Knowledge Distillation Framework for Cross-Architecture Text Encoders

Ozel Yilmazel
oyilmazel@umass.edu

Julia Epshtein
jepshtein@umass.edu

Phounthassen (Ricky) Lam
prlam@umass.edu

Abstract

Knowledge distillation has become a standard technique for compressing large teacher models into smaller, more efficient student models. However, existing knowledge distillation approaches are largely confined to same-architecture settings where teacher and student share an identical block structure and only different in depth, leaving the cross-architecture knowledge transfer underexplored. In this work, we propose a cross-architecture layer-wise distillation framework that ***transfers knowledge from a fine-tuned BERT teacher to a Funnel Transformer student****. Our training objective combines two losses: a task-specific cross-entropy loss and layer-wise CLS alignment loss. We train dataset-specific teachers and distill a separate student for each of three text classification benchmarks. We compare student performance against both the fine-tuned BERT teacher and the original pretrained Funnel Transformer reported in prior works. Finally, we conduct ablation studies on the loss function to isolate the contribution of each component to distillation performance, and propose a new loss to further improve task performance.*

Resources: GitHub Repository | Experiment Results & Visualizations

1. Introduction

Transformer based text encoders are fundamental to a wide range of NLP tasks, including semantic classification [4], sentence completion [7], and information retrieval [9]. The growing availability of strong pretrained models has reduced the need to pretrain from scratch, with experts increasingly relying on methods such as task-adaptive fine-tuning [8] or knowledge distillation [3] to adapt existing models to new domains. Knowledge distillation in particular has become a standard technique for compressing large teacher models into smaller, more efficient students. Common approaches distill through logit matching, intermediate hidden-state alignment, or margin-based objectives that encourage the student to preserve the teacher's ranking over positive and negative examples.

Among these, layer-wise distillation [11] leverages intermediate representations rather than only the final output. This technique has shown particular promise, as it provides a richer supervisory signal across the depth of the network. However, existing work on layer-wise distillation has largely focused on same-architecture settings, where the teacher and student share an identical block structure and differ only in size (ex. BERT [2] vs TinyBERT [5]), which allowed finer information distillation using attention weights. On the other hand, a cross-architecture setting, where the teacher and student follow broadly similar layered designs but differ in their intermediate operations, remains underexplored.

We investigate this gap by proposing a layer-wise distillation framework that transfers knowledge from a standard BERT-style encoder to a Funnel Transformer [1] student. The Funnel Transformer progressively reduces sequence length through pooling across blocks, making it architecturally distinct from BERT despite sharing the same underlying attention and layered progress. Crucially, we ask whether this cross-architecture distillation can yield strong models for downstream task data using only the teacher's layer-wise `[CLS]` representations and task loss as supervision, and what are the natural language understanding capabilities of these distilled models.

We evaluate our approach on 3 text classification benchmarks to compare with prior work (IMDB, Yelp, Amazon), and MNLI to measure natural language understanding of a distilled model. We investigate the following research questions:

- **RQ1:** Can cross-architecture layer-wise distillation from a fine-tuned BERT teacher enable an untrained Funnel Transformer student to achieve competitive classification performance?
- **RQ2:** What is the compression–accuracy tradeoff across student depths (2-2-2, 4-4-4, 8-8-8), and at what configuration do efficiency gains justify the accuracy cost?
- **RQ3:** What is the relative contribution of the layer-wise CLS alignment component?

2. Related work

2.1. Knowledge Distillation for Language Models

Knowledge distillation was introduced by Hinton et al. [3] as a general framework for transferring knowledge from a large teacher model to a smaller student by training the student to match the teacher's soft output distribution. In the context of language model compression, DistilBERT [10] demonstrated that distillation can be applied during pretraining to produce a general-purpose compressed model.

2.2. Layer-Wise Distillation

A limitation of output-only distillation is that it discards the rich intermediate representations learned across the depth of the teacher. Patient Knowledge Distillation (PKD) [11] addressed this by training the student to learn from multiple intermediate layers of the teacher, proposing learning from the last k layers or from every k-th layer. Both techniques improved from output-only distillation across several NLP benchmarks by providing a more granular and informative supervision signal.

TinyBERT [5] extended this idea further by distilling not only hidden states but also attention matrices at each layer, introducing a Transformer-specific distillation objective that aligns both the representational and the attentional behavior of the student with the teacher. Of the existing methods, TinyBERT is closest to our approach in its use of intermediate layer alignment. However, both PKD and TinyBERT assume a same-architecture setting in which teacher and student differ only in depth or width, and the layer correspondence is straightforward. Additionally, the attention weight distillation is very powerful, but inapplicable to our study, as our student model does not preserver the same sequence length throughout the model. Our work instead explores distilling across architectures with fundamentally different sequence-processing behavior.

2.3. Cross-Architecture Distillation

Cross-architecture distillation, where the teacher and student differ not just in scale but in structural design, has received limited attention. Liu et al. [6] recently explored this setting in computer vision, distilling knowledge from a Vision Transformer to a CNN. Because the two architectures produce structurally incompatible intermediate representations (spatial feature maps vs. token sequences), they introduce learned projectors to align features in a shared space rather than directly matching hidden states. While their work establishes that cross-architecture distillation is viable, their alignment mechanisms are specific to the vision domain and the CNN–Transformer gap. In NLP, however, the Transformer architecture is widely adapted and accepted, allowing us to relax the cross-architecture assumption for our study. Nevertheless, cross-architecture distillation in fundamentally different Transformers based models remains largely unexplored

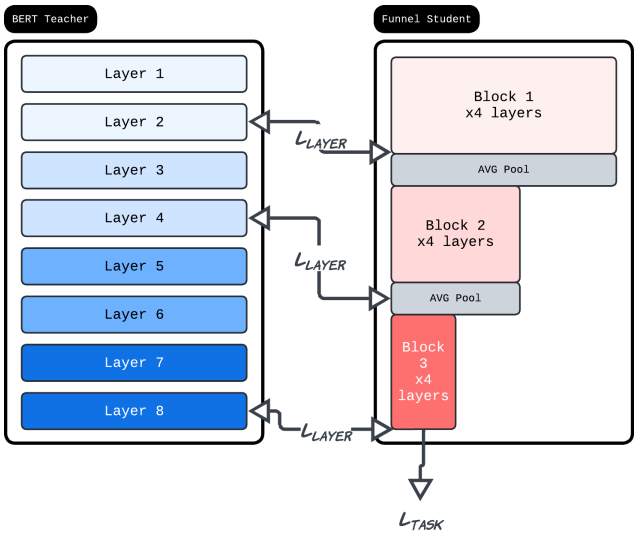


Figure 1. High-level diagram of our proposed framework. In this example, the student is separated in 3 blocks with 4 layers per block, the teacher is 8 layers. We use the `[CLS]` from layers 2-4-8 and task loss to distill the student model.

2.4. Funnel Transformer

The Funnel Transformer [1] improves upon the standard Transformer encoder by progressively compressing the sequence of hidden states across blocks through pooling, reducing computational cost while maintaining model capacity. The architecture is organized into blocks, where each block reduces the sequence length by half, creating a funnel-shaped structure. The first layer of block 2 and 3, use the last key and value of the previous block to do attention. This operation allows the model look back at information before compression and learn to compress the important values on its own. For sequence-level tasks such as text classification or indexing, this design is particularly efficient, as maintaining full token-level resolution throughout the network is unnecessary. The Funnel Transformer was originally pretrained using ELECTRA, in this work, we investigate whether this pretraining can be bypassed entirely through cross-architecture layer-wise distillation from a fine-tuned BERT teacher. For our study we chose the Funnel Transformer as our student model, as it is a fundamentally different architecture, while efficient and fast to train. For more information on the Funnel Transformer, we refer readers to the original study [1].

3. Method

3.1. Teacher Model Preparation

To provide a strong supervision signal for distillation, we first prepare a task-specific teacher model for each evaluation dataset. We fine-tune a pretrained BERT model by appending a linear classification head over the `[CLS]` token representation of the final layer and training on the task objective with cross-entropy loss. The resulting fine-tuned teacher M_T supplies the student with intermediate layer `[CLS]` representations to provide layer-wise alignment.

3.2. Student Model Architecture

We implement the Funnel Transformer student M_S in PyTorch, following the block-pooling design of Dai et al. [1]. The architecture consists of three sequential blocks of N Transformer layers each, with average pooling applied to the sequence dimension between blocks to progressively reduce the sequence length. In our implementation we use the same hidden dimension size for embeddings as the teacher model.

Each Transformer layer uses PyTorch's multi-head attention module. We initialize the student's token embeddings from the teacher's pretrained embedding matrix, which serves as a fixed lookup table after tokenization. This provides the student with a non-contextual starting representation. A linear classification head is appended over the final `[CLS]` token, identical in structure to the teacher's.

3.3. Layer-Wise Cross-Architecture Distillation

Our training objective combines two complementary loss terms designed to transfer knowledge from the teacher at multiple levels.

Layer-wise CLS loss. Inspired by PKD [11], we align the `[CLS]` representation at the end of each student block with the `[CLS]` representation at a corresponding teacher layer. Since the student and teacher share the same hidden dimension, we directly minimize the mean squared error between the paired `[CLS]` representations: $\mathcal{L}_{\text{layer}} = \frac{1}{|L|}\sum_{i=1}^{|L|}(\mathbf{c}_S^{(i)} - \mathbf{c}_T^{(m(i))})^2$, where $\mathbf{c}_S^{(i)}$ is the `[CLS]` representation at the end of student block i and $\mathbf{c}_T^{(m(i))}$ is the `[CLS]` representation at the mapped teacher layer.

Task-specific loss. We compute the standard cross-entropy loss between the student's predictions and the ground-truth labels: $\mathcal{L}_{\text{task}} = \text{CE}(\hat{y}_S, y)$.

Combined objective. The final training loss is a weighted combination of the two terms:

$$\mathcal{L} = \alpha\,\mathcal{L}_{\text{task}} + \gamma\,\mathcal{L}_{\text{layer}} \tag{1}$$

where α and γ control the relative contribution of each objective. For a better perspective on the training objective and model architecture, we refer readers to Figure 1.

3.4. Evaluation

We evaluate our distillation framework on three text classification benchmarks: Stanford's IMDB dataset, the Amazon Customer Review dataset, and the Yelp 2015 Challenge dataset. IMDB is a binary sentiment classification task, while Amazon and Yelp are more challenging two-class and five-class rating prediction tasks. We selected these datasets to ensure fair comparison against the original Funnel Transformer results. Finally, we also evaluate natural language understanding on the GLUE MNLI benchmark by using the distilled student on Yelp (5-class) as the initial checkpoint for training. We report the accuracy performance on each benchmark to ensure fair comparison to prior studies.

4. Experiments

4.1. Baselines

We compare our distilled students against two baselines: (1) a fine-tuned BERT-Medium model, which serves as the teacher and represents the performance upper bound for each dataset, and (2) the original Funnel Transformer results reported by Dai et al. [1], which were obtained after full pretraining followed by task-specific fine-tuning.

4.2. Experiment Settings

Student model weights are initialized randomly, the distillation procedure therefore trains the Funnel Transformer end-to-end from scratch. For all datasets we train for 4 epochs with a learning rate of 2e-5, weight decay of 0.01, the Adam optimizer, a warm-up ratio of 0.06, and a batch size of 32. BERT teacher fine-tuning uses the same hyper-parameters except for a reduced warm-up ratio of 0.01 to not change the pre-trained weights too much. For our experiments we set α and γ both to 1 for equal contribution to the total loss. When evaluating student models on MNLI, we used the final Yelp-5-class checkpoint of the corresponding configuration. This was done to measure whether a task-trained student can yield decent natural language understanding. Compared to IMDB and Amazon datasets, Yelp provided enough datapoints to train and experiment in a feasible amount of time. Additionally, the 5 class setting can provide more discriminative features compared to 2 class setting. Note that students are not distilled for the MNLI benchmark, but instead the Yelp-5 checkpoint is further fine-tuned using the train set with task loss. For fine-tuning students on MNLI, we used a smaller learning rate of 3e-5, while other settings stayed the same.

Table 1. Classification Accuracy Across All Benchmarks.

Model	IMDB	Yelp-2	Yelp-5	Amz-2	Amz-5	MNLI
Funnel-4-4-4 (Pretrained)	0.947	0.980	0.706	0.974	0.668	0.855
BERT-Medium (Teacher)	0.906	0.968	0.677	0.963	0.658	0.811
Student-2-4-8-L	0.854	0.947	0.635	-	-	**0.719**
Student-2-4-8-M	0.858	0.947	0.638	0.951	0.638	0.707
Student-2-4-8-S	0.855	0.947	0.637	-	-	0.682
Student-4-8-M	0.814	0.945	0.637	0.950	0.636	0.706

4.3. Main Findings

Table 1 reports classification accuracy for the fine-tuned BERT-Medium teacher, the pretrained Funnel Transformer baselines, and our distilled students across all benchmarks. The student suffix (L, M, S) denotes the number of layers per block (8, 4, 2, respectively), while the numeric prefix indicates the layer-mapping configuration used for distillation (for example 2-4-8 uses `[CLS]` of 2nd, 4th, 8th (last) layers of teacher to distill the student from the last `[CLS]` of each block).

RQ1: Can cross-architecture layer-wise distillation from a fine-tuned BERT teacher enable an untrained Funnel Transformer student to achieve competitive classification performance? Our distilled students, despite being initialized with random weights and trained entirely through cross-architecture distillation, achieve accuracy within approximately 5% of the fine-tuned BERT teacher across all classification benchmarks. On Yelp-2, all three student configurations reach 0.947 compared to the teacher's 0.968, a gap of just 2%. On Amazon-2, Student-2-4-8-M achieves 0.951 versus the teacher's 0.963. Although our results confirm that cross-architecture layer-wise distillation can serve as a viable method for producing task-specific models, it does not have the capacity to completely replace pretrained models. We can see that with substantial pretraining, the Funnel Transformer can outperform all student models, and the state-of-the-art BERT model across all benchmarks significantly.

RQ2: What is the compression-accuracy tradeoff across student depths, and at what configuration do efficiency gains justify the accuracy cost? Across the three hidden-dimension sizes, in-distribution performance remains stable, accuracy varies by less than 0.5% among the L, M, and S configurations on IMDB, Yelp-2, and Yelp-5. Amazon evaluations were conducted only for Student-M due to computational constraints. MNLI accuracy, however, shows greater sensitivity to model capacity, ranging from 0.682 for Student-2-4-8-S to 0.719 for Student-2-4-8-L. This suggests that additional capacity contributes more to cross-task generalization than to task performance.

We additionally compare Student-4-8-M, which maps distillation signals to two teacher layers (4th and 8th), against Student-2-4-8-M, which maps to three (2nd, 4th, and 8th). Student-2-4-8-M and Student-4-8-M achieve similar results across all benchmarks except IMDB. This gap suggests that denser layer-wise supervision provides a stronger training signal, particularly on smaller datasets such as IMDB where the model may otherwise lack sufficient gradient signal to learn low-level features.

4.4. Ablation Studies

RQ3: What is the relative contribution of the layer-wise CLS alignment component? To isolate the contribution of each loss component, we conduct ablation experiments on the Student-2-4-8-M configuration. Table 2 reports results under three settings: the full combined objective, a task-loss-only variant, and the full objective augmented with a pretext loss.

Effect of removing the layer-wise distillation signal. Removing the layer-wise CLS alignment loss has a modest effect on task performance, the impact on MNLI generalization, however, is substantially larger, with accuracy falling from 0.707 to 0.679. The t-SNE visualizations in Figures 4a and 4b support this result, the full-objective student produces noticeably cleaner class boundaries on the MNLI validation set, approaching the separation quality of the BERT teacher itself (Figure 4d), whereas the task-loss-only model yields more overlapping clusters. Together, these results suggest that the layer-wise distillation signal contributes primarily to transferable representations rather than task-specific performance.

The training dynamics under the task-loss-only configuration reveal a further insight. As shown in Figure 2b, the layer-wise loss (which receives no gradient signal in this configuration) initially spikes but begins to decrease after several epochs. This suggests that the student, trained solely on the task objective, eventually learns intermediate representations that partially resemble the teacher's, despite never being explicitly supervised to do so. Nevertheless, the

Table 2. Ablation Study Results for Student-2-4-8-M.

Configuration	IMDB	Yelp-2	Yelp-5	MNLI
Student-2-4-8-M	0.858	0.947	0.638	0.707
Task-loss only	0.856	0.942	0.628	0.679
+ Pretext loss	**0.864**	**0.948**	**0.639**	0.706

gap in MNLI accuracy confirms that this implicit alignment is insufficient to match the representational quality achieved through direct layer-wise distillation.

Effect of adding a pretext loss. To improve robustness and self-consistency, we introduce a pretext loss inspired by masked language modeling. During training, a random 10% of the input tokens are masked, and the model processes both the original and masked sequences. The pretext loss is defined as the KL divergence between the output logits of the two forward passes: $\mathcal{L}_{\text{pretext}} = D_{\text{KL}}\big(\hat{y}_S \parallel \hat{y}_S^{\text{mask}}\big)$, where $\hat{y}_S$ are the logits from the original input and $\hat{y}_S^{\text{mask}}$ are the logits from the masked input. This encourages the student to produce consistent predictions regardless of minor input perturbations. This objective augments the full distillation loss and does not replace either component. The total loss becomes $\mathcal{L} = \alpha\,\mathcal{L}_{\text{task}} + \gamma\,\mathcal{L}_{\text{layer}} + \beta\,\mathcal{L}_{\text{pretext}}$, where $\beta = 0.5$.

Adding the pretext loss yields the best in-distribution results across all benchmarks, with IMDB improving from 0.858 to 0.864 and Yelp-2 from 0.947 to 0.948. However, MNLI accuracy shows a negligible change (0.707 to 0.706), indicating that the self-consistency signal improves task-level robustness but does not further enhance cross-task generalization. The t-SNE visualization also shows a similar amount of clustering compared to Student-2-4-8-M in Figure 4c. Combined with the observation that MNLI accuracy scales with model capacity (Section 4.3), this suggests that broader natural language understanding in this distillation framework is driven primarily by model size rather than by additional training objectives. Another explanation for this result can be that MNLI input formats differ substantially from what the student model has seen in sentiment classification training.

4.5. Training Stability

To verify that our distilled students do not overfit during training, we plot training and validation accuracy over time for the Student-2-4-8-M configuration on the Yelp-5 dataset under two distillation settings. As shown in Figure 3, both the full combined objective and the task-loss-only configuration exhibit training and validation curves that closely track each other throughout training, with no significant divergence.

5. Conclusion

In this work, we introduced a cross-architecture layer-wise distillation framework that transfers knowledge from a fine-tuned BERT teacher to a Funnel Transformer student. Our approach combines a task-specific cross-entropy loss with a layer-wise CLS alignment loss, enabling an untrained student to acquire competitive classification performance without any pretraining. Across three text classification benchmarks (IMDB, Yelp, Amazon), our distilled students consistently achieved accuracy within approximately 5% of the BERT-Medium teacher, and within 2% on Yelp-2, despite being initialized entirely from random weights.

Our ablation studies revealed that the layer-wise CLS alignment signal contributes primarily to learning transferable representations rather than boosting task task accuracy. This was most evident in the MNLI generalization experiments, where removing the alignment loss caused a notable drop in accuracy, and the t-SNE visualizations confirmed that the full-objective student produces substantially cleaner class boundaries. Additionally, our proposed pretext loss, based on KL divergence between predictions on original and masked inputs, yielded the best classification performance across all task benchmarks, demonstrating that self-consistency regularization is a promising augmentation to the distillation objective.

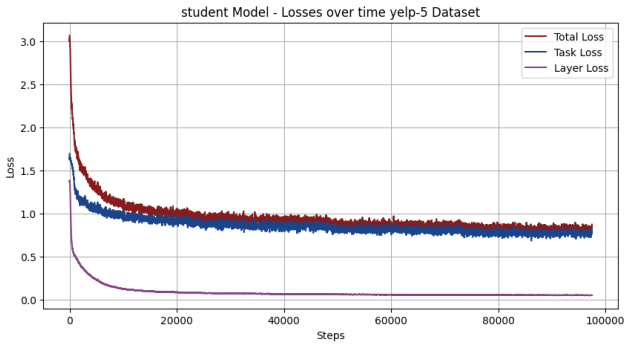
While our distilled students did not surpass the fully pretrained Funnel Transformer baseline, this outcome is expected given that pretraining leverages orders of magnitude more data and compute. Notably, the gap between our students and the BERT-Medium teacher was consistently small, suggesting that student performance may be bottlenecked by teacher quality rather than by the distillation framework itself. A stronger teacher, obtained through more extensive fine-tuning, larger model capacity, would likely yield corresponding improvements.

5.1. Limitations

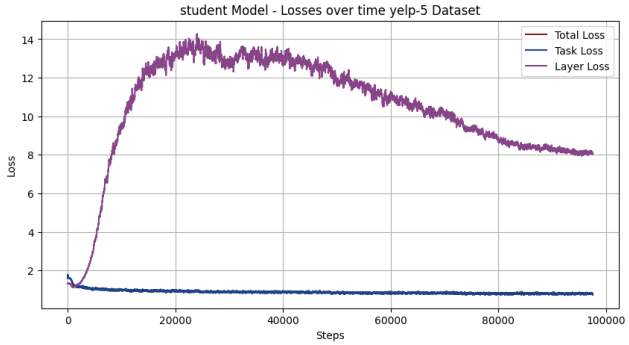
We acknowledge several resource constraints that limited the scope of our evaluation. Due to the size of the Amazon dataset, training a single model required over two days of compute, which prevented us from evaluating all student configurations (L, M, S) on the Amazon benchmarks. As a result, Amazon experiments were conducted only for the Student-2-4-8-M and Student-4-8-M configurations. With additional computational resources, a more comprehensive comparison across all model sizes on Amazon would strengthen the generality of our findings.

5.2. Future Work

Several promising directions remain for future investigation. First, our framework can be naturally extended to other cross-architecture settings beyond the BERT-to-

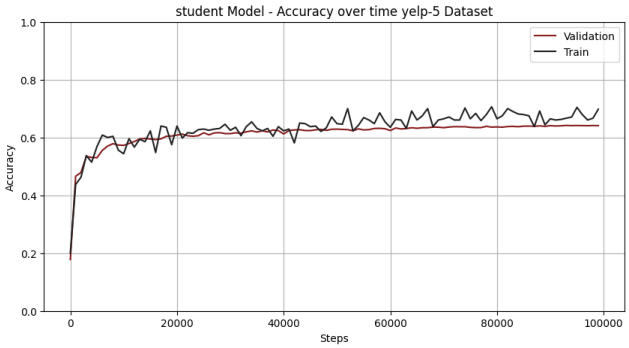


(a) Full combined objective on Yelp-5.

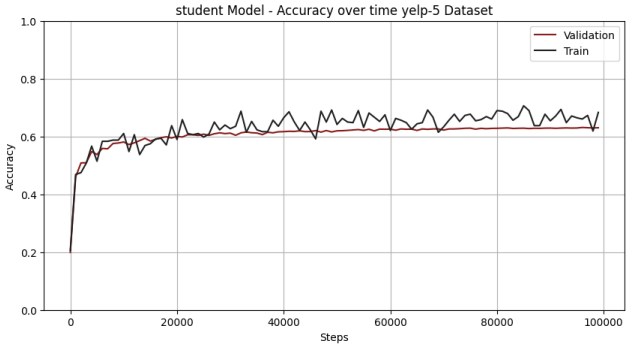


(b) Task-loss-only configuration on Yelp-5. The layer-wise loss receives no gradient but is tracked for analysis.

Figure 2. Training loss curves for Student-2-4-8-M under two distillation configurations.



(a) Full combined objective on Yelp-5.



(b) Task-loss-only configuration on Yelp-5.

Figure 3. Training and validation accuracy curves for Student-2-4-8-M on Yelp-5. Both configurations show closely aligned training and validation accuracy, indicating no overfitting.

Funnel pairing explored here. In particular, slim text encoders that collapse the entire input sequence to a single embedding throughout the network, rather than preserving token-level representations, represent an interesting target for distillation. Applying our layer-wise CLS alignment strategy to such architectures would test the generality of the framework under even more aggressive structural differences. More broadly, systematically evaluating this distillation approach across a diverse set of architecturally distinct encoders would help establish the true scope and significance of cross-architecture knowledge transfer as a practical alternative to pretraining.

References

[1] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V. Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing, 2020. 1, 2, 3

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 1

[3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 1, 2

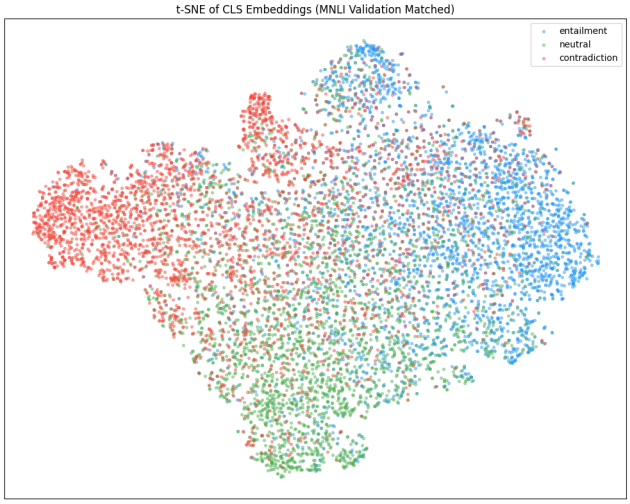
[4] Md Touhidul Islam, Farjana Parvin, Saad Ahmed Sazan, and Tariq Bin Amir. Comparative analysis of sentiment classification on imdb 50k movie reviews: A study using cnn, lstm, cnn-lstm, and bert models. In *2024 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON)*, pages 512–517, 2024. 1

[5] 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 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online, 2020. Association for Computational Linguistics. 1, 2

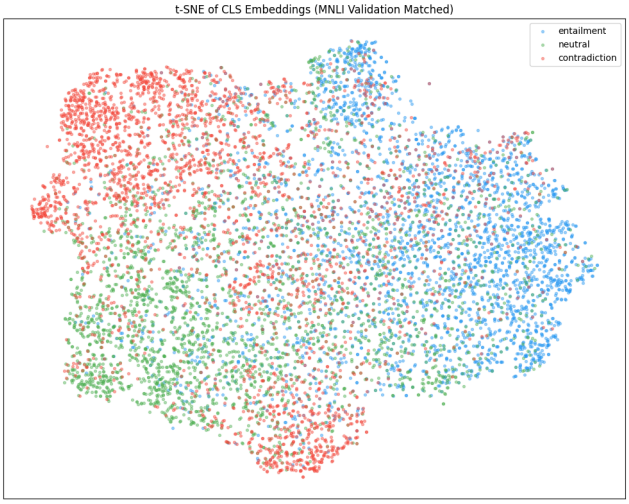
[6] Yufan Liu, Jiajiong Cao, Bing Li, Weiming Hu, Jingting Ding, and Liang Li. Cross-architecture knowledge distillation, 2022. 2

[7] Heewoong Park and Jonghun Park. Assessment of word-level neural language models for sentence completion. *Applied Sciences*, 2020. 1

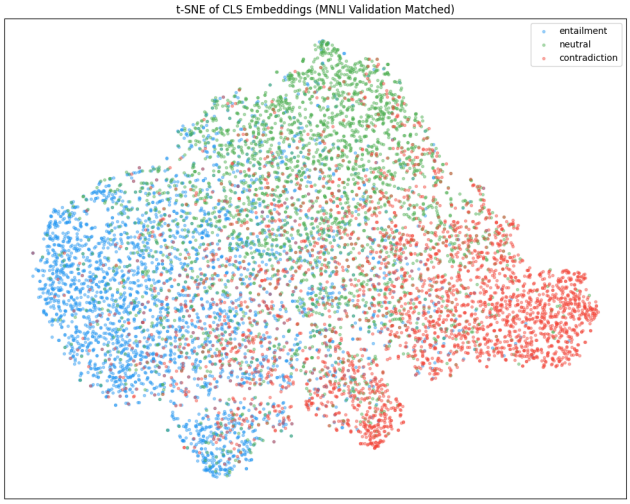
[8] Matthew E Peters, Sebastian Ruder, and Noah A Smith. To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, 2019. 1



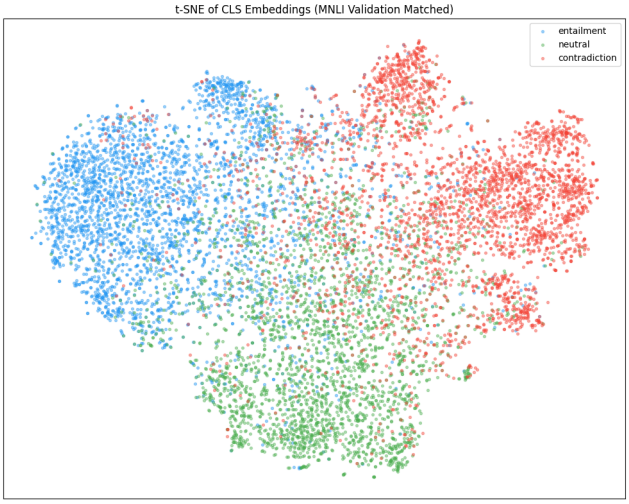
(a) Student-2-4-8-M (full objective).



(b) Student-2-4-8-M (task-loss only).



(c) Student-2-4-8-M (+ pretext loss).



(d) BERT-Medium teacher.

Figure 4. t-SNE visualizations of [CLS] embeddings on the MNLI validation set across model configurations. The full-objective student (a) and pretext-augmented student (c) exhibit similar class separation, while the task-loss-only student (b) shows more diffuse clusters. The teacher (d) provides the reference decomposition.

[9] Thilina C. Rajapakse. Dense passage retrieval: Architectures and augmentation methods. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 3494, New York, NY, USA, 2023. Association for Computing Machinery. 1

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. 2

[11] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China, 2019. Association for Computational Linguistics. 1, 2, 3