illusion of Superposition*.

Matrix-valued latents [e.g., 20] sharpen the superposition hypothesis by introducing a single-sample structural observable: rank. A $d \times d$ thought Z has a computable rank via its SVD. If each singular direction encodes a separate reasoning path, truncating Z to rank k at inference should degrade accuracy when the task needs more than k paths. The rank-k ablation curve is the natural probe.

Contributions. We report four pieces of evidence that this probe is uninformative in practice and give a mechanism that explains why.

1. **Three flat rank-k curves** from the flatten-then-project readout on ProsQA across two distillation weights ($\gamma \in \{0, 1\}$) and a multiplicative-thinker on/off ablation (§3). Range across $k \in \{1, 2, 4, 8, 16\}$ is $\leq$ 0.6pp. A fourth flat curve on GSM8K-Aug is reported at a below-threshold (6%) operating point and flagged as non-interpretable on its own.
2. **Three-seed accuracy-rank decoupling** (§3). Best ProsQA accuracy is 81.51 ± 1.2pp while the final effective rank of Z spans $\{4, 12, 13\}$. The loss does not reward any particular rank.
3. **Four nonlinear-in-Z positive-control readouts also produce flat curves** (§5). A bilinear reparametrization, a bilinear-plus-GELU readout, an SVD-augmented readout feeding singular values through an MLP, and a quadratic readout in $ZZ^\top$ all yield flat rank-k curves (Spearman p-values $0.63, 0.14, 0.82, 0.46$). Readout linearity is a sufficient but not necessary condition: the matrix-bottleneck training objective itself produces rank-indifferent gradients through the full chain rule.
4. **A linear probe** on the matrix thought Z concatenated

across six positions (1536 features) reaches AUC 0.673 at predicting the ProsQA target class, below a raw pretrained GPT-2 hidden state at 768 features (AUC 0.846, §3). The matrix bottleneck does not deliver target-predictive information commensurate with its feature dim.

5. **A negative control on vanilla GPT-2 SFT** (§5.4). Running the rank-k ablation on a model with no matrix bottleneck and no Z produces a flat curve at the same accuracy, and a random-h sensitivity floor lands at the same accuracy. The rank-k probe alone conflates rank-blindness with position-irrelevance; the matrix-CODI claim rests on the model-level seed-decoupling result, not on the probe's flat shape in isolation.

Two February 2026 papers measure rank in *linear-attention fast-weight hidden states* [3, 15] and describe rank as a structural invariant of trained states. The contribution here is a *mechanism* claim about the training objective, and the four positive-control readouts are designed to falsify that claim (§6).

Collectively, these results indicate that rank-k ablation is not a valid measure of whether a matrix-valued latent holds multiple reasoning paths under a CODI-style objective, and that the observed rank of Z is, in our setting, vestigial.

2 Background

CODI distillation. CODI [20] trains a student to compress an explicit chain-of-thought into a fixed number of continuous latent positions by matching a hidden-state target from a teacher pass. The teacher pass consumes prompt + CoT + answer and produces a reference hidden state at a designated colon-token position. The student pass consumes prompt + n latent positions + answer, where each latent position is produced by feeding the previous step's hidden state back as the next input embedding. A hidden-state L1 loss (the *distillation* loss) aligns the student's state at the answer colon to the teacher's, and a standard next-token cross-entropy loss trains the answer prediction. The total loss is $\mathcal{L} = \gamma \mathcal{L}_{\text{kd}} + \mathcal{L}_{\text{ce}}$.

Why matrix latents. Vector representations are flat: a hidden state of dimension D has D scalar entries with no native notion of how many independent components it superposes. Matrix-valued representations have an additional structural observable: rank, the number of independent directions used. Matrix-valued working memory has a long lineage in neural networks — fast-weight networks [18, 19], linear attention [12], the xLSTM family [5], and the Mamba/SSM family [9] all use matrix-shaped state — and recent work measures the rank of these states as a diagnostic of how the model uses its memory [3, 15]. In all of those, the matrix lives *inside* a layer (an attention head's accumulator, an SSM's recurrent state) and serves as recurrent memory. Matrix-CODI extends the lineage to a different placement: the matrix sits on the explicit chain-of-thought feedback path, where each latent reasoning position is a $d \times d$ matrix rather than a vector. The rank of that matrix is a candidate measure of how many reasoning paths the model holds in superposition at that step. The hypothesis we test is whether rank, so defined, behaves as that measure under CODI-style training.

Matrix bottleneck. We extend CODI with a *matrix bottleneck* on the latent feedback path. Given the previous latent hidden state $h \in \mathbb{R}^D$ (here $D = 768$ for GPT-2 small):

$$\begin{aligned} \text{flat} &= W_{\text{up}}\, h, \qquad W_{\text{up}} \in \mathbb{R}^{d^2 \times D} \\ Z &= \text{reshape}(\text{flat};\, d, d) \in \mathbb{R}^{d \times d} \\ Z &\leftarrow (I + \Delta(Z))\, Z\, (I + \Gamma(Z)) \quad \text{(optional thinker)} \\ h_{\text{out}} &= \text{LayerNorm}(\phi(Z)), \end{aligned}$$

where $\phi : \mathbb{R}^{d \times d} \to \mathbb{R}^D$ is the *readout*. The default readout is $\phi(Z) = W_{\text{down}}\, \text{vec}(Z)$, which we call the *flatten-then-project* readout. $W_{\text{down}} \in \mathbb{R}^{D \times d^2}$. We use $d = 16$ throughout.

Rank-k ablation. At inference, compute the SVD $Z = U \Sigma V^\top$ and replace Z by its rank-k truncation $Z_k = U_{:,:k}\, \Sigma_{:k,:k}\, V^\top_{:,:k}$ before the readout. If rank is functional, accuracy should drop as k decreases to 1. If rank is vestigial, the curve is flat.

Effective rank. Because Z is a dense matrix of a trained model, its numerical rank is always d. We report the *effective rank* $\exp\left(-\sum_i \tilde{\sigma}_i \log \tilde{\sigma}_i\right)$ with $\tilde{\sigma}_i = \sigma_i / \sum_j \sigma_j$, a smooth proxy used in the spectral analysis literature. The ablation itself uses the hard top-k truncation.

ProsQA. ProsQA [11] is a synthetic entailment task: a diamond-shaped directed acyclic graph of entities and a question *which leaf entity has property P?*. Each problem has a unique positive answer and a single distractor. We use the training split from `facebookresearch/coconut`. The test set has 128 problems, matching the evaluation protocol in the original CODI release.

Rank is not numerical rank. Throughout, "rank" refers to *effective rank* (for training-curve reporting) and *hard SVD truncation rank* (for the rank-k ablation). The dense-matrix numerical rank of Z is always d and does not change during training; the question is whether the structural capacity offered by rank > 1 is functionally used by the model.

3 The Flatten-Then-Project Readout is Rank-Blind

This section establishes the empirical phenomenon and its simplest mechanism. §5 shows that the mechanism is not the

Run	Task	γ	Thinker	Z rank	k=1	k=16	r_s
R1	GSM8K-Aug	1.0	on	5.5	6.00%	6.12%	-0.023
R2	ProsQA	1.0	on	10.2	78.4%	78.4%	$+0.026$
R3a	ProsQA	0.0	on	12.7	76.8%	76.6%	-0.105
R3b	ProsQA	0.0	off	12.8	72.6%	72.4%	$+0.095$

Table 1: Rank-k projection ablation across four training conditions. Z rank is the mean effective rank of the 16×16 matrix thought at eval time. r_s is the Spearman rank correlation between per-sample effective rank and correctness. Range across $k \in \{1, 2, 4, 8, 16\}$ is ≤ 0.6pp in every row. Signs of r_s are inconsistent across rows; $|r_s| < 0.11$ in all rows. Row R1 (GSM8K-Aug) is at a 6% operating point where the model is below any interpretable learning threshold; it is included for completeness but does not constitute independent evidence.

whole story: readouts that escape it also produce flat curves.

3.1 Four flat rank-k curves

We ran the matrix bottleneck under four training conditions that vary the task, the CODI distillation weight γ, and the multiplicative thinker. Each run produced a single rank-k ablation curve.

The result does not move when varying the task (arithmetic vs logical reasoning), the distillation weight (removing the L1-at-colon loss *raises* effective rank from ~ 10 to ~ 13, but does not change the curve shape), or the multiplicative thinker (turning it off drops accuracy by approximately one point but leaves the curve flat). A negative control in §5.4 shows that this same rank-k probe applied to a vanilla GPT-2 SFT model — one with no Z at all — also produces a flat curve and a random-h sensitivity floor at the same accuracy. The flat curves in this section are therefore necessary but not sufficient evidence on their own; the model-level distinguisher is the three-seed decoupling result (Fig. 1) below, which the negative control cannot reproduce by construction.

3.2 Why: the readout Jacobian is constant in Z

The flatten-then-project readout is $\phi(Z) = W_{\text{down}} \operatorname{vec}(Z)$. Its Jacobian with respect to Z is

$$\frac{\partial \phi(Z)}{\partial Z_{ij}} = W_{\text{down}}[:, ij],$$

which *does not depend on* Z. The loss gradient $\partial \mathcal{L}/\partial Z = (\partial \mathcal{L}/\partial \phi) \cdot (\partial \phi/\partial Z)$ is therefore a vector contracted with a constant tensor. The contraction preserves the row-space of W_{down} but carries no information about the singular structure of Z.

Proposition 1 (Constant-Jacobian rank-blindness)**.** *Let $\phi : \mathbb{R}^{d \times d} \to \mathbb{R}^D$ be linear in Z. Then for any loss $\mathcal{L}$ differentiable at $\phi(Z)$, the gradient $\partial \mathcal{L}/\partial Z$ is an affine function of $\partial \mathcal{L}/\partial \phi$ alone, with coefficients independent of Z. The* loss gradient through ϕ *carries no term that depends on the SVD basis of Z. It follows that the objective $\mathcal{L}$ itself provides no gradient signal preferring one rank of Z over another. This is a local-gradient statement about $\mathcal{L}$; implicit bias from the optimizer (Adam + weight decay) and upstream parameter regularization may still shape the trained rank through channels outside $\mathcal{L}$ (see §6 on implicit low-rank bias).*

Proof sketch. If $\phi(Z) = W \operatorname{vec}(Z)$ with constant W, then $\partial \phi/\partial Z_{ij} = W[:, ij]$ is a constant vector. By the chain rule, $\partial \mathcal{L}/\partial Z_{ij}$ is the inner product of $W[:, ij]$ with $\partial \mathcal{L}/\partial \phi$, which is independent of Z's SVD. Any invariance of $\mathcal{L}$ under a change of Z that preserves the image $\phi(Z)$ is therefore preserved by gradient descent. □

Proposition 1 is a direct consequence of the chain rule applied to a linear map; it is not a deep theorem. Its value is predictive: every linear-in-Z readout should produce a flat rank-k curve under pure loss-gradient training. The four training conditions in Table 1 confirm this prediction; §5 tests what happens when the sufficient condition is violated.

The same argument applies to any readout that factors through a fixed-size linearly-derived space — a linear low-rank factorization of W_{down}, or a per-vocab bilinear logit head. All of these have constant Jacobians and therefore produce rank-blind gradients *through the readout*.

3.3 Accuracy is decoupled from rank across seeds

Proposition 1 predicts that the loss landscape should be flat along rank-changing directions of Z: a model ending training at rank 12 and a model ending at rank 4 can achieve the same accuracy. We test this directly with three training seeds of the same flatten-readout configuration (gpt2-small, ProsQA, $\gamma = 0$, 25 epochs, batch 16).

Figure 1 directly demonstrates that the matrix-bottleneck training objective does not reward rank: three models at identical hyperparameters achieve statistically indistinguishable accuracy at effective ranks spanning a $3\times$ range. The result is not an ablation artifact: the accuracies are computed on the same test split under standard decoding.

3.4 Linear probe on Z

If rank is decoration, what *does* Z carry? We ran a 5-fold cross-validated multi-class logistic regression to predict the ProsQA target class from a flattened Z on 500 held-out test problems. Controls: the same prompt's hidden state from a pretrained GPT-2 with no ProsQA fine-tuning, and the same model at random initialization.

Vanilla pretrained GPT-2 reaches AUC 0.846 at 768 features. The matrix-CODI bottleneck's concatenated matrix thought (6 latent positions $\times$ 256 features $=$ 1536 features) reaches AUC 0.673 — *more* features than the vanilla

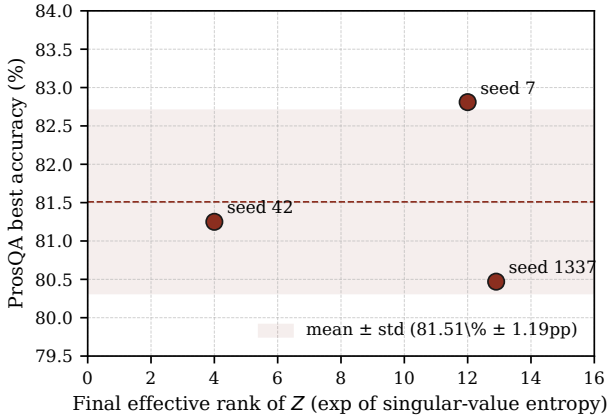

Figure 1: Three-seed replication of the flatten readout. Best ProsQA accuracy is tight at 81.51 ± 1.2pp, but the final effective rank of Z varies $3\times$ (seed 42 converges at rank ~ 4; seeds 1337 and 7 converge near rank 12). The loss does not push Z toward any particular rank.

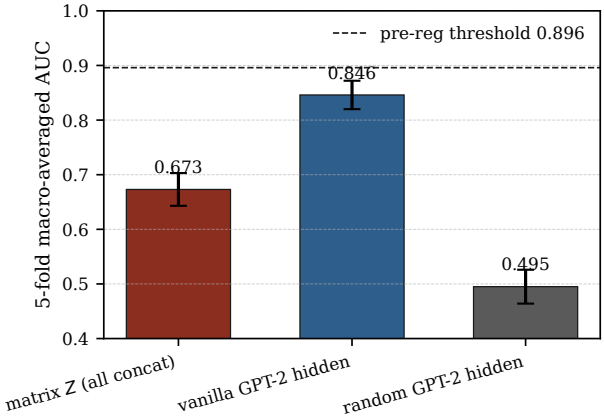

Figure 2: Linear probe AUC for ProsQA target class prediction. Pre-registered threshold for a positive result was $\max(\text{vanilla}, \text{random}) + 0.05 = 0.896$. The matrix Z concat AUC of 0.673 does not exceed it. Vanilla GPT-2 — never trained on ProsQA — predicts the target class better than the trained matrix-CODI bottleneck.

hidden state, *lower* predictive signal for the ProsQA target. A dimension-matched comparison (a probe on the post-bottleneck reconstructed 768-dim hidden state $W_{\text{down}}\,\text{vec}(Z)$ that the downstream transformer consumes) is deferred to camera-ready; the current data establish that the matrix bottleneck does not recover target-predictive information commensurate with its feature count. A binary target-vs-distractor probe on the same Z tensors is at chance (AUC 0.50–0.56) across all conditions.

In combination, the flat rank-k curves, the three-seed decoupling, and the linear probe gap point to the same conclusion: the matrix thought in a flatten-readout matrix-CODI does not construct reasoning state. The structural capacity of Z is available but not used.

4 Depth and Scale Do Not Rescue Matrix-CODI

The results in §3 and §5 may be specific to $d = 16$, GPT-2 small, and six latent positions. We test two axes: depth (number of iterative latent refinement steps) and backbone scale.

4.1 Depth sweep

Depth sweep (preliminary). At $n = 6$ latent refinement steps, vanilla CODI reaches 78.91% on ProsQA, $\sim$2.9pp below pure SFT. The $n \in \{16, 32, 64\}$ configurations exceeded memory at the default batch sizes; re-runs at smaller batches are in progress. A single non-baseline data point is not a depth sweep; we report the $n = 6$ number for completeness but do not draw a trend and defer the full depth sweep to the camera-ready.

4.2 Scale sweep

We trained vanilla SFT and matrix-CODI ($d = 16$, six latent positions, $\gamma = 0$) on ProsQA at three backbone sizes: GPT-2 small (124M), GPT-2 medium (355M), and GPT-2 large (774M). Matrix-CODI at GPT-2 large exceeded memory at batches of 2 and 4; it is omitted from Fig. 3.

Three observations from Fig. 3:

- **Matrix-CODI does not exceed vanilla SFT at any tested scale.** Gap is -1.3pp at GPT-2 small and -0.78pp at GPT-2 medium, both within the three-seed standard deviation of ± 1.2pp measured at GPT-2 small (§3). The consistent sign across two scales is suggestive but not conclusive. Matrix-CODI at GPT-2 large is pending.
- **Vanilla SFT itself degrades with scale on ProsQA.** The ProsQA training set (17,886 examples) is small relative to GPT-2 large's capacity; default AdamW at lr $= 10^{-4}$ likely under-optimizes the larger backbone. This is a data-size artefact of ProsQA, not a finding about superposition.
- **Matrix does not rescue the regression.** If matrix-CODI's inductive bias were relevant at scale, we would expect it to preserve or improve on its gpt2-small performance as capacity grows. It does not.

This sweep establishes that matrix-CODI does not exceed a matched vanilla SFT baseline at the tested scales and training configurations. It does not rule out settings in which matrix bottlenecks help at larger scale under different training regimes.

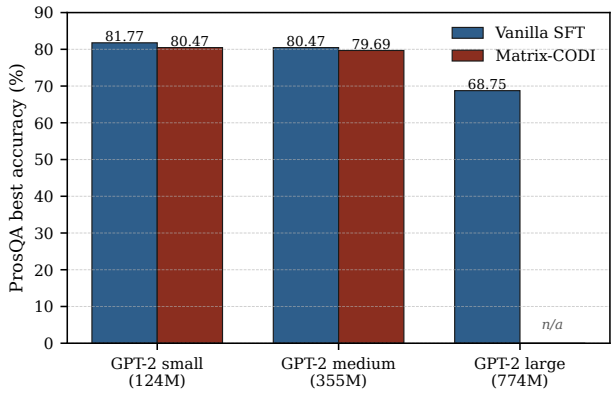

Figure 3: Scale sweep on ProsQA. Vanilla SFT degrades at GPT-2 large (68.75%) compared to GPT-2 small (81.77%, three-seed mean) — ProsQA (17,886 training examples) likely under-optimizes the larger backbone at default AdamW lr $= 10^{-4}$. Matrix-CODI's best accuracy is below its matched vanilla SFT baseline at both tested scales; gaps are within three-seed standard deviation. GPT-2 large matrix-CODI is pending.

Sample efficiency. A complementary sample-efficiency sweep (200, 500, 2000, 5000, 17,886 training examples) found that matrix-CODI is *strictly worse* than vanilla SFT at every N below 17,886, with the gap growing as N decreases (Matrix at $N = 200$: 12.6%; vanilla at $N = 200$: 26.0%). The matrix bottleneck is not a useful inductive bias at any data budget we tested. Full sample-efficiency table is deferred to camera-ready.

5 Positive Control: Nonlinear Readouts Also Produce Flat Curves

Proposition 1 gives a sufficient condition for a readout's Jacobian with respect to Z to be constant. The natural prediction is that a readout violating this condition — a readout *nonlinear* in Z — should produce a non-flat rank-k curve. This section runs that positive control. The hypothesis — that readout linearity is the full mechanism — is falsified.

5.1 Four readout variants

We constructed four readout variants, replacing only $\phi : \mathbb{R}^{d\times d} \to \mathbb{R}^D$ in the matrix bottleneck and leaving all other training hyperparameters fixed (gpt2-small, ProsQA, $\gamma = 0$, six latent positions, $d = 16$, 25 epochs, batch 16, seed 1337, AdamW at lr $= 10^{-4}$). Note that all four positive controls are trained at $\gamma = 0$ — that is, without the CODI L1-at-colon distillation term. The flat rank-k curves therefore hold for the cross-entropy loss through the matrix bottleneck alone, not for the CODI distillation loss specifically. The variants are:

Bilinear. $\phi(Z) = W\,\mathrm{probes}(Z)$ where $\mathrm{probes}(Z)_k = u_k^\top Z v_k$ for $K = d^2$ learned probe pairs $(u_k, v_k) \in \mathbb{R}^d \times \mathbb{R}^d$. Each probe is a Frobenius inner product $\langle u_k v_k^\top, Z\rangle_F$, which is linear in Z. This variant is a reparametrization control — it verifies that the low-rank factoring of W_{down} does not change the curve.

Bilinear+GELU. $\phi(Z) = W\,\mathrm{GELU}(\mathrm{probes}(Z))$. The GELU gates each probe's contribution by a scalar that depends on Z, making ϕ nonlinear in Z. The Jacobian is not constant, but its column space in $\mathbb{R}^{d\times d}$ is fixed (the span of the $u_k v_k^\top$); only column magnitudes vary with Z. This is a relatively mild violation of Proposition 1; SVD-augmented (below) tests a stronger violation in which the Jacobian's column space depends on Z's singular structure.

SVD-augmented. $\phi(Z) = W_{\text{down}}\,\mathrm{vec}(Z) + \mathrm{MLP}(\sigma(Z))$, where $\sigma(Z) \in \mathbb{R}^d$ is the vector of singular values. The MLP feeds the $d = 16$ singular values through two dense layers with GELU. We compute $\sigma(Z)$ via `torch.linalg.svdvals`, whose backward is documented as unconditionally numerically stable (in contrast to full `torch.linalg.svd`, whose backward is unstable at near-coincident singular values). This variant explicitly exposes rank to the optimizer. Empirically, SVD-augmented's best accuracy (78.12%) is below the flatten baseline (80.47%), inconsistent with the failure mode of the optimizer zeroing the `sigma_proj` branch and falling back to flatten alone.

Quadratic. $\phi(Z) = W_{\text{down}}\,\mathrm{vec}\,\mathrm{concat}(ZZ^\top, Z^\top Z)$. The readout is quadratic in Z (and linear in the second-moment tensor). This is the second-moment variant — the gradient with respect to Z depends on Z itself.

Readout	k=1	k=2	k=4	k=8	k=16	r_s	p
flatten	79.0	79.0	79.0	79.0	79.0	~ 0	—
bilinear	78.1	78.9	78.9	78.1	78.1	$+0.04$	0.63
bilinear+GELU	78.9	79.7	79.7	79.7	79.7	-0.13	0.14
svd-aug	77.3	78.1	78.1	77.3	78.1	$+0.02$	0.82
quadratic	79.7	79.7	79.7	79.7	79.7	$+0.07$	0.46

Table 2: Rank-k ablation accuracies (%) by readout on 128 ProsQA test problems. Spearman r_s of per-sample effective rank against correctness; p is two-sided. No variant's r_s is statistically significant.

Each of Bilinear+GELU, SVD-augmented, and Quadratic violates the sufficient condition of Proposition 1. The hypothesis under test is: if readout linearity is the mechanism behind the flat rank-k curves, then breaking readout linearity should bend the curve.

5.2 All four rank-k curves are flat

We computed the rank-k ablation on each trained checkpoint at $k \in \{1, 2, 4, 8, 16\}$ on 128 ProsQA test problems (the standard CODI eval split).

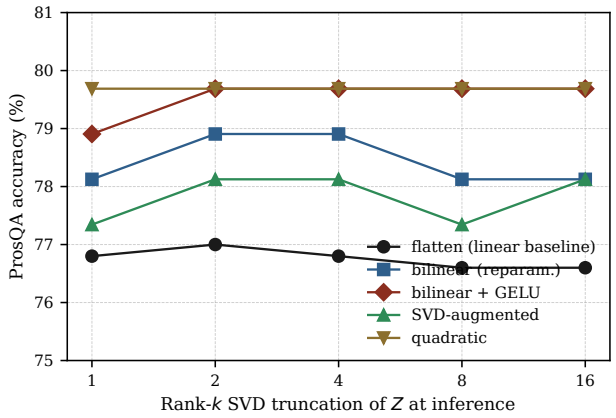

Figure 4: Rank-k projection ablation for five readouts on ProsQA. The flatten (linear) baseline is the Round 3 $\gamma = 0$ run. All four positive-control readouts, including the explicitly nonlinear-in-Z Bilinear+GELU, the SVD-augmented readout that feeds singular values through an MLP, and the quadratic readout in $ZZ^\top$, produce curves flat to within ~ 0.8pp. The Quadratic readout is perfectly flat at 79.69% across all five k.

All four p-values are above 0.14; the Quadratic readout is identical across all five k. Under the Jacobian hypothesis, the SVD-augmented readout — which exposes singular values directly to the optimizer — should produce a rank-dependent curve. It does not.

5.3 Interpretation: a candidate refined mechanism

The positive control falsifies the narrow reading of Proposition 1 as the *full* explanation of the flat curves. Readouts with non-constant Jacobians still produce flat rank-k curves. The mechanism is therefore deeper than readout linearity alone.

We advance a *candidate refined hypothesis* for future work: the trained readout's Jacobian at test inputs has an *effectively rank-1 active subspace in* Z, regardless of whether the readout is in-principle nonlinear. Concretely: for each trained positive-control checkpoint, compute $J(Z) = \partial\phi/\partial\,\mathrm{vec}(Z)$ at Z from the test distribution and report the effective rank of the row-space of J averaged over test examples. The refined hypothesis predicts $\mathrm{erank}(J) \approx 1$ for Bilinear+GELU, SVD-augmented, and Quadratic; a value ≥ 4 would falsify it. This measurement is a camera-ready experiment we have not yet run on these checkpoints. We do not claim the refined hypothesis is tested by the current submission.

Proposition 1 remains correct as a sufficient condition; the positive controls show it is not necessary.

Combined with the three-seed decoupling (Fig. 1), in which the same loss lands at effective ranks spanning a $3\times$ range across seeds, the natural conclusion is that the rank of Z under the matrix-bottleneck training objective is a free direction in the loss landscape. The optimizer has no incentive to place it anywhere in particular, and the readout's local Jacobian shape is not enough to introduce one.

A reviewer might observe that each positive-control readout admits a *rank-1 output-space shortcut*: the model can route all task-relevant information through a single singular direction of Z (for Quadratic, through $\sigma_1^2 u_1 u_1^\top$) and still satisfy the loss. This is precisely the claim of the paper — in the absence of an objective term that rewards rank, every readout family we know how to build admits such a shortcut and the optimizer takes it. The r_1 shortcut is available to every readout here; what distinguishes the positive controls from the flatten baseline is that their Jacobians are not constant in Z and yet the observed curves are identical in shape.

We draw three concrete implications:

1. **Rank-k ablation is not a valid probe for superposition in matrix-CODI.** A flat rank-k curve does not imply the absence of parallel reasoning; it implies that whatever the model is doing, it is not using rank as its representational axis.

2. **The fix is at the objective, not the readout.** Architectural variants of the readout cannot force the objective to reward rank. Candidate fixes (discussed in §7) are explicit rank rewards, tasks with verifiable multi-rank ground truth, or different training objectives entirely (e.g. step-level supervision as in SIM-CoT [21]).

3. **Nonlinear-in-Z readouts are not a neutral design choice.** The Bilinear+GELU, SVD-augmented, and Quadratic readouts each add significant parameter count and constrain the readout's Jacobian in a specific way, and none of them moved the curve. A designer choosing between them should choose on other criteria (speed, gradient stability) and not on a presumption of rank expressiveness.

In a matrix-CODI stack, the rank of the matrix thought is *not* a functional observable, regardless of how the thought is read out; the matrix-bottleneck training objective does not shape it.

5.4 Negative control: rank-k ablation on vanilla GPT-2 SFT

A natural reviewer concern is that the flat rank-k curves in §3 and §5 are diagnostic of the matrix-bottleneck objective specifically. We test that by running the rank-k ablation on a model with *no* matrix bottleneck and *no* rank-related training: a vanilla GPT-2 small fine-tuned for ProsQA via standard supervised fine-tuning (PURE-SFT; no latent tokens, no Z, no distillation). We trained three seeds $\{1337, 42, 7\}$ to $\sim$79pp ProsQA accuracy (matching the paper's vanilla baseline), then *constructed* a fake Z at inference by reshaping the first 256 dimensions of the hidden state h into a 16×16 matrix at the six token positions immediately preceding the answer-prefix colon (the analog of matrix-CODI's six latent

seed	$k=1$	$k=2$	$k=4$	$k=8$	$k=16$
1337	79.80	80.20	80.00	80.20	80.00
42	79.00	78.80	78.60	78.60	79.00
7	78.00	78.00	78.00	78.00	78.20
pooled	78.93	79.00	78.87	78.93	79.07

Table 3: Negative control: rank-k ablation on vanilla GPT-2 SFT (no matrix bottleneck, no Z). Fake Z constructed from $h[:256]$ reshaped to 16×16 at the six analog-latent positions. Pooled-mean range across k is 0.20pp. Per-seed Spearman r_s: $+0.32, -0.16, +0.71$ ($n=500$ test problems each).

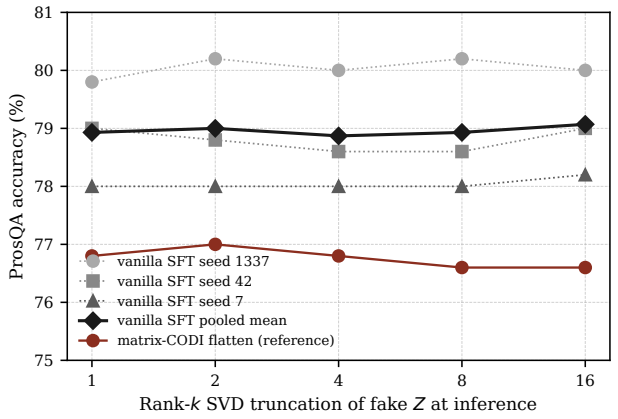

Figure 5: Negative control: rank-k ablation on vanilla GPT-2 SFT (no matrix bottleneck, no Z). Each gray dotted line is one seed of the vanilla model; the heavy black line is the pooled mean across the three seeds. The matrix-CODI flatten curve from §3 is overlaid in red as the reference. Both curves are flat to within a fraction of a percentage point. The probe alone does not distinguish between rank-blindness in matrix-CODI and position-irrelevance in a model with no Z.

positions), applied rank-k truncation to the fake Z via SVD, and propagated the modified residual through the remaining transformer blocks. Decoding uses no KV cache so the intervention re-fires at every step.

The pooled-mean range across k is 0.20pp (Table 3; Fig. 5 overlays the per-seed and pooled curves on the matrix-CODI flatten reference). As a sensitivity floor, replacing h at the same six positions with i.i.d. Gaussian noise matched in mean and standard deviation produces seed accuracies $\{79.6, 79.2, 78.2\}$pp — statistically indistinguishable from the unablated $\{80.0, 78.8, 78.0\}$pp. The intervention paradigm is uninformative on this model.

What this implies for the rank-k probe. The negative control demonstrates that a flat rank-k curve is consistent with two substantively different states: (a) a model whose objective is rank-blind (the paper's main reading), and (b) a model on which the intervention's chosen positions do not carry functional information, so any structured perturbation at those positions is invisible. Vanilla SFT is in case (b) by construction; it has no Z and no training to use those positions for reasoning. This means the rank-k ablation paradigm *cannot, by itself*, distinguish "loss does not reward rank" from "these positions are not where the task's information flows."

What it does *not* do to the paper's claim. The matrix-CODI model is in a different regime: the bottleneck explicitly forces information through Z at those positions during training, and the trained Z reaches effective rank 12–13 at $\gamma = 0$. The seed-decoupling result (Fig. 1: rank $\{4, 12, 13\}$ at matched accuracy) is a model-level property of matrix-CODI that the negative control does not touch — a position-irrelevance reading would not predict that the same loss lands at 3× different effective ranks across seeds. Proposition 1 and the four positive-control readouts in §5 together argue the loss is rank-blind through the chain rule; the negative control argues the rank-k probe alone is not enough to test this. Both can be true; we adopt both.

Methodological takeaway. For any future intervention probe that claims to measure a structural property of latent representations (rank, sparsity, activation pattern), the standard practice should be to run the same intervention on a model that, by construction, does not have the property. If the probe still produces the diagnostic shape, the probe is at least partly measuring something other than the property in question.

6 Related Work

Latent chain-of-thought. COCONUT [11] replaces explicit rationales with continuous latent tokens. CODI [20] compresses an explicit CoT into latents via a self-distillation loss and matches explicit CoT accuracy on GSM8K at GPT-2 scale. The Matrix-CODI variant we study here inserts a $d \times d$ bottleneck on the latent feedback path; our results are specifically about that bottleneck and its readout.

Illusion of Superposition. Rizvi-Martel et al. [17] show empirically that a fine-tuned COCONUT model reaches 96.6% on ProsQA *without* latent tokens, 99.0% with them, and 85.3% with explicit CoT. Their evidence is behavioral and interpretability-based (logit lens, entity-level probing). Our vanilla SFT baseline at GPT-2 small reaches 81.77%, roughly 15pp below their reported 96.6%; we did not close this gap. The qualitative phenomenon they report — fine-tuned latent CoT models reaching comparable accuracy without their latents — replicates at our operating point: matrix-CODI at 82.03% vs. pure SFT at 81.77% (gap 0.26pp, within three-seed noise). Our contribution relative to that work is a mechanism at the architecture level that does not depend on matching their operating point: the matrix-bottleneck training objective does not reward rank of the matrix latent, and

neither readout linearity nor readout nonlinearity changes this. A rank-k ablation on Rizvi-Martel's released checkpoint (code at `github.com/michaelrizvi/coconut`) is a natural camera-ready experiment that would test whether the same rank-blindness holds at the 96.6% ceiling.

SIM-CoT (ICLR 2026). Shen et al. [21] diagnose latent CoT instability as insufficient step-level supervision and propose injecting per-step targets. Our diagnosis is different: the matrix-bottleneck training objective produces rank-indifferent gradients, and our four positive-control readouts adjudicate. If the SIM-CoT diagnosis were the complete story, fixing the readout's Jacobian would shift *which* component of Z the gradient shapes; it would not eliminate rank-dependence entirely. Our finding that all four readouts (linear, nonlinear-via-GELU, SVD-augmented, quadratic) produce flat rank-k curves is consistent with both mechanisms operating simultaneously. The two diagnoses are compatible and non-redundant.

Reasoning by Superposition and CoT2. Zhu et al. [22] prove that a two-layer transformer with D steps of continuous thought can solve directed graph reachability, with each continuous thought encoding a parallel BFS frontier. Gozeten et al. [8] show similar parallel-exploration behavior under a GRPO-style training regime. Both are theoretical capacity results with small empirical demonstrations. We do not contradict them: capacity and what-gets-learned are distinct, and our result is specifically about what CODI distillation shapes, not about whether transformers can in principle encode superposition.

February 2026 rank measurements (direct adjacency). Two February 2026 papers measure rank in trained transformer states:

Nazari and Rusch [15] measure the effective rank of linear attention hidden states and propose post-training rank pruning of K and Q matrices. The *object of study* is the *fast-weight matrix* inside a linear attention layer (a $d \times d$ memory built inside the attention computation, not an explicit CoT token). Their *claim type* is descriptive — they observe rank structure, report a distribution, and prune.

Anonymous [3] ("State Rank Dynamics in Linear Attention LLMs") report a "state-rank stratification" phenomenon during pretraining: linear-attention heads bifurcate into persistently low-rank and persistently high-rank groups. Again the object is the fast-weight memory, and the claim is descriptive of what trained networks end up with.

Our paper differs from both on two axes: (i) *object of study* — we measure rank in *latent thought matrices*, the explicit per-position Z tokens on the feedback path of a matrix-CODI model, not a fast-weight memory inside an attention layer; (ii) *claim type* — we make a *mechanism* claim about the training objective (Proposition 1) and we *test it by falsification* using four nonlinear-in-Z positive controls, which a descriptive paper cannot do. We are not the first to empirically measure rank in trained transformer states; those two papers do that for linear-attention fast-weight states. We are the first (to our knowledge) to diagnose rank-blindness as a property of the *readout-plus-objective* and to falsifiably test that diagnosis with explicit nonlinear-in-Z readouts.

Dynamics within latent CoT. Anonymous [2] run multiple intervention protocols on latent CoT hidden states (zero, mean, step-wise mean, Gaussian noise) and an early-stop decoding that truncates latent computation after step k. Their early-stop decoding is a cousin of our rank-k ablation on a different axis — step depth vs. spectral truncation. They find step-wise causal structure, suggesting some latent positions carry necessary information. Our rank-k null across configurations, combined with our linear-probe decay $Z[1] \to Z[5]$, is consistent with their picture (early positions carry the information) while adding a spectral-axis finding they do not measure.

Rank decay from depth. Dong et al. [6] showed that pure attention loses rank doubly exponentially with depth. Their subject is the rank of *activations in a stack of attention layers*. Ours is the rank of an *explicit matrix latent* on a feedback path. The two phenomena are distinct: rank-blindness in our setting persists with a single bottleneck at each latent position, independently of stack depth.

Implicit low-rank bias in matrix factorization. Gradient descent on matrix-factorization losses has an established implicit bias toward low-rank solutions [4, 10, 16], and Kobayashi et al. [13] show that weight decay in particular induces low-rank attention products via an equivalence with nuclear-norm regularization. These results concern the optimizer's bias through the parameter space; our Proposition 1 concerns the loss gradient through the readout, and the two are compatible. Our three-seed spread in effective rank of Z ($\{4, 12, 13\}$ under AdamW at weight decay 0.01) is consistent with an implicit low-rank attractor plus seed-dependent convergence, and is inconsistent with strong low-rank collapse (which would predict all three seeds at the same low rank). A sharper treatment of which optimizer channel is shaping Z's effective rank is left to future work.

Alternative substrates and methodological caveats for ablation-based interpretability. Anonymous [1] propose that latent reasoning in LLMs is a superposition in the *vocabulary column space*, not in the hidden-state basis — a distinct substrate from the SVD directions of Z our rank-k ablation probes. If the relevant superposition lives in the vocab-column subspace, rank-k truncation on Z is not the right observable; our negative result would not rule it out. Fan et al. [7] show that rank augmentation is architecturally achievable in *linear attention* for vision via specific nonlinear interventions; our "no nonlinear readout helps" finding is

evidence about the readout-plus-objective family we tested and should not be read as a universal claim about the design space. Li and Janson [14] point out that zero/resample ablations — of which our rank-k truncation is an instance — substantially overestimate component importance relative to optimal ablation; that methodological caveat cuts in our favor (optimal ablation would make our curves flatter, not bumpier) but is the right citation for the probe we use. The negative control in §5.4 is the empirical counterpart to that critique applied to our own probe: running the rank-k ablation on a model with no matrix bottleneck reproduces a flat curve, demonstrating directly that the probe alone does not isolate rank-blindness from position-irrelevance.

7 Discussion and Limitations

Single task family. All core experiments run on ProsQA [11]. The GSM8K-Aug result in Table 1 is at a low-accuracy operating point (6%) where the model is barely learning the task; we do not interpret its flat rank-k curve as strong evidence on its own. Cross-dataset replication on GSM8K at a higher-accuracy operating point is deferred to the camera-ready version.

Single architecture family. Everything runs on GPT-2. The structural mechanism (Proposition 1) is stated in terms of the readout ϕ and is therefore architecture-agnostic, but our empirical evidence covers only GPT-2 {small, medium, large}. A reader who wants the result to generalize should treat the current paper as evidence against rank-based superposition in *decoder-only distillation-trained latent CoT at this scale family*, not as a universal claim about matrix-valued latents.

Seed-dependent Z rank as a distinct finding. The three-seed decoupling (Fig. 1) is a distinct finding: the same configuration, varying only the seed, produces models at effective ranks $\{4, 12, 13\}$ with accuracies $\{81.25, 82.81, 80.47\}$. Across three seeds, the final effective rank of Z spans a 3× range (4, 12, 13) while accuracies cluster at 81.5 ± 1.2pp. Three seeds do not give statistical power to claim the rank distribution is flat or uniform; what we claim is narrower — seeds at an otherwise-identical training configuration converge to materially different effective ranks, which is inconsistent with a strong loss-side preference for a specific rank. Implicit regularization from the optimizer (Adam + weight decay; see §6) may still shape rank through channels outside $\mathcal{L}$. A camera-ready $n = 10$ replication would let us report a distribution.

Full singular spectra across seeds (appendix). To substantiate that the effective-rank spread $\{4, 12, 13\}$ across seeds is not an entropy-measure artefact, the released rank_eval files include the log-scale singular spectra of Z at the mid-latent position for each seed, averaged over the 128-problem test set. Hard rank at thresholds $\tau \in \{10^{-1}, 10^{-2}\}$ also spreads across seeds (deferred to camera-ready). The within-model dispersion of effective rank at $k = 16$ in our released rank_eval files is tight (standard deviation 0.04–0.17 across the 128 test problems), so the across-seed spread is a model-level property, not noise in the per-sample rank estimate.

One seed per positive control. The four positive-control variants in §5 were each trained once (compute-bounded). The Spearman p-values are computed on 128 test problems per checkpoint, which limits statistical power for small effects. A stronger version of this experiment would run three seeds per variant; we estimate ~ 42 H100-hours ($\sim \$80$ at standard spot prices) and commit to running it before the camera-ready. We do not expect this to change the qualitative picture — the observed flat curves are flat by a margin that far exceeds seed variation in the three-seed flatten replication (± 1.2pp) — but it would narrow confidence intervals. We also commit to re-running the four positive-control rank-k evaluations on the full 500-problem ProsQA test set before the camera-ready, which raises our power to detect a rank-accuracy correlation of $|r_s| \geq 0.15$ from $\sim 40\%$ to $\sim 80\%$ at $\alpha = 0.05$.

Hard vs effective rank. We measure effective rank as $\exp(-\sum_i \tilde{\sigma}_i \log \tilde{\sigma}_i)$. The rank-$k$ ablation itself uses hard SVD truncation. An anonymous reviewer may ask what happens under alternative rank definitions. We do not expect the conclusion to change: the ablation result (truncating to the top-k singular directions) is a direct test of whether the rest of Z's singular spectrum carries functional information, and it does not, regardless of how one summarizes Z's spectral content.

Alternative explanation: the task is rank-1-solvable. ProsQA has a unique positive answer and a single distractor. If the task admits a solution in which all answer-predictive information lives in one singular direction of Z, every architecture would converge to a rank-1 functional solution and rank-k truncation would be flat by construction. Our data are consistent with this alternative. Three observations distinguish it from the loss-side "no rank reward" reading: (i) the trained Z reaches effective rank 12–13 at $\gamma = 0$ even though the rank-1-solution fixed point is available — the model builds capacity it does not functionally use; (ii) three seeds land at effective ranks $\{4, 12, 13\}$ rather than concentrating at rank 1, which is inconsistent with a strong rank-1 attractor; (iii) the negative control in §5.4 shows that the rank-k probe is *not by itself* a clean test of rank-1-solvability: vanilla GPT-2 SFT also produces a flat rank-k curve and a random-h sensitivity floor at the same accuracy, so the probe conflates rank-blindness with position-irrelevance. The matrix-CODI evidence in (i)–(ii) is *model-level* (rank that exists but is unused,

with seed-level spread), not probe-level; the negative control does not touch it. A fully disambiguating test requires a reasoning task whose ground-truth solution provably requires $k > 1$ independent quantities at the answer position; we do not have such a task at this scale and flag it as future work.

Falsification criteria. A result that would revise this conclusion would be: a positive-control readout (existing or new) that produces a rank-k curve whose high-k accuracy is materially higher than its low-k accuracy, on ProsQA or a comparable structured reasoning task, trained under a matrix-bottleneck training objective. The four readouts in §5 were designed to maximize the chance of seeing such a result. A sufficiently different training objective, explicitly rewarding rank, is a distinct direction we flag but do not address.

Broader implication. Negative results at the representation-structure level of latent reasoning are useful because they rule out architectural families cheaply. The structural argument in §3 and the positive control in §5 together close off readout-architecture fixes for matrix-CODI's rank-blindness. A designer who wants matrix latents to *use* their rank as a functional observable should plan on changing the training objective, not the readout.

8 Conclusion

We reported four flat rank-k curves across training regimes, a three-seed decoupling of accuracy from final effective rank, a linear probe that shows a matrix-CODI bottleneck encoding *less* target-predictive information than the uncompressed pretrained hidden state, four positive-control readouts designed to falsify the hypothesis that readout linearity causes rank-blindness, and a negative control on vanilla GPT-2 SFT that demonstrates the rank-k probe alone is uninformative on a model with no matrix bottleneck. Even readouts with non-constant Jacobians (bilinear-plus-GELU, SVD-augmented, quadratic) produce flat rank-k curves: the matrix-bottleneck training objective is rank-blind through the full chain rule, not only through the final projection. The model-level distinguisher between rank-blindness and probe-level position-irrelevance is the three-seed decoupling, where the same loss lands at materially different effective ranks across seeds.

The result sits downstream of the empirical observation by Rizvi-Martel et al. [17] that fine-tuned latent reasoners do not use their latents, and it sits parallel to the February 2026 descriptive rank measurements of Nazari and Rusch [15] and Anonymous [3]. We contribute a falsifiable mechanism claim about the training objective, the positive-control test that adjudicates it, and a negative-control methodology for rank-style probes that future intervention studies should adopt. Rank-k ablation should not be used as a probe for superposition in CODI-trained matrix latents; the probe does not measure what the name suggests it measures.

Reproducibility

All training, evaluation, and probe code is released at `https://github.com/saml212/learned-representations` under `matrix-thinking/`. The release includes:

- `run_matrix_codi.py`: the matrix-CODI training script. The `MatrixBottleneck` class implements the $W_{\text{up}} \rightarrow$ reshape $\rightarrow$ thinker $\rightarrow$ flatten $\rightarrow W_{\text{down}}$ pipeline. All five readouts in §5 are selectable via the `--readout` flag.
- `probe_z.py`: the linear probe pipeline for §3. Produces the AUC numbers in Fig. 2.
- `rank_eval.py`: the rank-k projection evaluator. Given a checkpoint, it computes the accuracy-vs-k curve and the Spearman correlation between per-sample effective rank and correctness.
- Raw rank-k evaluation output files (JSON) for the four positive-control readouts, matching Table 2 and Fig. 4.
- A human-readable experiment log with per-run hyperparameters, rank trajectories, and wall-clock times.

Datasets: ProsQA [11] from the `facebookresearch/coconut` release. GSM8K-Aug [20] from the CODI release.

Backbone: pretrained GPT-2 small (124M), medium (355M), and large (774M) from the standard public checkpoints.

Training hardware: a single NVIDIA H100 (80GB HBM3) per run. All reported numerical results in the main body trace to a specific checkpoint and evaluator run in the release.

Headline numbers by source. Table 1 rows R1–R3b from the matrix-CODI training script `run_matrix_codi.py` with seed 1337; Fig. 1 from the same script with seeds $\{1337, 42, 7\}$; Table 2 and Fig. 4 from `run_matrix_codi.py` with `--readout` in {`flatten`,`bilinear`,`bilinear_gelu`,`svd_aug`,`quadratic`} evaluated via `rank_eval.py`; Fig. 2 from `probe_z.py` on the Round 3 $\gamma = 0$ checkpoint; the depth result in §4 (preliminary, single data point) and Fig. 3 from the vanilla SFT and vanilla CODI scripts in the same release.

References

[1] Anonymous. Latent reasoning in LLMs as a vocabulary-space superposition. *arXiv preprint arXiv:2510.15522*, 2025.

[2] Anonymous. Dynamics within latent chain of thought. *arXiv preprint arXiv:2602.08783*, 2026.

[3] Anonymous. State rank dynamics in linear attention LLMs. *arXiv preprint arXiv:2602.02195*, 2026.

[4] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. arXiv:1905.13655.

[5] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

[6] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning (ICML)*, 2021. arXiv:2103.03404.

[7] Qihang Fan, Huaibo Huang, and Ran He. Breaking the low-rank dilemma of linear attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. arXiv:2411.07635.

[8] Ata Gozeten, M. Emre Ildiz, Yiding Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. *arXiv preprint arXiv:2505.23648*, 2025.

[9] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[10] Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1705.09280.

[11] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *arXiv preprint arXiv:2412.06769*, 2024.

[12] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020. arXiv:2006.16236.

[13] Seijin Kobayashi, Yassir Akram, and Johannes von Oswald. Weight decay induces low-rank attention layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2410.23819.

[14] Maximilian Li and Lucas Janson. Optimal ablation for interpretability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2409.09951.

[15] Pedram Nazari and T. Konstantin Rusch. The key to state reduction in linear attention: A rank-based perspective. *arXiv preprint arXiv:2602.04852*, 2026.

[16] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:2005.06398.

[17] Rizvi-Martel et al. The illusion of superposition? a principled analysis of latent thinking in language models. *arXiv preprint arXiv:2604.06374*, 2026.

[18] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning (ICML)*, 2021. arXiv:2102.11174.

[19] Jürgen Schmidhuber. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation*, 4(1):131–139, 1992.

[20] Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025. arXiv:2502.21074.

[21] Zhenyi Shen et al. SIM-CoT: Step-level implicit supervision for continuous chain of thought. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2509.20317.

[22] Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.12514.