Latent Thinking Optimization: Your Latent Reasoning Language Model Secretly Encodes Reward Signals in Its Latent Thoughts

Hanwen Du[1], Yuxin Dong[1] and Xia Ning[1, 2, 3]

[1]Department of Computer Science and Engineering, The Ohio State University, USA

[2]Department of Biomedical Informatics, The Ohio State University, USA

[3]Translational Data Analytics Institute, The Ohio State University, USA

{du.1128, dong.1357, ning.104}@osu.edu

LLM Thinking in the Latent Space

- Model the thinking processes as ***latent representations***, not ***natural language***
- Representative example: Huginn-3.5B, a latent reasoning language model

- Strength: efficient, can work with abstract logic
- Weakness: lack of interpretability and supervision

- How does Huginn-3.5B think in the latent space?
- How external supervision signals can improve its latent thinking process?

Visualization of Latent Thoughts

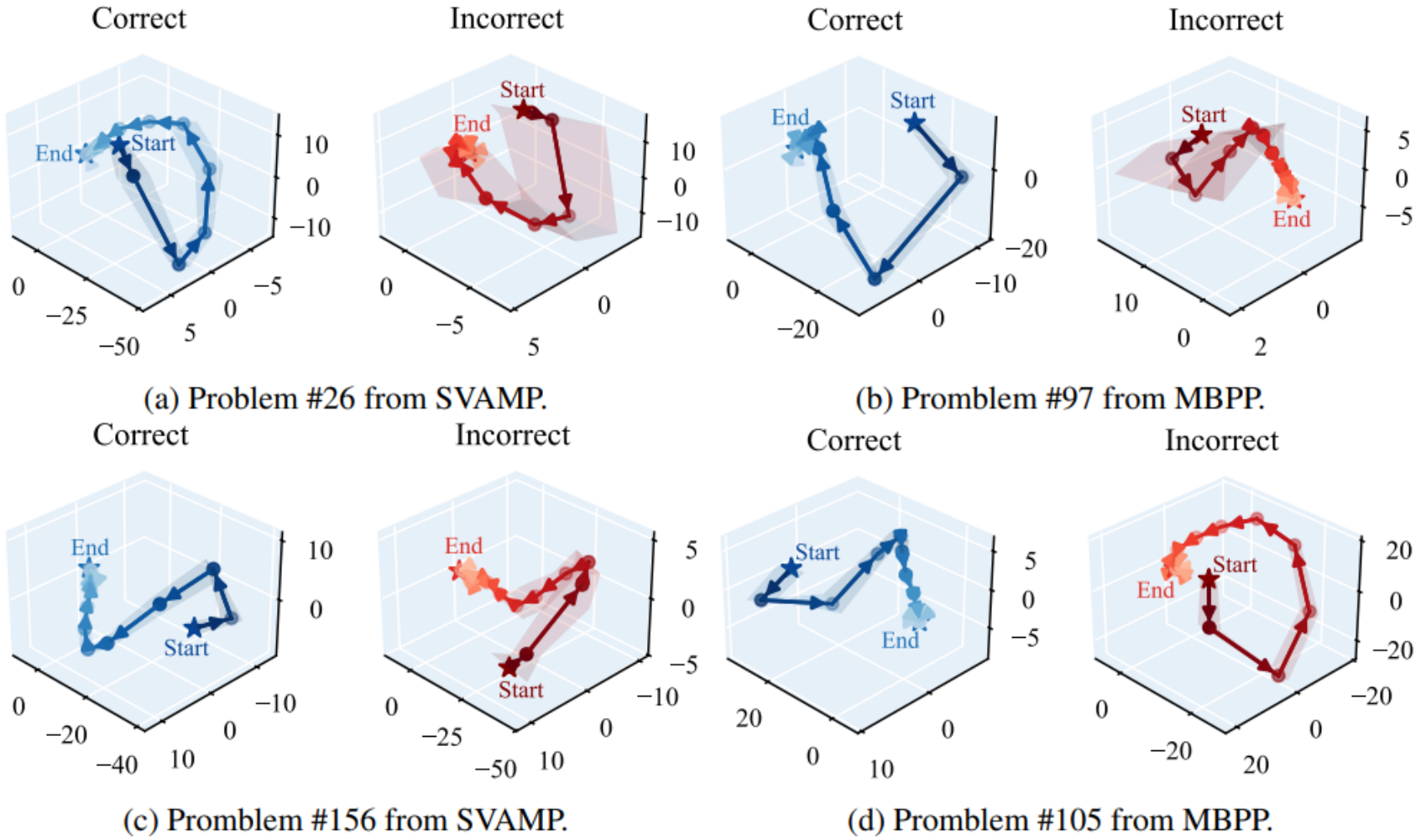


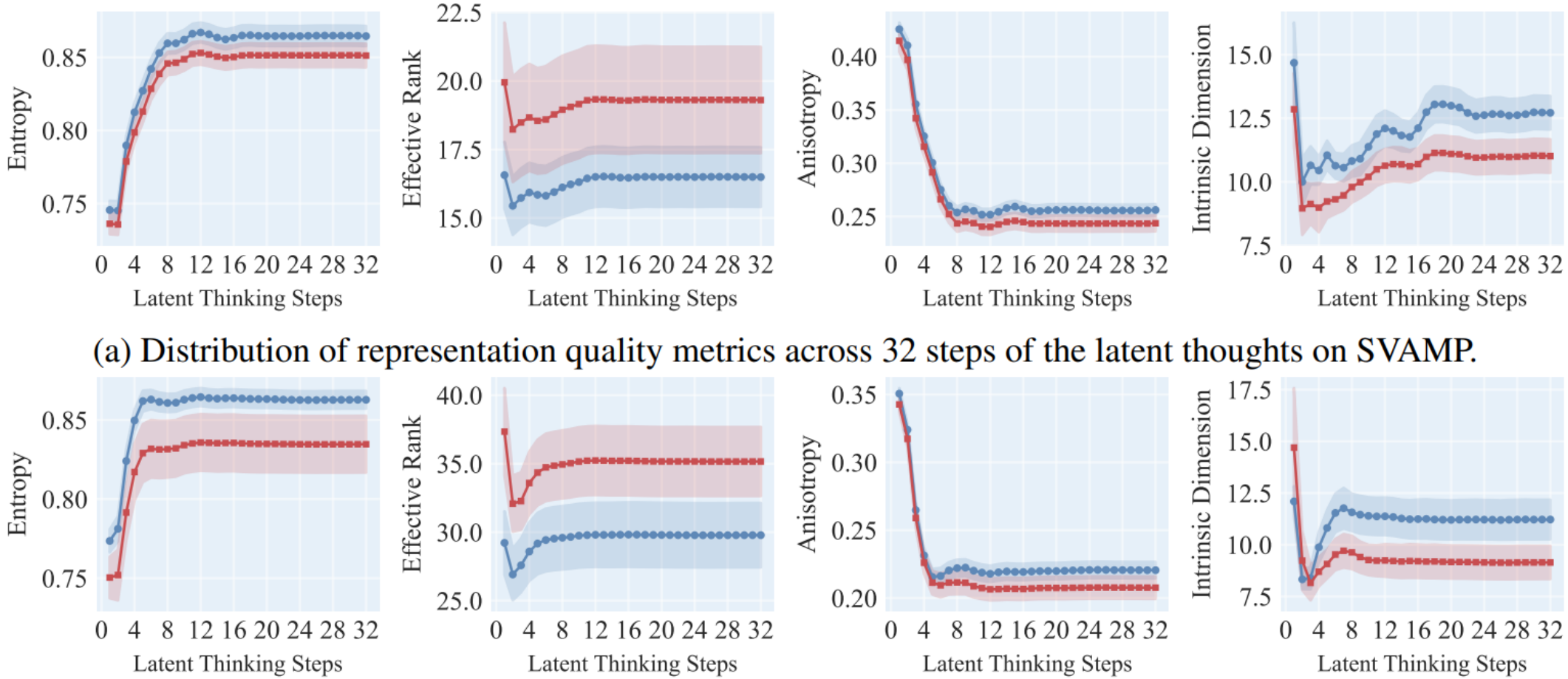
Figure 1. Visualization of the distribution of the correct and incorrect latent thoughts projected onto 3D space using PCA for dimension reduction.

Qualitative and Quantitative Analyses

Analyze latent thoughts with representation quality metrics:

- **Entropy** [1] quantifies the information content in the latent representations.
- **Effective Rank** [2] measures how dimensionality of the latent representation effectively shrinks under strong compression.
- **Anisotropy** [3] measures the non-uniformity of a distribution in the latent space.
- **Intrinsic Dimension** [4][5] quantifies the minimal number of coordinates to describe the structure of the representations

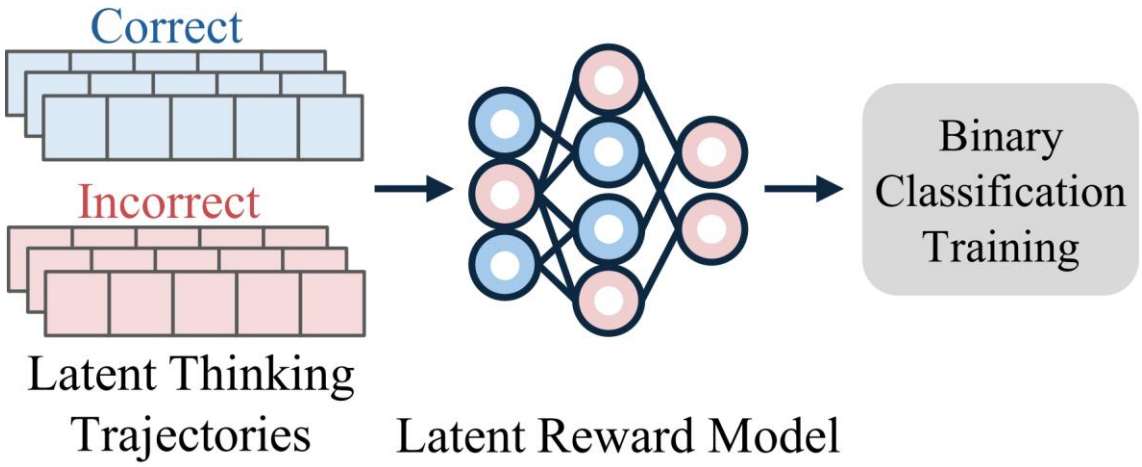
Qualitative and Quantitative Analyses



(a) Distribution of representation quality metrics across 32 steps of the latent thoughts on SVAMP.

Figure 2. Representation quality metrics of the latent thoughts.

Training a Latent Reward Model (LRM)

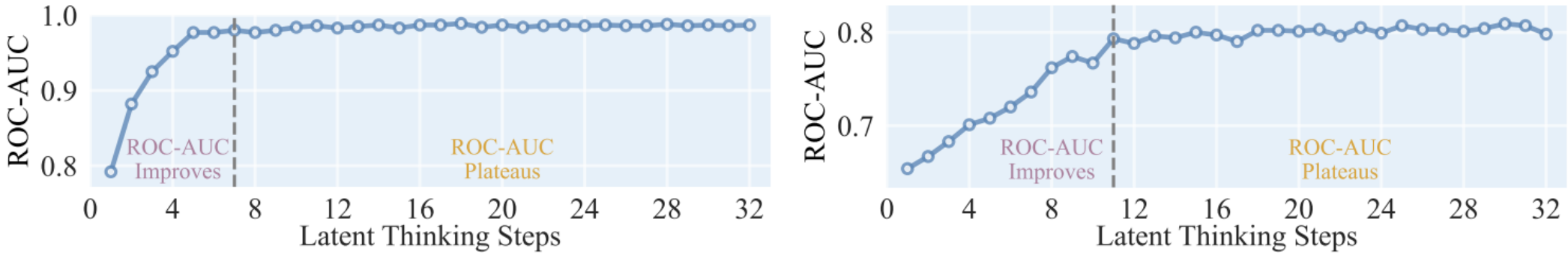


An illustration of training a latent classifier as the Latent Reward Model (LRM)

If these signals indeed capture the distinction between correct and incorrect thinking processes,

can we train a latent classifier to identify their correctness directly from the latent thoughts?

Training a Latent Reward Model (LRM)



(a) ROC-AUC of the latent classifier on SVAMP. (b) ROC-AUC of the latent classifier on MBPP.

Figure 3. Performance of the latent classifier trained with varying numbers of thinking steps.

Strong classification performance on the test set!

Major Observations

- There are distinct thinking patterns between correct and incorrect thinking processes.
- Such difference can be reliable distinguished by a latent classifier,
- Especially after a few thinking steps.

LTO Algorithm

Algorithm 1 Latent Thinking Optimization

Input: question x, the original policy $\pi_{\text{ref}}(z|x)$, LRM $r(x, z)$, sampling budget N, the number of required samples M, weight $\beta > 0$ to control the strength of KL-regularization
Output: sampled set of latent thinking trajectories $\mathcal{C}$
$\mathcal{C} \leftarrow \emptyset$, $r_{\max} \leftarrow 0$ ▷ Initialize the output set and the maximum reward.
for $i = 1$ **to** N **do**
    $z_i \leftarrow z \sim \pi_{\text{ref}}(z|x)$ ▷ Sample the i-th latent thinking trajectory from the original policy.
    $r_{\max} \leftarrow \max\{r_{\max}, r(z_i, x)\}$ ▷ Update the maximum reward.
while $|\mathcal{C}| < M$ **do** ▷ Repeat until M samples are collected.
    $z_i \sim \text{Uniform}\{z_j\}_{j=1}^{N}, u_i \sim \text{Uniform}(0, 1)$
    $\phi_i \leftarrow \exp((r(z_i, x) - r_{\max})/\beta)$ ▷ Calculate the acceptance probability ϕ_i.
    if $u_i \geq \phi_i$ **then continue** ▷ Reject the sample z_i with probability $1 - \phi_i$.
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{z_i\}$ ▷ Otherwise, accept the sample z_i with probability ϕ_i.
return $\mathcal{C}$

Workflow Summary:

- Collect latent thinking trajectories to train LRM
- Sample multiple latent thinking trajectories
- Accept trajectories that are more likely to be correct

Performance on Huginn-3.5B

Table 1: Comparison of the answer correctness rate of Huginn-3.5B using different correction methods. The best performance in each column is in **bold**, and the performance of the best baseline in each column is underlined. $*$ indicates statistically significant improvement with $p < 0.05$.

Method	GSM8K	GSM-Symbolic	SVAMP	CommonsenseQA	MBPP
Base Model	0.326	0.265	0.517	0.500	0.278
Majority Voting	0.333	0.269	0.511	0.504	0.288
Self-Correction w. Confidence Score	0.342	0.281	0.524	0.507	0.288
Self-Correction w. Verbal Evaluation	0.262	0.193	0.518	0.505	0.226
Latent Thinking Correction w. CoE-R	0.330	0.259	0.510	0.504	0.276
Latent Thinking Correction w. CoE-C	0.324	0.256	0.516	0.507	0.280
Weighted Majority Voting w. LRM	0.375*	0.301*	0.537*	0.509	0.295*
Latent Thinking Optimization w. LRM	**0.385***	**0.305***	**0.538***	**0.517***	**0.299***

- LTO significantly improves the latent thinking processes.
- LRM is highly effective in detecting incorrect latent thinking patterns.

Performance on General LLMs

Table 2: Performance of LTO on general LLMs. The best-performing method for each model is in **bold**. $*$ indicates the improvement over the best runner-up is statistically significant with $p < 0.05$.

Model	Method	GSM8K	GSM-Symbolic	SVAMP	CommonsenseQA	MBPP
OLMo-7B	Base Model	0.124	0.078	0.297	0.464	0.244
	Majority Voting	0.209	0.149	0.469	0.521	0.240
	Latent Thinking Optimization	**0.252***	**0.154***	**0.552***	**0.602***	**0.308***
Llama-2-7B	Base Model	0.223	0.204	0.473	0.399	0.189
	Majority Voting	0.275	0.302	0.598	0.493	0.193
	Latent Thinking Optimization	**0.389***	**0.316***	**0.776***	**0.606***	**0.237***
Llama-2-13B	Base Model	0.306	0.273	0.521	0.398	0.247
	Majority Voting	0.417	0.379	0.612	0.501	0.263
	Latent Thinking Optimization	**0.534***	**0.442***	**0.791***	**0.650***	**0.322***
Mistral-7B	Base Model	0.368	0.278	0.548	0.671	0.315
	Majority Voting	0.529	0.413	0.624	0.687	0.334
	Latent Thinking Optimization	**0.565***	**0.462***	**0.771***	**0.708***	**0.388***

- LTO can also improves the latent thinking processes of general LLMs.

Performance on General LLMs

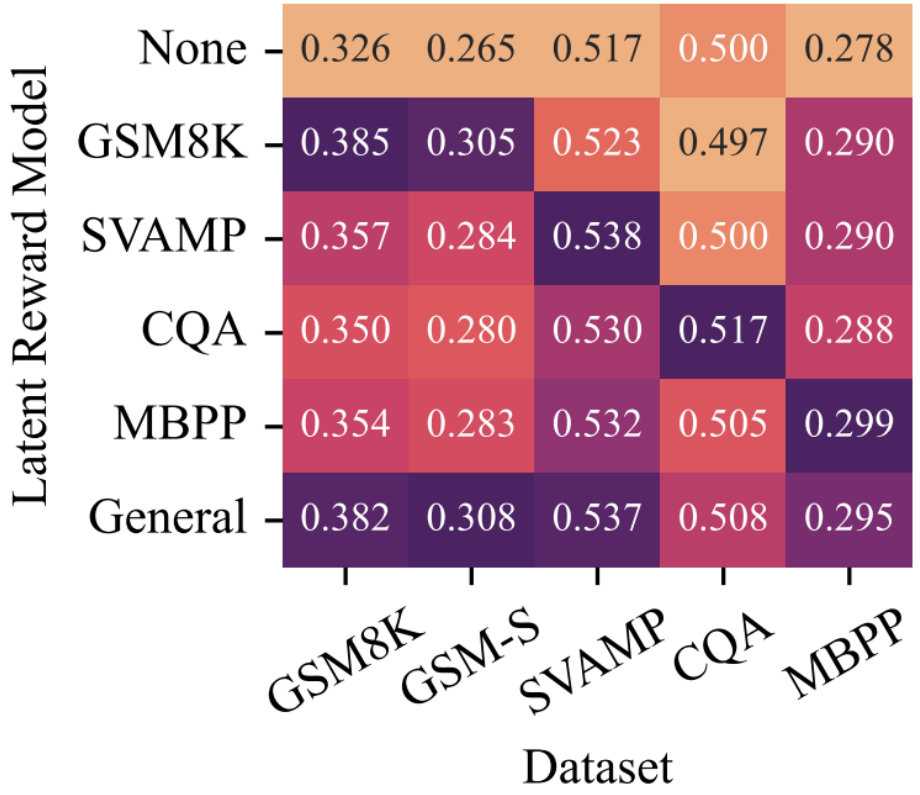


Figure 4: Performance of LTO using different LRMs.

- LRMs demonstrate transferability across different domains.
- A general LRM can be applicable to multiple domains.
- LRMs show strong potential for building a generalist reward model in the latent space.

Takeaways

- Latent thoughts leading to correct versus incorrect answers exhibit highly distinguishable patterns.
- A latent classifier can reliably predict answer correctness directly from latent thoughts.
- The latent classifier can serve as a reward model to guide the optimization of the latent thinking processes.
- Scaling test-time thinking with supervision can be performed directly in the latent space, as a general, efficient, and domain-agnostic approach.

Acknowledgement

- The authors thank the anonymous reviewers for their insightful comments and constructive feedback.

References

[1] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In Forty-second International Conference on Machine Learning, 2025.

[2] Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. Diff-eRank: A novel rank-based metric for evaluating large language models. In Advances in Neural Information Processing Systems, volume 37, pp. 39501–39521, 2024.

[3] Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models. In Findings of the Association for Computational Linguistics: EACL 2024, pp. 868–874, 2024.

References

[4] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. Scientific reports, 7:12140, 2017.

[5] Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Lei Yu, Alessandro Laio, and Marco Baroni. Emergence of a high-dimensional abstraction phase in language transformers. In The Thirteenth International Conference on Learning Representations, 2025.

Thank you!