Dispersion loss counteracts embedding condensation and improves generalization in small language models

Chen Liu*[1], Xingzhi Sun*[1], Xi Xiao*[2,3], Alexandre Van Tassel*[1], Ke Xu[1], Kristof Reimann[1], Danqi Liao[1], Mark Gerstein[1], Tianyang Wang[2], Xiao Wang[3], Smita Krishnaswamy[1]

* Equal contribution [1] Yale University [2] University of Alabama at Birmingham [3] Oak Ridge National Laboratory. Correspondence to: Smita Krishnaswamy <smita.krishnaswamy@yale.edu>.

ICML International Conference On Machine Learning

One-liner summary: *What makes LLMs better than small LMs? Data? Parameters? Geometry might play a role!*

Motivation

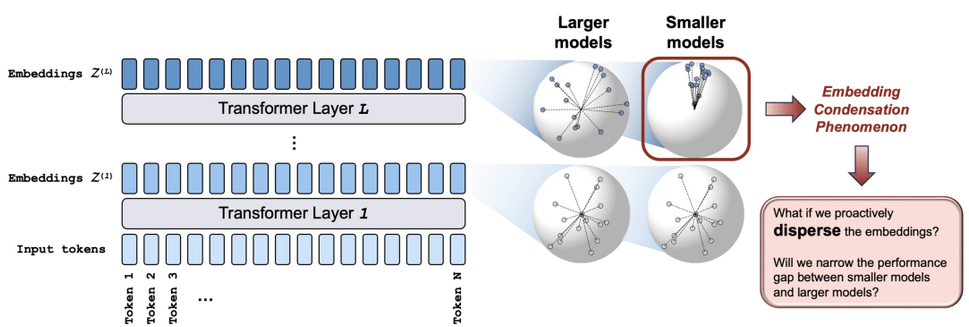

Figure 1. Illustration of the embedding condensation phenomenon. In pre-trained language models, embeddings of all tokens from the same input sequence condense into a narrow cone after being processed by many Transformer layers. This phenomenon is substantially more pronounced in smaller models than in larger models within the same family, which motivates our hypothesis in Section 3.3.

Features of embedding condensation

Feature 1. More severe in smaller models than in larger counterparts (Figure 2).

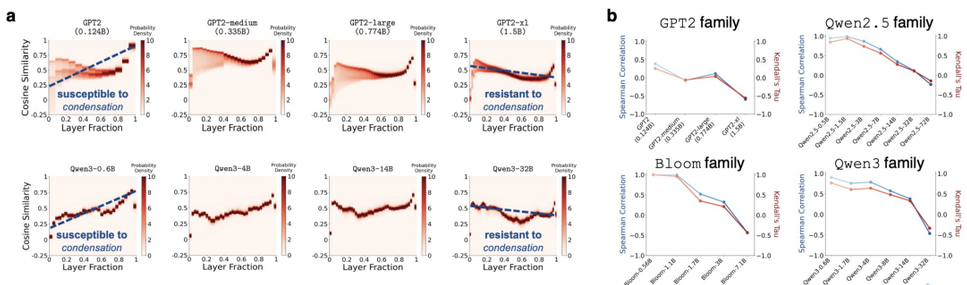

Figure 2. Qualitative and quantitative observations of the embedding condensation phenomenon. **a.** The cosine similarity heatmaps demonstrate that smaller models (e.g., `GPT2`, `Qwen3-0.6B`) are susceptible to condensation, since token cosine similarities become increasingly positive as the embeddings proceed to deeper layers. In contrast, larger models (e.g., `GPT2-xl`, `Qwen3-32B`) are more resistant to embedding condensation. **b.** Quantifications using Spearman correlation and Kendall's Tau demonstrate a consistent trend of "larger model, less condensation" across multiple families of language models. Additional results can be found in Figure S1.

Feature 2. Reproducible under confounder-controlled settings (Figure 3).

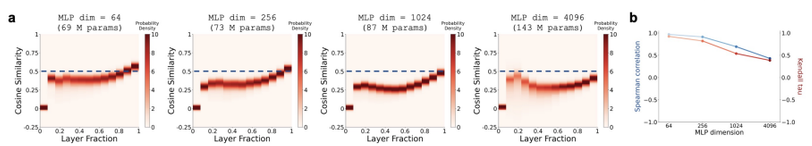

Figure 3. In a highly controlled experiment, we reproduced the observation of "larger model, less condensation". We pre-trained four `GPT2`-like models of varying sizes that differ only in MLP dimension, while keeping all other factors fixed, including the number of layers, embedding dimension, dataset, and training configuration. The resulting models exhibit consistent trends in embedding condensation, shown qualitatively (panel **a**) and quantitatively (panel **b**). Horizontal dashed lines are added to panel **a** for easier visual comparison.

Feature 3. Emerging at model initialization (Figure 4).

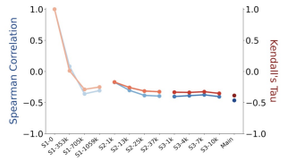

Figure 4. Embedding condensation is observed immediately after model initialization. We analyze checkpoints of `Olmo-3-1025-7B` spanning initialization, intermediate pre-training stages, and the final base model. Each checkpoint is annotated by its training stage and the number of training tokens.

Feature 4. Not resolved by distillation from a larger model (Figure 5).

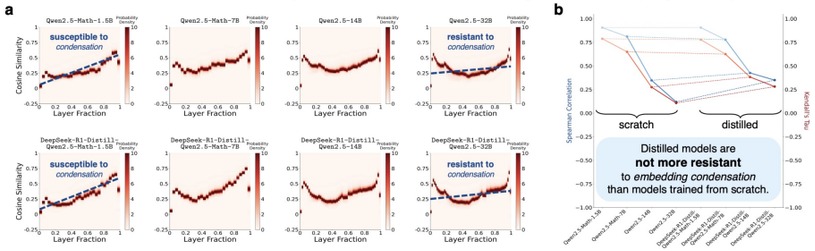

Figure 5. Knowledge distillation is not a remedy to embedding condensation, shown qualitatively (panel **a**) and quantitatively (panel **b**).

Solution: dispersion loss

Table 1. Our dispersion loss and its alternative formulations. Main implementation differences from (Wang & He, 2025) are highlighted in teal and magenta. Including or excluding diagonal terms yields identical gradients and is therefore cosmetic. For dispersion loss and ℓ_2-repel, we adopt the `log-sum-exp` trick for numerical stability, which differs from $\log(\text{mean}(\exp(\cdot)))$ only by an additive constant. For ℓ_2-repel, we include a norm regularization term to prevent unbounded expansion of embeddings. For Orthogonalization, the distance margin is fixed to $\frac{1}{2}$ since we use angular distance, where $\frac{1}{2}$ corresponds to orthogonality and thus serves as the ideal margin.

	For generative modeling in diffusion-based models (Wang & He, 2025)		For improving generalization of Transformer-based language models (Ours)	
	formulation	term definition	formulation	term definition
Dispersion loss	$\log \mathbb{E}_{i,j}[\exp(-D(z_i,z_j)/\tau)]$	$D(z_i,z_j) = -\text{cossim}(z_i,z_j)$	$\log \sum_{i,j}^{i\neq j}[\exp(-D(z_i,z_j)/\tau)]$	$D(z_i,z_j) = \frac{\arccos(\text{cossim}(z_i,z_j))}{\pi}$
Alternative formulations				
Decorrelation	$\sum_{m,n}\text{Cov}^2_{mn}$	$\text{Cov}^2 = \frac{Z_c^\top Z_c}{d-1}$, $Z_c = \frac{Z-\mu_Z(Z)}{\sigma_Z(Z)}$	$\sum_{m,n}^{m\neq n}\text{Cov}^2_{mn}$	$\text{Cov}^2 = \frac{Z_c^\top Z_c}{d-1}$, $Z_c = \frac{Z-\mu_Z(Z)}{\sigma_Z(Z)}$
ℓ_2-repel	$\log \mathbb{E}_{i,j}[\exp(-D(z_i,z_j)/\tau)]$	$D(z_i,z_j) = \|Z_{i,:} - Z_{:,j}\|_2^2$	$\log \sum_{i,j}^{i\neq j}[\exp(-D(z_i,z_j)/\tau)]+\lambda_{\text{norm}}\|Z\|_2^2$	$D(z_i,z_j) = \|Z_{i,:} - Z_{:,j}\|_2^2$
Orthogonalization	$\mathbb{E}_{i,j}[\max(0, \epsilon - D(z_i,z_j))^2]$	$D(z_i,z_j) = -\text{cossim}(z_i,z_j)$	$\mathbb{E}_{i\neq j}^2[\max(0, \frac{1}{2} - D(z_i,z_j))^2]$	$D(z_i,z_j) = \frac{\arccos(\text{cossim}(z_i,z_j))}{\pi}$

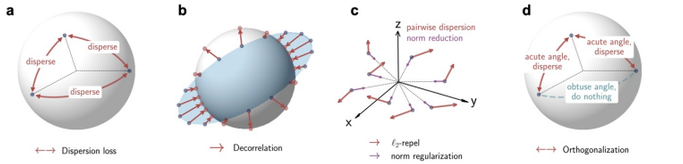

Figure 6. Illustration of how dispersion loss and its alternative formulations promote embedding dispersion. **a.** Dispersion loss enforces uniform angular dispersion by spreading out all pairs along the unit hypersphere. **b.** Decorrelation loss encourages different feature dimensions to remain uncorrelated. **c.** ℓ_2-repel loss increases pairwise Euclidean distance, while the norm regularization prevents unbounded expansion. **d.** Orthogonalization loss spreads out vectors forming acute angles while leaving obtuse ones unchanged.

Results: mid-training

Table 2. Using dispersion loss during mid-training improves performance on language tasks. For each base model, the best single-benchmark metrics are displayed in bold, whereas the best average ranks and best average performances are boxed and in bold. We also perform the Student's t-tests on the average performances and report the significance level with respect to "$\mathcal{L}_{\text{train}}$ + Dispersion loss".

Model	Mid-training		Training time	Zero-shot					Few-shot				Rank↓	Average↑	t-test	
	Train	Loss	A100 hours	ANLI_{R3}↑	$\text{LAMBADA}_{\text{openai}}$↑	OpenbookQA↑	PIQA↑	TruthfulQA↑	WinoGrande↑	ARC_{easy}↑	$\text{ARC}_{\text{challenge}}$↑	MedMCQA↑	MMLU↑				
GPT2	✗	—	—	34.4	30.8	15.6	61.6	40.3	52.4	42.0	16.2	25.4	24.8	6.1	34.35	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$	1.122 (1.00×)	34.0 ± 0.3	32.4 ± 0.6	16.4 ± 0.1	62.2 ± 0.7	43.6 ± 0.2	53.8 ± 3.1	41.6 ± 0.0	17.4 ± 0.6	24.2 ± 3.8	24.8 ± 0.4	6.2	34.95 ± 0.11	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + noisy embedding	1.122 (1.00×)	34.3 ± 0.1	33.0 ± 0.3	16.9 ± 0.4	61.1 ± 0.7	43.9 ± 0.2	53.6 ± 2.0	43.7 ± 0.4	18.0 ± 0.6	21.6 ± 2.5	25.4 ± 0.4	4.3	35.15 ± 0.06	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + active forgetting	1.127 (1.00×)	**35.0** ± 0.3	33.8 ± 1.1	16.6 ± 0.3	61.0 ± 0.0	44.0 ± 0.1	52.2 ± 0.8	**44.3** ± 0.4	**18.6** ± 1.1	22.4 ± 1.7	**25.7** ± 0.2	3.2	35.36 ± 0.15	*n.s.*	
	✓	$\mathcal{L}_{\text{train}}$ + Decorrelation	1.221 (1.09×)	**35.0** ± 0.0	33.6 ± 0.5	**17.0** ± 0.6	60.8 ± 0.6	43.9 ± 0.3	52.0 ± 1.7	43.8 ± 0.5	18.0 ± 1.3	24.4 ± 2.3	25.6 ± 0.2	3.6	35.41 ± 0.06	*n.s.*	
	✓	$\mathcal{L}_{\text{train}}$ + ℓ_2-repel	1.175 (1.05×)	34.6 ± 0.3	32.8 ± 0.2	16.6 ± 0.2	60.8 ± 0.3	43.9 ± 0.2	**54.2** ± 0.8	43.4 ± 0.2	17.6 ± 0.8	24.6 ± 2.0	**25.7** ± 0.1	4.1	35.42 ± 0.11	*n.s.*	
	✓	$\mathcal{L}_{\text{train}}$ + Orthogonalization	1.210 (1.08×)	34.4 ± 0.1	31.2 ± 1.1	16.8 ± 0.7	61.6 ± 0.0	**44.2** ± 0.4	53.4 ± 0.8	44.2 ± 1.0	18.0 ± 0.3	24.8 ± 3.3	25.6 ± 0.2	3.2	35.42 ± 0.19	*n.s.*	
	✓	$\mathcal{L}_{\text{train}}$ + Dispersion loss	1.176 (1.05×)	**34.8** ± 0.0	34.6 ± 0.0	16.6 ± 0.0	**62.4** ± 0.3	**44.2** ± 0.1	51.6 ± 0.1	42.6 ± 0.4	18.0 ± 0.8	**26.6** ± 2.3	25.4 ± 0.2	3.2	**35.61** ± 0.12	*ref.*	
GPT2-m	✗	—	—	33.3	40.4	18.4	66.3	40.1	54.8	50.2	19.9	**29.1**	25.3	5.4	38.80	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$	2.541 (1.00×)	33.1 ± 0.6	43.2 ± 0.3	19.1 ± 0.0	67.7 ± 0.4	40.5 ± 1.3	55.5 ± 1.0	**54.2** ± 2.7	18.9 ± 0.7	28.2 ± 2.3	25.1 ± 0.3	4.5	38.55 ± 0.18	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + noisy embedding	2.580 (1.01×)	33.3 ± 0.1	44.0 ± 0.3	18.6 ± 0.6	66.7 ± 0.1	**44.5** ± 0.6	52.1 ± 1.6	51.7 ± 1.6	19.5 ± 1.8	25.5 ± 0.1	25.8 ± 0.3	4.8	38.16 ± 0.25	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + active forgetting	2.580 (1.01×)	33.2 ± 0.3	43.0 ± 0.6	18.9 ± 0.1	66.7 ± 0.1	44.3 ± 0.2	51.8 ± 2.5	19.8 ± 1.4	26.1 ± 0.6	**26.1** ± 0.0	25.4 ± 0.4	4.3	38.43 ± 0.14	*n.s.*	
	✓	$\mathcal{L}_{\text{train}}$ + Decorrelation	2.760 (1.09×)	33.4 ± 0.0	**45.4** ± 1.3	18.9 ± 0.8	66.6 ± 0.1	42.2 ± 0.1	56.2 ± 0.4	53.8 ± 3.3	18.0 ± 0.8	28.2 ± 3.0	25.4 ± 0.2	4.0	38.81 ± 0.12	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + ℓ_2-repel	2.675 (1.05×)	**33.6** ± 0.8	44.2 ± 0.8	18.5 ± 0.6	66.2 ± 0.1	42.4 ± 0.1	54.4 ± 1.1	54.0 ± 2.0	18.6 ± 0.6	28.9 ± 2.1	25.3 ± 0.4	4.7	38.61 ± 0.03	$p < 0.05$	
	✓	$\mathcal{L}_{\text{train}}$ + Orthogonalization	2.692 (1.06×)	33.0 ± 0.4	45.2 ± 0.4	18.6 ± 0.6	**68.1** ± 0.4	41.4 ± 0.2	55.1 ± 0.6	53.6 ± 2.7	18.4 ± 0.4	29.0 ± 0.4	25.0 ± 0.1	4.8	38.74 ± 0.30	$p < 0.05$	
	✓	$\mathcal{L}_{\text{train}}$ + Dispersion loss	2.673 (1.05×)	**33.6** ± 0.1	45.2 ± 0.8	**19.2** ± 0.2	67.5 ± 0.1	43.4 ± 0.2	**56.4** ± 1.1	53.8 ± 1.1	**20.1** ± 0.9	28.6 ± 0.1	25.7 ± 0.3	2.2	**39.35** ± 0.15	*ref.*	
GPT2-1	✗	—	—	33.6	47.8	19.6	71.6	38.9	58.4	54.0	22.4	26.2	25.6	—	39.81	—	
GPT2-xl	✗	—	—	36.4	49.4	22.2	71.8	38.0	57.2	58.0	23.6	27.0	25.2	—	40.89	—	
Qwen3-0.6B	✗	—	—	35.0	43.0	19.5	66.5	40.7	60.5	67.5	32.5	26.5	49.4	5.7	44.11	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$	4.676 (1.00×)	32.5 ± 0.3	**52.0** ± 0.3	21.5 ± 0.5	**67.5** ± 1.4	44.3 ± 0.9	**61.0** ± 0.8	68.0 ± 2.8	33.0 ± 1.3	29.5 ± 0.9	**50.0** ± 0.1	4.1	45.93 ± 0.37	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + noisy embedding	4.859 (1.03×)	34.0 ± 0.7	48.8 ± 0.4	20.5 ± 0.7	67.2 ± 1.1	44.5 ± 0.7	57.5 ± 0.7	65.0 ± 1.4	33.0 ± 1.4	28.8 ± 0.0	49.4 ± 0.2	5.3	44.77 ± 0.28	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + active forgetting	4.829 (1.03×)	35.0 ± 0.7	49.0 ± 4.2	20.0 ± 2.1	67.2 ± 0.4	48.6 ± 0.9	58.8 ± 1.8	69.0 ± 1.4	34.2 ± 1.1	36.8 ± 0.4	49.3 ± 0.0	4.0	46.79 ± 0.27	$p < 0.05$	
	✓	$\mathcal{L}_{\text{train}}$ + Decorrelation	4.939 (1.06×)	35.0 ± 0.0	50.5 ± 0.4	19.5 ± 1.8	**67.5** ± 1.1	47.0 ± 2.3	59.5 ± 1.8	68.5 ± 3.2	**35.0** ± 1.8	34.0 ± 0.4	49.8 ± 0.2	3.5	46.62 ± 0.25	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + ℓ_2-repel	4.864 (1.04×)	33.5 ± 0.0	46.0 ± 2.1	19.0 ± 0.7	66.0 ± 0.4	47.4 ± 0.2	59.5 ± 1.4	69.0 ± 3.2	**35.0** ± 1.8	**40.0** ± 0.4	46.9 ± 0.2	4.8	46.23 ± 0.04	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + Orthogonalization	4.936 (1.06×)	**35.5** ± 1.1	50.0 ± 0.7	22.0 ± 0.7	65.5 ± 0.4	49.1 ± 0.1	56.5 ± 2.1	71.5 ± 3.2	33.0 ± 1.1	36.0 ± 0.7	48.9 ± 0.2	4.2	46.80 ± 0.20	$p < 0.05$	
	✓	$\mathcal{L}_{\text{train}}$ + Dispersion loss	4.878 (1.04×)	**35.5** ± 0.8	49.5 ± 0.8	**22.5** ± 0.6	65.0 ± 1.6	**49.8** ± 1.1	58.5 ± 1.6	**72.5** ± 2.3	34.5 ± 0.8	37.5 ± 1.3	49.2 ± 0.2	3.2	**47.45** ± 0.16	*ref.*	
Qwen3-1.7B	✗	—	—	39.5	53.0	29.0	71.5	47.6	60.5	72.0	50.0	45.5	63.1	5.1	53.18	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$	9.148 (1.00×)	40.0 ± 0.9	60.7 ± 0.3	28.5 ± 0.9	74.3 ± 0.6	49.1 ± 0.2	61.3 ± 0.3	71.3 ± 1.4	47.7 ± 1.2	49.7 ± 1.2	63.1 ± 0.0	4.3	54.57 ± 0.17	$p < 0.001$	
	✓	$\mathcal{L}_{\text{train}}$ + noisy embedding	9.345 (1.02×)	35.0 ± 0.0	60.5 ± 2.8	28.0 ± 0.0	70.5 ± 0.7	**50.3** ± 0.9	60.8 ± 3.2	73.2 ± 0.4	48.5 ± 0.0	48.5 ± 2.1	63.5 ± 0.6	5.0	53.78 ± 0.07	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$ + active forgetting	9.298 (1.02×)	35.2 ± 0.4	57.2 ± 1.1	27.5 ± 0.7	72.5 ± 0.0	49.9 ± 0.2	60.5 ± 1.4	**74.2** ± 0.4	47.2 ± 1.1	49.0 ± 1.4	63.1 ± 0.2	5.0	53.89 ± 0.07	$p < 0.001$	
	✓	$\mathcal{L}_{\text{train}}$ + Decorrelation	9.535 (1.04×)	39.5 ± 1.5	60.7 ± 1.0	28.3 ± 1.2	74.8 ± 1.0	49.7 ± 0.5	61.5 ± 0.9	72.7 ± 1.5	49.7 ± 0.8	49.7 ± 1.0	63.5 ± 0.2	3.1	55.01 ± 0.13	$p < 0.01$	
	✓	$\mathcal{L}_{\text{train}}$ + ℓ_2-repel	9.440 (1.03×)	33.5 ± 0.0	47.8 ± 0.6	23.0 ± 0.0	67.8 ± 1.2	42.1 ± 0.2	57.5 ± 0.5	68.3 ± 1.9	41.3 ± 1.8	38.5 ± 2.6	42.3 ± 0.5	8.0	46.22 ± 0.57	$p < 0.0001$	
	✓	$\mathcal{L}_{\text{train}}$ + Orthogonalization	9.565 (1.05×)	**40.8** ± 1.3	61.0 ± 0.5	29.2 ± 0.3	74.8 ± 1.3	49.5 ± 0.2	61.3 ± 0.8	72.3 ± 1.9	48.2 ± 1.6	49.5 ± 1.3	63.7 ± 0.0	3.0	55.04 ± 0.25	$p < 0.05$	
	✓	$\mathcal{L}_{\text{train}}$ + Dispersion loss	9.455 (1.03×)	40.2 ± 2.0	**62.2** ± 0.8	29.0 ± 1.0	**75.2** ± 0.8	50.2 ± 0.4	**62.7** ± 1.2	73.0 ± 1.3	**49.3** ± 0.8	**51.0** ± 1.8	**64.1** ± 0.2	1.7	**55.68** ± 0.21	*ref.*	
Qwen3-4B	✗	—	—	41.5	60.5	27.5	75.0	52.8	67.0	80.0	58.5	55.5	73.1	—	59.14	—	
Qwen3-8B	✗	—	—	49.5	67.5	33.0	77.5	54.7	71.0	86.0	63.5	56.0	78.2	—	63.68	—	
Qwen3-14B	✗	—	—	54.0	65.5	35.5	77.0	54.7	74.5	86.0	68.5	62.0	81.7	—	65.93	—	

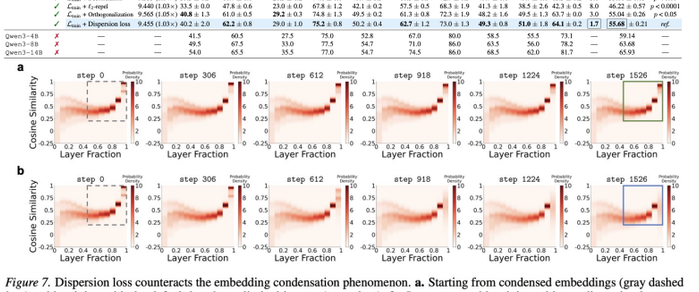

Figure 7. Dispersion loss counteracts the embedding condensation phenomenon. **a.** Starting from condensed embeddings (gray dashed box), mid-training with the default loss has a limited impact (green box). **b.** In contrast, mid-training with our dispersion loss as a regularizer substantially mitigates embedding condensation (blue box).

Results: pre-training

Table 5. Using dispersion loss during pre-training improves performance on language tasks. Experiments are performed under the `Qwen3-0.6B` pre-training setting.

Loss	Zero-shot						Few-shot				Average↑
	ANLI_{R2}↑	$\text{LAMBADA}_{\text{openai}}$↑	OpenbookQA↑	PIQA↑	Truthful QA↑	WinoGrande↑	ARC_{easy}↑	$\text{ARC}_{\text{challenge}}$↑	MedMCQA↑	MMLU↑	
$\mathcal{L}_{\text{train}}$	**34.0**	24.0	**15.5**	64.5	37.8	**58.0**	**41.5**	22.0	26.5	24.6	34.84
$\mathcal{L}_{\text{train}}$ + Dispersion loss	32.0	**27.5**	13.5	**68.5**	**45.2**	54.5	40.0	**24.5**	**29.5**	**24.9**	**36.01** (+1.17)