A Compute-Matched Study of Hidden Layer Distillation for LLM Pre-Training

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

Knowledge Distillation (KD) is a critical tool for training Large Language Models (LLMs), yet the majority of research focuses on approaches that rely solely on output logits, neglecting semantic information in the teacher's intermediate representations. While Hidden Layer Distillation (HLD) showed potential for encoder architectures, its application to decoder-only pre-training at scale remains largely unexplored. Through compute-controlled experiments, we benchmark HLD against logit-based KD and self-supervised baselines with Gemma3 3.4B as teacher and 123M and 735M students trained on up to 168B tokens from the C4 dataset. Our experiments show that HLD does not consistently outperform standard KD on downstream evaluation tasks. Nevertheless, we show that HLD can yield a systematic perplexity gain over KD across all shared-hyperparameter configurations, suggesting that a latent signal can be extracted, but a breakthrough may be needed for it to play a more significant role in LLM pre-training.

1. Introduction and Background

The continuous scaling of Large Language Models (LLMs) [10, 13, 22] has driven remarkable performance gains but at the cost of soaring deployment, latency, and energy expenses. Knowledge Distillation (KD) [11]—training a small "student" to reproduce the behavior of a large "teacher"—is a robust paradigm to mitigate these costs and has shown utility across vision [1], speech [25], and NLP, including encoder-only [23] and modern decoder-only LLMs [7, 27]. Yet, current methods rarely close the student–teacher gap [18, 19, 23, 29] despite known teacher redundancy [5, 8], and academic study of distillation during cost-intensive pre-training remains limited, mostly confined to logits-level KD [4, 19]. Hidden Layer Distillation (HLD), introduced for CNNs by Romero et al. [21] (FitNets), exploits the teacher's hidden states and has shown promise on smaller encoder models [6, 14–16, 23, 24, 29, 30, 32, 33], but has never been scaled to large scale LLMs. We address this gap with a systematic study of HLD for decoder-only pre-training, building on the NanoDo codebase [17], the C4 dataset [20], and Gemma models [27]. Our contributions are: **(1)** we evaluate HLD on 123M and 735M Gemma students trained on up to 168B tokens with a 3.4B Gemma teacher; **(2)** we adopt a rigorous FLOPs-matched protocol that accounts for the non-negligible de-embedding and loss cost; and **(3)** we show that joint hidden+logit optimization (HLDC) matches KD, while sequential optimization (HLDF) yields modest C4 perplexity gains with comparable downstream performance on Wikitext-103, HellaSwag, WinoGrande, LAMBADA, PIQA, and ARC-E.

Transformers and self-supervised training. A decoder-only Transformer [28] maps token indices $\mathbf{T} = [t_1, \ldots, t_n] \in \{0, \ldots, V-1\}^n$ to embeddings $\mathbf{H}^0 \in \mathbb{R}^{d_{emb} \times n}$, applies D residual layers

© .

$\mathbf{H}^{k+1} = \mathbf{H}^k + f_{layer}^{k+1}(\mathbf{H}^k)$, and projects $\mathbf{H}^D$ to logits $\mathbf{Z} \in \mathbb{R}^{V \times n}$, yielding $\mathbf{p}_n = \text{softmax}(\mathbf{z}_n/\tau)$. Without a teacher, pre-training uses the causal negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = -\frac{1}{n-1}\sum_{i=1}^{n-1} \log \mathbf{p}_i[t_{i+1}]. \tag{1}$$

Knowledge Distillation. KD [11] aligns the student distribution $\mathbf{p}_i^S$ with the teacher's $\mathbf{p}_i^T$ via a KL term, $\mathcal{L}_{\text{logits}}(\boldsymbol{\theta}_S) = \frac{\tau^2}{n}\sum_{i=1}^{n} \text{KL}(\mathbf{p}_i^T \| \mathbf{p}_i^S)$, exploiting the richer signal of soft logits over one-hot labels. The standard objective combines it with NLL:

$$\mathcal{L}_{\text{KD}}(\boldsymbol{\theta}_S) = (1-\alpha)\mathcal{L}_{\text{data}}(\boldsymbol{\theta}_S) + \alpha\mathcal{L}_{\text{logits}}(\boldsymbol{\theta}_S). \tag{2}$$

Open-source studies of KD on decoder-only models include Peng et al. [19], who use 9B/32B teachers with 330M–6.8B students, and Busbridge et al. [4], who derive a distillation scaling law with models up to 12.6B parameters. In parallel, industry releases [7, 26, 27, 31] provide strong empirical evidence for distillation in pre-training, though pipelines remain proprietary.

Hidden Layer Distillation. FitNets [21] originally used a two-phase strategy: align student intermediate activations with the teacher's via a learned regressor, then apply standard KD. Transformer adaptations have explored attention-map matching [14], value relations [29, 30], multi-layer allocation schemes [32], task-aware filters [16], transformation-invariant alignment [6], RMS matching [24, 33], and cosine distance on normalized activations [23]. Modern variants typically replace the multi-layer regressor with a simple linear map and merge the two phases into a single aggregated objective:

$$\mathcal{L}_{HLD} = \beta\mathcal{L}_{data} + \alpha\mathcal{L}_{logits} + \gamma\mathcal{L}_{emb}. \tag{3}$$

However, these works almost exclusively target post-training of encoder or encoder-decoder models; the only decoder-only HLD study we are aware of is Liang et al. [16], which uses GPT-2_{12}/GPT-2_6 in a continual pre-training setup. To our knowledge, our work is the first open-source study of HLD during pre-training for causal LLMs.

2. Evaluated Methods

To investigate both the original HLD formulations and contemporary approaches in the Transformer literature, we evaluate the following two methods. See fig. 3 for an illustration.

Sequential Optimization (HLDF). We adapt Romero et al. [21]'s two-phase protocol as a minimal extension to standard distillation that isolates the value of intermediate feature guidance. The regressor $f_{reg}(.;\boldsymbol{\theta_R})$ is a one-layer dense perceptron. Phase 1 minimizes a normalized hint-training loss:

$$\mathcal{L}_{HT}(\boldsymbol{\theta_S}, \boldsymbol{\theta_R}) = \text{MeanSquaredError}\left(\frac{\mathbf{H_T^{D_T/2}}}{\|\mathbf{H_T^{D_T/2}}\|}, \frac{f_{reg}(\mathbf{H_S^{D_S/2}};\boldsymbol{\theta_R})}{\|f_{reg}(\mathbf{H_S^{D_S/2}};\boldsymbol{\theta_R})\|}\right). \tag{4}$$

Phase 2 runs standard KD with $\mathcal{L}_{KD}$ as objective (see section A.1 for details).

Joint Optimization (HLDC). To align with loss functions in the literature, we implement a single-stage training using $\mathcal{L}_{\text{HLD}}$ with $\mathcal{L}_{emb}(\boldsymbol{\theta_S}, \boldsymbol{\theta_R}) = \text{MeanSquaredError}\left(\frac{\mathbf{H_T^{D_T/2}}}{\|\mathbf{H_T^{D_T/2}}\|}, \frac{\mathbf{W_R}\mathbf{H_S^{D_S/2}}}{\|\mathbf{W_R}\mathbf{H_S^{D_S/2}}\|}\right)$. We use HLDF (F as FitNets) for the two-phase regime and HLDC (C as Composite) for the composite-loss formulation.

3. Experiments

We compare HLD against KD in two complementary ways. First, we perform pointwise comparisons at *shared hyperparameter* configurations, evaluating robustness across a broad grid without any method-specific tuning. Second, we give each method its independently selected *best hyperparameters*, enabling a ceiling-to-ceiling comparison. Experimental details can be found in section A.

Evaluation Protocol. A fair comparison must verify that the baseline is well-tuned. Rather than tuning HLD to beat KD, we characterize its general behavior over the peak learning rates η_{KD} and η_{HT}, the temperature τ, and the NLL weight α in the global objective. Also **all compared models are trained with equivalent computational budgets.** We follow the overtraining paradigm [9] and train the 123M student up to OT_{54} (108B tokens) and the 735M student up to OT_{12} (168B tokens). See sections A.8 and A.9 for accounting details.

Shared hyperparameters. For each $(\eta = \eta_{KD} = \eta_{HT}, \tau, \alpha)$ we run one NLL and one KD baseline. For HLDF we use $\eta_{HT} = \eta$ in Phase 1 and (η, τ, α) in Phase 2, varying the budget split P_1 between phases. For HLDC we explore small values of γ [16] at the same (η, τ, α) as the baseline. We choose (η, τ, α) from regimes where KD is known to perform well [19], adapting η to our batch size via square-root scaling [12]. Full ranges are summarized in table 2. To evaluate general performance we avoid "best vs best" comparisons (see section 3) and instead perform pointwise comparisons at identical (η, τ, α); detailed results are in figs. 1, 2 and 4 to 7.
Best Hyperparameters. We ask whether HLD outperforms KD when each method uses its best configuration. We select for each method the best-performing $(\eta^*, \tau^*, \alpha^*)$ from the shared hyperparameter grid table 2, and compare the resulting peak performances head-to-head in table 1.

4. Results

We present results in four views. Figures 1, 2 and 4 show histograms of HLD's improvement over KD at fixed hyperparameter sets. Figures 5 to 7 (appendix) are scatter plots where each point is a shared hyperparameter set, with the KD score on x and the compared method (HLD or NLL) on y. Table 1 reports the best score per method alongside teacher and NLL baselines, and Tables 3 and 4 (appendix) list the full training runs.

Baseline. A meaningful HLD comparison requires a well-tuned KD reference. Our best KD configuration outperforms NLL by a clear margin on every benchmark (table 1), even on noisier evaluations like WinoGrande. On the perplexity scatter plots (fig. 5), all but one KD run beats NLL for the 735M student, corroborated by downstream evaluations (fig. 7). The 123M picture is more mixed, as roughly half the KD runs fail to beat NLL on C4 perplexity, but downstream results (fig. 6) again show KD outperforming NLL on nearly all points. Our KD baseline is thus competitive, a prerequisite not always met in the HLD literature.

Shared hyperparameters. On C4 perplexity, HLDC matches KD while HLDF achieves a modest but systematic improvement (fig. 1). The effect is small and we lack the statistical power for significance claims from independent seeds; we instead rely on the consistent sign of improvement across the entire grid.
Downstream evaluations (figs. 2 and 4) tell a different story: distributions are more spread and roughly centered around zero, indicating that the hidden-layer signal does not translate into consis-

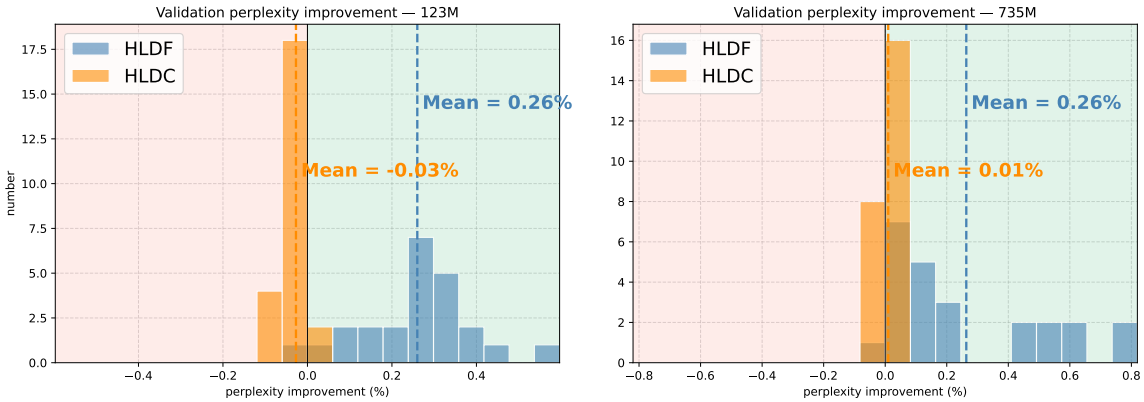

Figure 1: C4 perplexity improvement over KD. Distribution of pointwise improvements across all hyperparameter configurations for both sizes. Positive values indicate lower perplexity than KD.

tent gains. The perplexity scatter (fig. 5) further reveals that HLDF's gain over KD occurs predominantly where KD itself performs poorly, and becomes negligible where KD is already efficient.

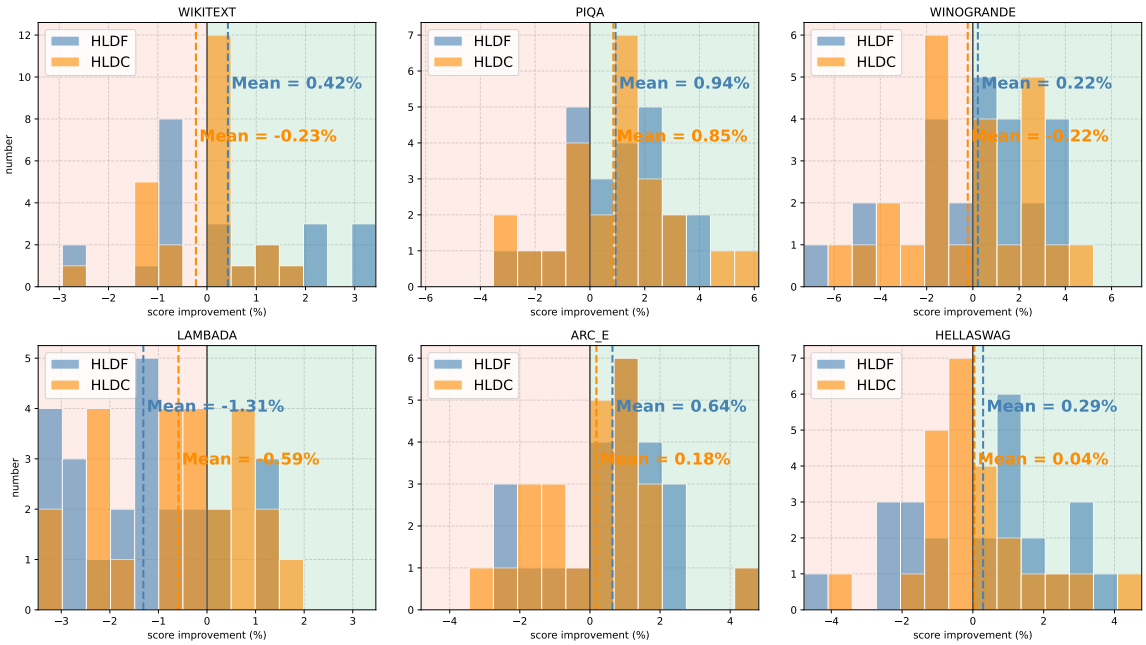

Figure 2: Shared-HP — downstream score improvement over KD for the **735M student**. Distribution of pointwise score improvements across all hyperparameter configurations on benchmarks.

Best hyperparameters. Table 1 reports best-configuration performance for both students. No method dominates KD across the seven evaluations: differences are small and split between methods, reinforcing that, under tuned and compute-matched conditions, HLD provides no consistent improvement over logit-based distillation.

Table 1: **Best Performance Comparison.** Test log-perplexity (↓) or the error rate (↓) for the optimal hyperparameters found within the compute-matched search space. Results are presented as **123M / 735M** student performances. **Bold** indicates the best result for a given student size.

Method	C4	Wikitext	HSwag	PIQA	ARC-E	WinoGd	LMBDA
Random	–	–	0.750	0.500	0.750	0.500	≈ 1.0
Teacher	2.397	2.203	0.319	0.225	0.339	0.365	0.377
Student Performances: 123M / 735M							
NLL	3.063 / 2.639	3.062 / 2.453	.644 / .440	.338 / .261	.529 / .416	.440 / .423	.527 / .445
KD	3.005 / 2.609	2.984 / **2.406**	**.622** / .428	**.321** / .256	**.505** / **.402**	.433 / .407	.513 / **.423**
HLDC	3.005 / 2.608	2.938 / 2.438	.630 / .430	.323 / **.251**	**.505** / .415	**.423** / .405	**.505** / .425
HLDF	**3.000** / **2.607**	**2.875** / 2.422	.626 / **.426**	.322 / .261	.519 / .409	.437 / **.404**	.507/.428

Reconciling with prior work. Our findings appear to contradict prior reports of consistent gains from intermediate-layer matching [6, 14, 16, 23, 24, 29, 30, 33]. Four methodological factors plausibly explain the discrepancy. (i) *Compute accounting:* most prior comparisons match tokens rather than FLOPs, and some HLD variants cost more per token than KD. (ii) *Architecture and training phase:* the HLD literature is dominated by encoder(-decoder) models in post-training or task-specific regimes, where representations are pre-shaped and the alignment target is well-defined; causal decoders trained from scratch present a different optimization landscape in which HLD's inductive bias appears less valuable. (iii) *Baseline tuning:* when HLD is the contribution, the KD baseline is rarely swept as carefully—an asymmetry our shared-hyperparameter protocol explicitly avoids. (iv) *Method maturity:* porting HLD from few-million-parameter CNNs to near-billion-parameter Transformers likely requires deeper design choices (losses, adapter, protocol) to extract value from the teacher's latent representations.

On the residual HLDF signal. HLDF's small but persistent perplexity gain admits a natural interpretation: hint-training may act as a warm-start, aligning the student's mid-network representations to provide a better initialization for the subsequent KD phase without changing what the student can ultimately learn. This would explain why the effect surfaces in C4 perplexity (sensitive to fine-grained distributional fit) but not in downstream benchmarks (sensitive to coarser capabilities that converge under both protocols), and why it vanishes in regimes where KD already performs best.

Conclusion and future work

This work presents a compute-controlled evaluation of Hidden Layer Distillation (HLD) for causal LLM pre-training on English C4. Despite the appeal of aligning intermediate representations, HLD does not consistently outperform a well-tuned logit-based KD baseline when training budgets are strictly equalized, and the complexity of HLD introduces hyperparameter sensitivity that may outweigh its benefits. These findings underscore the resilience of standard KD and the necessity of exact FLOPs accounting when evaluating novel supervision techniques.

Nevertheless, HLDF yields a systematic perplexity gain over KD, suggesting a latent signal that (i) requires stronger statistical evidence across broader contexts (larger scales, different compression ratios, alternative model families), and (ii) likely needs significant design breakthroughs (losses, adapter architecture, protocol) to fully extract the teacher's latent information.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9163–9171. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00938. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Ahn_Variational_Information_Distillation_for_Knowledge_Transfer_CVPR_2019_paper.html.

[2] Jacob Austin, Sholto Douglas, Roy Frostig, Anselm Levskaya, Charlie Chen, Sharad Vikram, Federico Lebron, Peter Choy, Vinay Ramasesh, Albert Webson, and Reiner Pope. How to scale your model. 2025. Retrieved from https://jax-ml.github.io/scaling-book/.

[3] Mathieu Blondel and Vincent Roulet. The elements of differentiable programming. *CoRR*, abs/2403.14606, 2024. doi: 10.48550/ARXIV.2403.14606. URL https://doi.org/10.48550/arXiv.2403.14606.

[4] Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russell Webb. Distillation scaling laws. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=1nEBAkpfb9.

[5] Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. Analyzing redundancy in pretrained transformer models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4908–4926. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.398. URL https://doi.org/10.18653/v1/2020.emnlp-main.398.

[6] Sayantan Dasgupta and Trevor Cohn. Improving language model distillation through hidden state matching. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=IcVSKhVpKu.

[7] DeepSeek-AI. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

[8] Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4865–4880. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.395. URL https://doi.org/10.18653/v1/2020.emnlp-main.395.

[9] Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco,

Pang Wei Koh, Shuran Song, Thomas Kollar, and et al. Language models scale reliably with over-training and on downstream tasks. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=iZeQBqJamf.

[10] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory F. Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017. URL http://arxiv.org/abs/1712.00409.

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531.

[12] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1731–1741, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/a5e0ff62be0b08456fc7f1e88812af3d-Abstract.html.

[13] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

[14] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.FINDINGS-EMNLP.372. URL https://doi.org/10.18653/v1/2020.findings-emnlp.372.

[15] Chen Liang, Haoming Jiang, Zheng Li, Xianfeng Tang, Bing Yin, and Tuo Zhao. Homodistil: Homotopic task-agnostic distillation of pre-trained transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=D7srTrGhAs.

[16] Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 20852–20867. PMLR, 2023. URL https://proceedings.mlr.press/v202/liang23j.html.

[17] Peter J. Liu, Roman Novak, Jaehoon Lee, Mitchell Wortsman, Lechao Xiao, Katie Everett, Alexander A. Alemi, Mark Kurzeja, Pierre Marcenac, Izzeddin Gur, Simon Kornblith, Kelvin Xu, Gamaleldin Elsayed, Ian Fischer, Jeffrey Pennington, Ben Adlam, and Jascha-Sohl Dickstein. Nanodo: A minimal transformer decoder-only language model implementation in JAX., 2024. URL http://github.com/google-deepmind/nanodo.

[18] Zechun Liu, Changsheng Zhao, Forrest N. Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=EIGbXbxcUQ.

[19] Hao Peng, Xin Lv, Yushi Bai, Zijun Yao, Jiajie Zhang, Lei Hou, and Juanzi Li. Pre-training distillation for large language models: A design space exploration. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3603–3618. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.181/.

[20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL https://jmlr.org/papers/v21/20-074.html.

[21] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6550.

[22] Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryenvpEKDr.

[23] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019. URL http://arxiv.org/abs/1910.01108.

[24] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4322–4331. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1441. URL https://doi.org/10.18653/v1/D19-1441.

[25] Ke Tan and DeLiang Wang. Towards model compression for deep learning based speech enhancement. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:1785–1794, 2021. doi: 10.1109/TASLP.2021.3082282. URL https://doi.org/10.1109/TASLP.2021.3082282.

[26] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. doi: 10.48550/ARXIV.2507.06261. URL https://doi.org/10.48550/arXiv.2507.06261.

[27] Gemma Team. Gemma 3 technical report. *CoRR*, abs/2503.19786, 2025. doi: 10.48550/ARXIV.2503.19786. URL https://doi.org/10.48550/arXiv.2503.19786.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[29] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[30] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.188. URL https://doi.org/10.18653/v1/2021.findings-acl.188.

[31] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

[32] Zony Yu, Yuqiao Wen, and Lili Mou. Revisiting intermediate-layer matching in knowledge distillation: Layer-selection strategy doesn't matter (much), 2025. URL https://arxiv.org/abs/2502.04499.

[33] Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. Moebert: from BERT to mixture-of-experts via importance-guided adaptation. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1610–1623. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.NAACL-MAIN.116. URL https://doi.org/10.18653/v1/2022.naacl-main.116.

Appendix A. Experiment details

A.1. Normalization Strategy.

Unlike FitNets, we compare activations after normalization, consistent with prior work [23, 24]. This approach is essential for residual transformer architectures, where the unnormalized residual stream tends to grow significantly in magnitude across layers. Since the student model possesses fewer layers than the teacher, it cannot naturally match the teacher's internal activation scale. For instance, the mean squared magnitude of the teacher ($\mathbf{H_T^{D_T/2}}$) reaches approximately 3×10^5, whereas the pre-trained Gemma3 270M student ($\mathbf{H_S^{D_S/2}}$) only reaches $\approx 1 \times 10^5$.

A.2. Data

We trained the models on the English subset of the C4 dataset [20]. This is a well-established corpus, also used by Busbridge et al. [4], which is sufficiently large to ensure that few samples are seen twice during the training.

A.3. Teacher model.

We pretrained a Gemma3 4B [27] as teacher on 106B tokens with an alternative tokenizer that has a vocabulary size of 32k (making it a 3.4B model). It is a 34 layers pretrained decoder-only Transformer with an embedding dimension d_T of 2560. This choice was motivated by the size of the original vocabulary of gemma models (256k) and the fact that it is multilingual and that we train only on English data.

A.4. Student model.

As student model we used Gemma3 270M and 1B [27] with standard random initialization. They are respectively 18 and 26 layers decoder-only Transformers sharing the architecture pattern of the teacher. Their embedding dimensions d_S are 640 and 1152. Gemma3 270M is a $\sim$100M backbone parameters model, with the 32k tokens vocabulary it becomes a 123M parameters model. Gemma3 1B is a $\sim$700M backbone parameters model, with the 32k tokens vocabulary it becomes a 735M parameters model.

A.5. Compression ratio.

Our choice is motivated by different considerations. We wanted a powerful state of the art teacher. The sizes of the teacher and student should fit but also saturate our computation and storage budget. The pair teacher-student must belong to the same family of models in order to make sure that the activation matching is not blurred by some architecture variation. This choice results in a 27:1 and 4.5:1 model compression ratio which we think is a fair first exploration. The investigation of alternative teacher-student pairs and different compression ratios remains a priority for future research.

A.6. Logits.

A common industry practice for optimizing offline storage is to retain only the top-k logits, where k is significantly smaller than the total vocabulary size; this method reduces storage requirements by

several orders of magnitude. For instance, the Gemma3 tokenizer possesses a vocabulary of approximately 256,000 tokens, and a standard value of $k = 128$ is typically employed. This threshold also functions as a regularizer, shielding the student model from the noise inherent in low-probability logits. In this study, k was fixed at 128 and is not subject to further analysis, as Peng et al. [19] observe that the choice of k makes little difference.

A.7. Intermediate Activations.

However, the application of HLD necessitates the storage of full activations as vectors of the teacher embedding space, preventing the use of the top-k truncation technique. Conventional knowledge distillation serves as the primary baseline for comparison in this study. In order to accommodate storage constraints, minimize hyperparameter optimization, and align with the configurations established by [21], we consider that evaluating the efficacy of hidden layer distillation using the activations from a single teacher layer presents a significant challenge and a robust initial step for investigating hidden layer distillation within this specific framework. Adhering to the methodology established in the FitNet literature, the median layer of the teacher model was selected for data retention and subsequent distillation.

A.8. Compute-matched comparisons.

Following the establishment of scaling laws for LLMs [13], compute-matched comparisons have become a methodological necessity. Consequently, all experimental evaluations in this work are conducted between models trained using equivalent computational budgets. We quantify computational expenditure using an "overtraining unit", denoted as OT_k. This unit represents k times the compute required to reach the Chinchilla-optimal point for our student model with NLL [13]. Since compute-optimal models do not account for serving costs, we follow the overtraining paradigm [9] and trained:

- the 123M student up to OT_{54} (108B tokens)
- the 735M student up to OT_{12} (168B tokens)

A.9. FLOPs accounting.

We refine the standard $6ND$ approximation for training compute, as the conventional formula neglects embedding and de-embedding layers. While these layers are negligible in large-scale Transformers, they represent a significant portion of the total FLOPs for our student models due to the large vocabulary size ($V = 32,000$), and would have been the dominant portion with the original gemma vocabulary.

To ensure a precise comparison across different loss functions, we explicitly include the costs of the de-embedding layer and loss computations.

Following Blondel and Roulet [3], the backward pass cost is equal to the forward pass for non-parametric layers and twice the forward pass for layers with learnable parameters.

Let N denote the number of student backbone parameters, d_s the internal dimension, d_T the teacher's hidden dimension, and N_{reg} the parameters in the mapping regressor. Following Austin

et al. [2], the estimated compute costs per token are:

$$\begin{aligned}
\mathcal{C}_{data} &\approx \underbrace{6N}_{\text{backbone}} + \underbrace{6d_S V}_{\text{logits}} \\
\mathcal{C}_{KD} &\approx \mathcal{C}_{data} + \mathcal{C}_{\text{Teacher}} \\
\mathcal{C}_{HT} &\approx \underbrace{3N}_{\text{half backbone}} + \underbrace{6N_{reg}}_{\text{Mapping}} + \mathcal{C}_{\text{Teacher}}/2 \\
\mathcal{C}_{HLDC} &\approx \mathcal{C}_{KD} + \underbrace{6N_{reg}}_{\text{Mapping}}
\end{aligned}$$

We assume teacher logits are pre-computed or cached for KD, making the forward cost of the teacher C_{Teacher} negligible (overhead limited to KL-divergence computation). For the 123M student, the relative costs are:

$$\frac{\mathcal{C}_{KD}}{\mathcal{C}_{data}} \approx 1.0000, \quad \frac{\mathcal{C}_{HT}}{\mathcal{C}_{data}} \approx 0.442 \quad \text{and} \quad \frac{\mathcal{C}_{HLDC}}{\mathcal{C}_{data}} \approx 1.027$$

Therefore, while Phase 1 of HLDF updates approximately half the model's backbone parameters, its computational cost is significantly less than 50% of the KD baseline for our student. This efficiency is amplified by the high ratio of vocabulary size to model parameters in the 123M architecture, where the bypassed output projection accounts for a large fraction of FLOPs. As model size scales and backbone computations dominate, this advantage vanishes. For example, a Gemma3 27B student [27] with a $2\times$ wider teacher yields $\frac{\mathcal{C}_{HT}}{\mathcal{C}_{data}} \approx 0.5001$.

HLD-specific hyperparameters. To emulate a "no-tuning" scenario, HLDF-specific hyperparameters are fixed to heuristic values: a 1 layer MLP regressor with expansion factor 4 (matching the student's feed-forward layers), and a constant Phase 1 learning rate η_{HT} after a 1,000-step warmup with no decay, so global decay curves remain aligned with standard distillation except for re-warmup at the start of Phase 2. For HLDC the additional hyperparameter is the embedding-loss weight γ.

Appendix B. Figures and Tabs

B.1. HLD schema

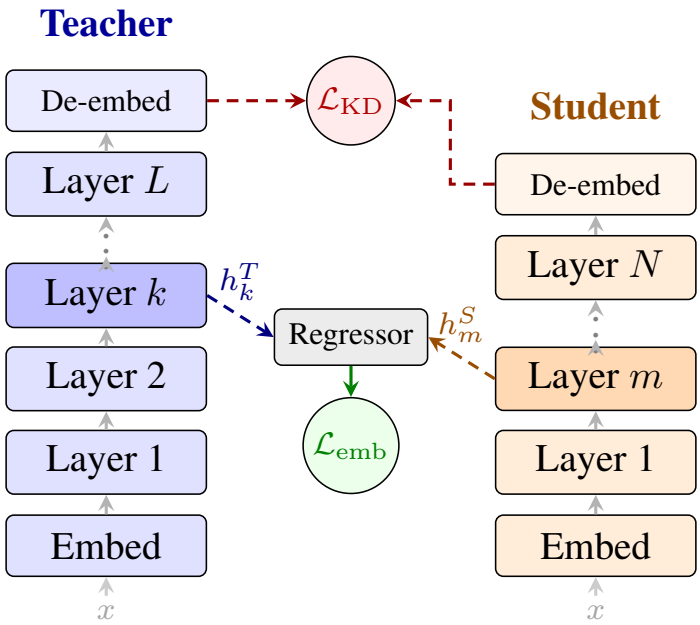

Figure 3: Overview of HLD. The student receives two training signals: $\mathcal{L}_{\text{KD}}$ matches the teacher's output logits, while $\mathcal{L}_{\text{emb}}$ aligns a student hidden state h_m^S with a teacher hidden state h_k^T through a learned regressor.

B.2. Hyperparameters

Table 2: Hyperparameter grid.

Student size	η	τ	α	γ	P_1
123M	$\{1, 4, 16\} \times 10^{-4}$	$\{0.5, 1\}$	$\{0.7, 0.9\}$	$\{0.1, 0.05\}$	$\{1\%, 5\%\}$
735M	$\{1, 4, 16\} \times 10^{-4}$	$\{0.5, 1\}$	$\{0.7, 0.9\}$	$\{0.1, 0.05\}$	$\{1\%, 4\%\}$

B.3. Downstream evaluation 123M

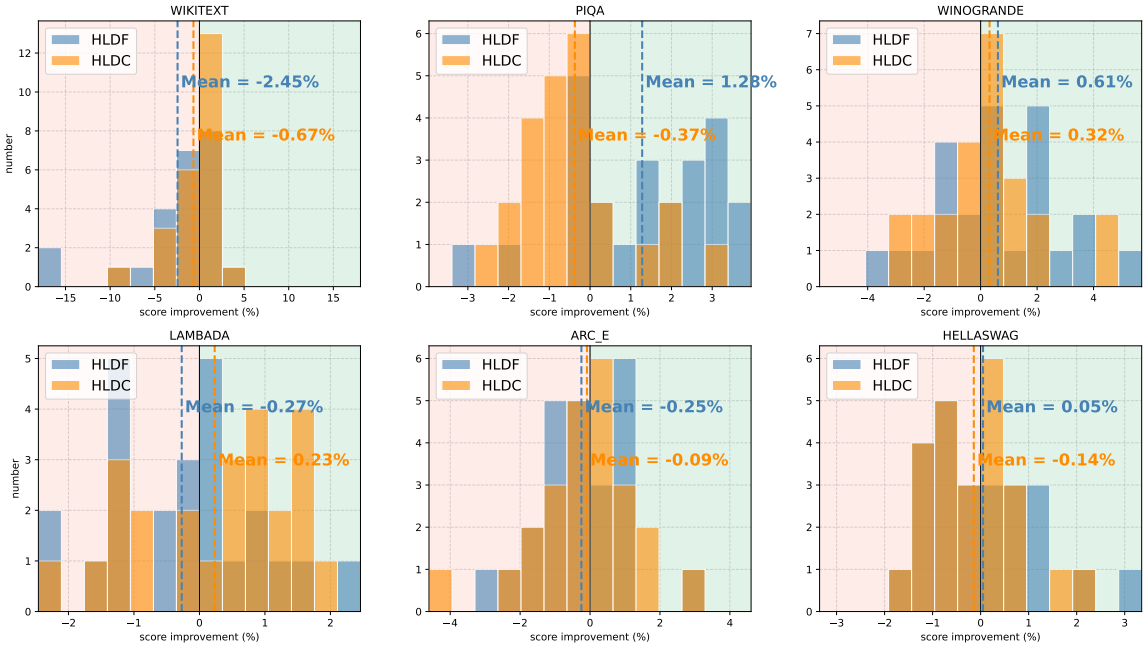

Figure 4: Shared-HP — downstream score improvement over KD for the **123M student**. Distribution of pointwise score improvements across all hyperparameter configurations on benchmarks. Green indicates HLD beats KD.

B.4. Scatter plots

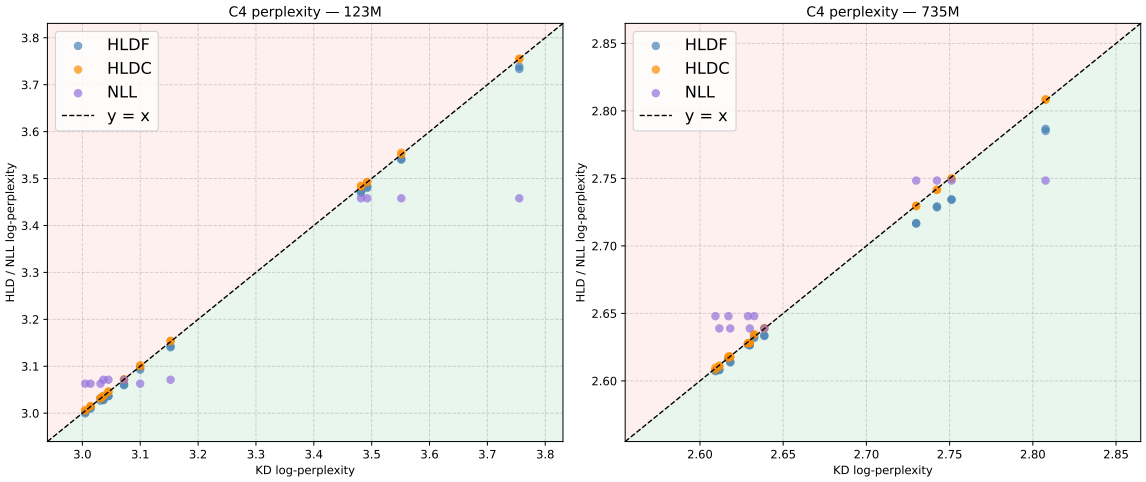

Figure 5: C4 perplexity improvement over KD. Scatter plot of pointwise score improvements across all hyperparameter configurations on C4 evaluation set. Each point is one hyperparameter set. For each point, the x axis is the KD log-perplexity associated with the hyperparameter set, the y axis is the HLD&NLL log-perplexity associated with the same hyperparameter set. The lower is always the better so the green zone is where HLD or NLL beats KD.

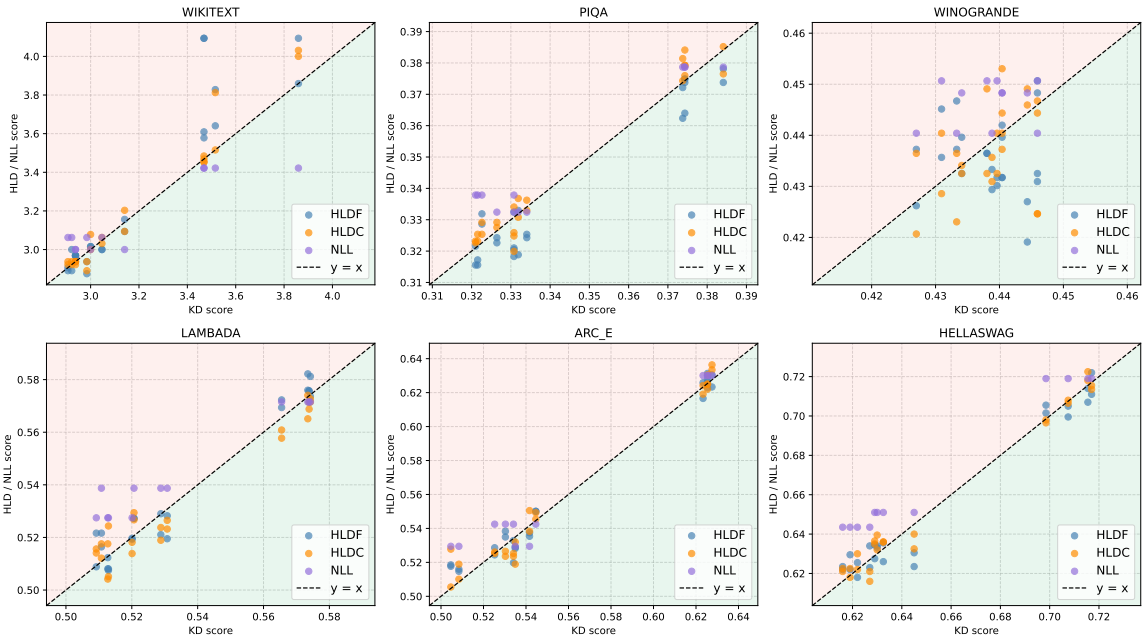

Figure 6: Shared-HP — downstream score improvement over KD for the **123M student**. Scatter plot of pointwise score improvements across all hyperparameter configurations on benchmarks. Each point is one hyperparameter set. For each point, the x axis is the KD score associated with the hyperparameter set, the y axis is the HLD&NLL score associated with the same hyperparameter set. The lower is always the better so the green zone is where HLD or NLL beats KD.

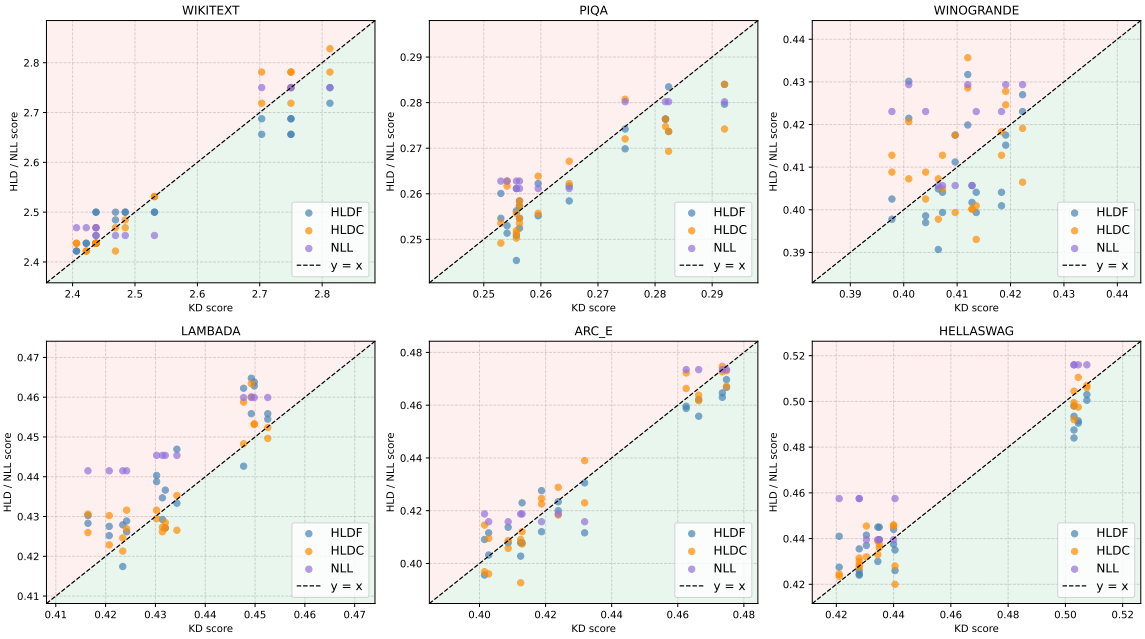

Figure 7: Shared-HP — downstream score improvement over KD for the **735M student**. Scatter plot of pointwise score improvements across all hyperparameter configurations on benchmarks. Each point is one hyperparameter set. For each point, the x axis is the KD score associated with the hyperparameter set, the y axis is the HLD&NLL score associated with the same hyperparameter set. The lower is always the better so the green zone is where HLD or NLL beats KD.

B.5. All results

Table 3: All training runs for the 123M student.

Method	η	P_1	α	τ	γ	C4	Wikitext	HSwag	Piqa	WinoG	Lambada	Arc-E
NLL	0.0001	0.00	–	–	–	3.458	3.422	0.719	0.379	0.451	0.572	0.630
NLL	0.0004	0.00	–	–	–	3.071	3.000	0.651	0.332	0.448	0.539	0.543
NLL	0.0016	0.00	–	–	–	3.063	3.062	0.643	0.338	0.440	0.527	0.529
KD	0.0001	0.00	0.7	0.5	–	3.551	3.516	0.708	0.374	0.440	0.574	0.625
KD	0.0001	0.00	0.7	1.0	–	3.481	3.469	0.717	0.384	0.438	0.573	0.623
KD	0.0001	0.00	0.9	0.5	–	3.755	3.859	0.699	0.374	0.431	0.565	0.628
KD	0.0001	0.00	0.9	1.0	–	3.492	3.469	0.716	0.374	0.446	0.574	0.625
KD	0.0004	0.00	0.7	0.5	–	3.072	3.000	0.645	0.331	0.444	0.531	0.530
KD	0.0004	0.00	0.7	1.0	–	3.036	2.938	0.630	0.334	0.440	0.521	0.534
KD	0.0004	0.00	0.9	0.5	–	3.152	3.141	0.632	0.332	0.434	0.511	0.545
KD	0.0004	0.00	0.9	1.0	–	3.045	2.938	0.629	0.326	0.440	0.529	0.525
KD	0.0016	0.00	0.7	0.5	–	3.031	2.922	0.627	0.331	0.427	0.520	0.535
KD	0.0016	0.00	0.7	1.0	–	3.005	2.984	0.622	0.321	0.433	0.513	0.505
KD	0.0016	0.00	0.9	0.5	–	3.100	3.047	0.616	0.323	0.446	0.513	0.542
KD	0.0016	0.00	0.9	1.0	–	3.014	2.906	0.619	0.322	0.439	0.509	0.508
HLDC	0.0001	0.00	0.7	0.5	0.05	3.551	3.812	0.708	0.384	0.433	0.571	0.625
HLDC	0.0001	0.00	0.7	0.5	0.10	3.555	3.516	0.707	0.376	0.440	0.569	0.625
HLDC	0.0001	0.00	0.7	1.0	0.05	3.482	3.469	0.716	0.385	0.449	0.574	0.619
HLDC	0.0001	0.00	0.7	1.0	0.10	3.485	3.453	0.714	0.376	0.433	0.565	0.624
HLDC	0.0001	0.00	0.9	0.5	0.05	3.756	4.000	0.697	0.374	0.440	0.561	0.636
HLDC	0.0001	0.00	0.9	0.5	0.10	3.755	4.031	0.698	0.381	0.429	0.558	0.633
HLDC	0.0001	0.00	0.9	1.0	0.05	3.493	3.484	0.723	0.379	0.447	0.573	0.630
HLDC	0.0001	0.00	0.9	1.0	0.10	3.492	3.453	0.718	0.379	0.444	0.572	0.622
HLDC	0.0004	0.00	0.7	0.5	0.05	3.072	3.078	0.640	0.326	0.446	0.523	0.524
HLDC	0.0004	0.00	0.7	0.5	0.10	3.072	3.000	0.632	0.334	0.449	0.526	0.527
HLDC	0.0004	0.00	0.7	1.0	0.05	3.037	2.922	0.639	0.333	0.440	0.529	0.524
HLDC	0.0004	0.00	0.7	1.0	0.10	3.037	2.938	0.632	0.336	0.437	0.527	0.525
HLDC	0.0004	0.00	0.9	0.5	0.05	3.154	3.094	0.636	0.331	0.433	0.512	0.549
HLDC	0.0004	0.00	0.9	0.5	0.10	3.154	3.203	0.636	0.337	0.434	0.518	0.545
HLDC	0.0004	0.00	0.9	1.0	0.05	3.046	2.938	0.636	0.329	0.453	0.524	0.525
HLDC	0.0004	0.00	0.9	1.0	0.10	3.045	2.938	0.635	0.328	0.444	0.519	0.526
HLDC	0.0016	0.00	0.7	0.5	0.05	3.032	2.938	0.616	0.320	0.436	0.514	0.519
HLDC	0.0016	0.00	0.7	0.5	0.10	3.031	2.922	0.621	0.325	0.421	0.518	0.532
HLDC	0.0016	0.00	0.7	1.0	0.05	3.005	2.938	0.630	0.323	0.423	0.505	0.505
HLDC	0.0016	0.00	0.7	1.0	0.10	3.007	2.891	0.622	0.323	0.436	0.524	0.528
HLDC	0.0016	0.00	0.9	0.5	0.05	3.100	3.031	0.623	0.325	0.425	0.504	0.538
HLDC	0.0016	0.00	0.9	0.5	0.10	3.102	3.062	0.621	0.329	0.425	0.518	0.551
HLDC	0.0016	0.00	0.9	1.0	0.05	3.015	2.922	0.618	0.323	0.431	0.516	0.519
HLDC	0.0016	0.00	0.9	1.0	0.10	3.015	2.938	0.623	0.325	0.436	0.514	0.510

Method	η	P_1	α	τ	γ	C4	Wikitext	HSwag	Piqa	WinoG	Lambada	Arc-E
HLDF	0.0001	0.01	0.7	0.5	–	3.540	3.828	0.705	0.375	0.430	0.576	0.628
HLDF	0.0001	0.05	0.7	0.5	–	3.542	3.641	0.700	0.364	0.432	0.576	0.623
HLDF	0.0001	0.01	0.7	1.0	–	3.473	4.094	0.722	0.378	0.436	0.576	0.617
HLDF	0.0001	0.05	0.7	1.0	–	3.470	3.578	0.711	0.374	0.436	0.582	0.626
HLDF	0.0001	0.01	0.9	0.5	–	3.739	4.094	0.706	0.372	0.436	0.569	0.623
HLDF	0.0001	0.05	0.9	0.5	–	3.733	3.859	0.702	0.362	0.445	0.572	0.630
HLDF	0.0001	0.01	0.9	1.0	–	3.482	4.094	0.707	0.375	0.448	0.581	0.631
HLDF	0.0001	0.05	0.9	1.0	–	3.480	3.609	0.714	0.374	0.451	0.574	0.626
HLDF	0.0004	0.01	0.7	0.5	–	3.061	3.016	0.630	0.320	0.427	0.528	0.535
HLDF	0.0004	0.05	0.7	0.5	–	3.060	3.016	0.624	0.332	0.419	0.520	0.538
HLDF	0.0004	0.01	0.7	1.0	–	3.029	2.969	0.633	0.324	0.442	0.527	0.520
HLDF	0.0004	0.05	0.7	1.0	–	3.028	2.969	0.634	0.325	0.432	0.527	0.533
HLDF	0.0004	0.01	0.9	0.5	–	3.143	3.156	0.635	0.333	0.440	0.522	0.550
HLDF	0.0004	0.05	0.9	0.5	–	3.140	3.094	0.626	0.319	0.433	0.516	0.550
HLDF	0.0004	0.01	0.9	1.0	–	3.037	2.969	0.634	0.323	0.440	0.529	0.524
HLDF	0.0004	0.05	0.9	1.0	–	3.036	2.969	0.627	0.324	0.432	0.521	0.529
HLDF	0.0016	0.01	0.7	0.5	–	3.032	3.000	0.623	0.321	0.426	0.519	0.528
HLDF	0.0016	0.05	0.7	0.5	–	3.027	2.891	0.634	0.318	0.437	0.520	0.529
HLDF	0.0016	0.01	0.7	1.0	–	3.002	2.938	0.618	0.316	0.447	0.508	0.518
HLDF	0.0016	0.05	0.7	1.0	–	3.000	2.875	0.625	0.322	0.437	0.507	0.519
HLDF	0.0016	0.01	0.9	0.5	–	3.097	3.000	0.624	0.332	0.431	0.512	0.535
HLDF	0.0016	0.05	0.9	0.5	–	3.093	3.000	0.622	0.329	0.433	0.508	0.537
HLDF	0.0016	0.01	0.9	1.0	–	3.012	2.906	0.622	0.317	0.433	0.522	0.516
HLDF	0.0016	0.05	0.9	1.0	–	3.010	2.891	0.629	0.316	0.429	0.509	0.515

Table 4: All training runs for the 735M student.

Method	η	P_1	α	τ	γ	C4	Wikitext	HSwag	Piqa	WinoG	Lambada	Arc-E
NLL	0.0001	0.00	–	–	–	2.748	2.750	0.516	0.280	0.429	0.460	0.473
NLL	0.0004	0.00	–	–	–	2.639	2.453	0.440	0.261	0.423	0.445	0.416
NLL	0.0016	0.00	–	–	–	2.648	2.469	0.458	0.263	0.406	0.441	0.419
KD	0.0001	0.00	0.7	0.5	–	2.751	2.750	0.503	0.282	0.419	0.448	0.475
KD	0.0001	0.00	0.7	1.0	–	2.730	2.750	0.504	0.282	0.412	0.450	0.466
KD	0.0001	0.00	0.9	0.5	–	2.808	2.812	0.503	0.292	0.401	0.453	0.473
KD	0.0001	0.00	0.9	1.0	–	2.742	2.703	0.507	0.275	0.422	0.449	0.463
KD	0.0004	0.00	0.7	0.5	–	2.618	2.484	0.430	0.260	0.398	0.434	0.419
KD	0.0004	0.00	0.7	1.0	–	2.612	2.469	0.434	0.256	0.404	0.430	0.409
KD	0.0004	0.00	0.9	0.5	–	2.639	2.531	0.435	0.265	0.414	0.431	0.432
KD	0.0004	0.00	0.9	1.0	–	2.630	2.438	0.440	0.256	0.418	0.432	0.403
KD	0.0016	0.00	0.7	0.5	–	2.617	2.422	0.441	0.254	0.413	0.424	0.413
KD	0.0016	0.00	0.7	1.0	–	2.609	2.406	0.428	0.256	0.407	0.423	0.402
KD	0.0016	0.00	0.9	0.5	–	2.633	2.438	0.428	0.256	0.406	0.421	0.424
KD	0.0016	0.00	0.9	1.0	–	2.629	2.438	0.421	0.253	0.410	0.416	0.412
HLDC	0.0001	0.00	0.7	0.5	0.05	2.750	2.781	0.498	0.276	0.428	0.459	0.467
HLDC	0.0001	0.00	0.7	0.5	0.10	2.750	2.781	0.504	0.275	0.425	0.448	0.473
HLDC	0.0001	0.00	0.7	1.0	0.05	2.729	2.719	0.497	0.274	0.429	0.453	0.464
HLDC	0.0001	0.00	0.7	1.0	0.10	2.730	2.750	0.510	0.269	0.436	0.453	0.462
HLDC	0.0001	0.00	0.9	0.5	0.05	2.809	2.781	0.499	0.274	0.407	0.450	0.473
HLDC	0.0001	0.00	0.9	0.5	0.10	2.808	2.828	0.492	0.284	0.421	0.452	0.475
HLDC	0.0001	0.00	0.9	1.0	0.05	2.742	2.719	0.507	0.272	0.406	0.463	0.472
HLDC	0.0001	0.00	0.9	1.0	0.10	2.741	2.781	0.506	0.281	0.419	0.460	0.466
HLDC	0.0004	0.00	0.7	0.5	0.05	2.618	2.484	0.446	0.264	0.413	0.427	0.423
HLDC	0.0004	0.00	0.7	0.5	0.10	2.618	2.469	0.432	0.256	0.409	0.435	0.425
HLDC	0.0004	0.00	0.7	1.0	0.05	2.611	2.422	0.439	0.252	0.403	0.432	0.406
HLDC	0.0004	0.00	0.7	1.0	0.10	2.611	2.469	0.433	0.256	0.409	0.429	0.409
HLDC	0.0004	0.00	0.9	0.5	0.05	2.639	2.531	0.438	0.267	0.401	0.427	0.423
HLDC	0.0004	0.00	0.9	0.5	0.10	2.639	2.531	0.436	0.262	0.393	0.426	0.439
HLDC	0.0004	0.00	0.9	1.0	0.05	2.628	2.438	0.446	0.255	0.413	0.427	0.410
HLDC	0.0004	0.00	0.9	1.0	0.10	2.628	2.469	0.446	0.254	0.418	0.428	0.396
HLDC	0.0016	0.00	0.7	0.5	0.05	2.617	2.422	0.420	0.263	0.400	0.432	0.407
HLDC	0.0016	0.00	0.7	0.5	0.10	2.618	2.422	0.428	0.262	0.400	0.427	0.412
HLDC	0.0016	0.00	0.7	1.0	0.05	2.608	2.438	0.430	0.251	0.405	0.425	0.415
HLDC	0.0016	0.00	0.7	1.0	0.10	2.610	2.438	0.431	0.250	0.413	0.421	0.397
HLDC	0.0016	0.00	0.9	0.5	0.05	2.634	2.438	0.428	0.257	0.407	0.423	0.429
HLDC	0.0016	0.00	0.9	0.5	0.10	2.635	2.438	0.427	0.258	0.398	0.430	0.418
HLDC	0.0016	0.00	0.9	1.0	0.05	2.628	2.438	0.423	0.249	0.418	0.426	0.393
HLDC	0.0016	0.00	0.9	1.0	0.10	2.628	2.438	0.424	0.254	0.399	0.431	0.409
HLDF	0.0001	0.01	0.7	0.5	–	2.734	2.656	0.498	0.276	0.415	0.443	0.467
HLDF	0.0001	0.04	0.7	0.5	–	2.734	2.688	0.493	0.276	0.418	0.462	0.470
HLDF	0.0001	0.01	0.7	1.0	–	2.716	2.656	0.490	0.283	0.420	0.463	0.462

Method	η	P_1	α	τ	γ	C4	Wikitext	HellaSwag	Piqa	WinoGrande	Lambada	Arc-E
HLDF	0.0001	0.04	0.7	1.0	–	2.717	2.688	0.491	0.274	0.432	0.464	0.456
HLDF	0.0001	0.01	0.9	0.5	–	2.787	2.750	0.487	0.284	0.430	0.454	0.463
HLDF	0.0001	0.04	0.9	0.5	–	2.785	2.719	0.484	0.280	0.421	0.456	0.465
HLDF	0.0001	0.01	0.9	1.0	–	2.729	2.688	0.500	0.270	0.427	0.456	0.460
HLDF	0.0001	0.04	0.9	1.0	–	2.729	2.656	0.503	0.274	0.423	0.465	0.459
HLDF	0.0004	0.01	0.7	0.5	–	2.614	2.500	0.442	0.262	0.398	0.433	0.428
HLDF	0.0004	0.04	0.7	0.5	–	2.614	2.500	0.437	0.255	0.403	0.447	0.412
HLDF	0.0004	0.01	0.7	1.0	–	2.608	2.500	0.430	0.256	0.399	0.439	0.414
HLDF	0.0004	0.04	0.7	1.0	–	2.608	2.484	0.445	0.251	0.397	0.440	0.408
HLDF	0.0004	0.01	0.9	0.5	–	2.634	2.500	0.440	0.258	0.399	0.429	0.431
HLDF	0.0004	0.04	0.9	0.5	–	2.633	2.500	0.445	0.262	0.404	0.435	0.412
HLDF	0.0004	0.01	0.9	1.0	–	2.626	2.500	0.438	0.255	0.401	0.427	0.412
HLDF	0.0004	0.04	0.9	1.0	–	2.626	2.500	0.444	0.252	0.404	0.437	0.403
HLDF	0.0016	0.01	0.7	0.5	–	2.617	2.438	0.426	0.251	0.402	0.429	0.423
HLDF	0.0016	0.04	0.7	0.5	–	2.617	2.438	0.435	0.253	0.406	0.426	0.408
HLDF	0.0016	0.01	0.7	1.0	–	2.607	2.422	0.426	0.261	0.404	0.428	0.409
HLDF	0.0016	0.04	0.7	1.0	–	2.608	2.422	0.424	0.245	0.399	0.417	0.396
HLDF	0.0016	0.01	0.9	0.5	–	2.633	2.453	0.435	0.257	0.391	0.428	0.420
HLDF	0.0016	0.04	0.9	0.5	–	2.632	2.438	0.424	0.258	0.405	0.425	0.423
HLDF	0.0016	0.01	0.9	1.0	–	2.627	2.438	0.427	0.255	0.418	0.430	0.403
HLDF	0.0016	0.04	0.9	1.0	–	2.627	2.438	0.441	0.260	0.411	0.428	0.408