
Learning Task-Agnostic Representations through Multi-Teacher Distillation

Philippe Formont*

Universite Paris-Saclay- ETS Montreal
Mila - Quebec AI Institute
LIVIA- ILLS

Maxime Darrin*

McGill University- Universite Paris-Saclay
Mila - Quebec AI Institute
ILLs

Banafsheh Karimian*

ETS Montreal
ILLs - LIVIA

Jackie CK Cheung

McGill University
Mila - Quebec AI Institute

Eric Granger

ETS Montreal
ILLs - LIVIA

Ismail Ben Ayed

ETS Montreal
ILLs - LIVIA

Mohammadhadi Shateri

ETS Montreal
LIVIA

Pablo Piantanida

CNRS - CentraleSupélec - Universite Paris-Saclay
ILLs - Mila - Quebec AI Institute

Abstract

Casting complex inputs into tractable representations is a critical step across various fields. Diverse embedding models emerge from differences in architectures, loss functions, input modalities and datasets, each capturing unique aspects of the input. Multi-teacher distillation leverages this diversity to enrich representations but often remains tailored to specific tasks. In this paper, we introduce a task-agnostic framework based on a “majority vote” objective function. We demonstrate that this function is bounded by the mutual information between student and teachers’ embeddings, leading to a task-agnostic distillation loss that eliminates dependence on task-specific labels or prior knowledge. Our evaluations across text, vision models, and molecular modeling show that our method effectively leverages teacher diversity, resulting in representations enabling better performance for a wide range of downstream tasks such as classification, clustering, or regression. Additionally, we train and release state-of-the-art embedding models, enhancing downstream performance in various modalities.

1 Introduction

Transforming complex inputs into tractable representations is crucial for numerous applications across different domains, from natural language processing (Li & Li, 2023; Pimentel et al., 2023), computer vision (Kubota et al., 2024; Bhalla et al., 2024) to bioinformatics (Morgan, 1965; Wang et al., 2022a). This is done using embedders, often large pretrained models (Touvron et al., 2023; Jiang et al., 2023), that project objects (image, text, molecules, ...) into numerical representations, enabling various downstream tasks (Murphy, 2013; Vilnis & McCallum, 2015).

Variations in model architecture, training paradigms (e.g., unsupervised vs. supervised), and objective functions (e.g., masked language modeling and contrastive learning) result in embedders that capture different aspects of the same input. To leverage this diversity, a common practice is to combine them into a single model through multi-teacher Knowledge Distillation (KD) (Zhang et al., 2023).

*Equal Contribution

Not only are these methods cost-effective at inference time (Hinton et al., 2015; Frosst & Hinton, 2017), they are also extremely useful to compress knowledge from larger models into smaller ones for resource-constrained environments (Pan et al., 2022; Wang et al., 2023; Zhang et al., 2023), or mend the weights of models whose architectures have been altered (Muralidharan et al., 2024). Most existing approaches, however, focus on single-task distillation. In this setting, the student model either learns to mimic teacher representations for a specific task (Dvornik et al., 2019), or the distillation process is explicitly paired with task-specific information. While effective, such methods cannot be used for or generalized to unseen tasks, requiring a new distillation process to be performed for every new task. **Our goal is to learn a highly informative representation that retains maximal utility across a wide range of downstream tasks.** In other words, we aim to maximize information density within a single representation, enabling general-purpose adaptability without sacrificing performance.

Task-agnostic multi-teacher distillation aims to compress teacher representations into a single student embedder, such that the student representation captures as much information as all the teachers combined. To our knowledge, few works address task-agnostic distillation from multiple teachers. Existing approaches often rely on mean squared error (MSE) loss and cross-encoder heads (Navaneet et al., 2022), which can be unstable in high-dimensional spaces (Farebrother et al., 2024).

To overcome these limitations, we introduce a novel task-enabling setting to task-agnostic multi-teacher distillation. Our goal is to develop representations that capture the maximum amount of information about the data distribution, ensuring their applicability to a wide range of tasks, even in the absence of prior knowledge about those tasks. We train the student model to learn representations that, when applied to downstream tasks, generate predictions consistent with the majority of predictions from the teachers’ representations. This approach allows our method to leverage the collective knowledge of the teachers’ ensemble. To achieve this, we introduce an ensembling loss that measures the agreement between the Bayesian predictor based on the student’s embeddings and the Bayesian predictors based on the teachers’ embeddings. We show that this loss can be bounded independently of the task, using the conditional differential entropy of the teachers’ embeddings given the student’s output, thus providing a task-agnostic student-teacher reconstruction loss.

Contributions. In this study, we investigate the following research question: **How can the knowledge from multiple large embedding models be effectively distilled and integrated into a smaller one to produce a more general-purpose representation?** Our main contributions are threefold:

1. **A task-enabling setting.** We frame the multi-teacher distillation problem in a task-enabling setting, in which we study the relationship between the Bayes classifiers obtained from the students and the teachers’ embeddings. We prove a simple, yet powerful result: the conditional entropy of the teachers given the student’s output controls the probability of the student’s Bayesian predictor disagreeing with the teachers’ for any task.
2. **A tractable implementation.** We leverage a recent differentiable high-dimensional Gaussian-Mixture based estimator of the differential conditional entropy to formulate an information-theoretic loss. This loss maximizes the mutual information between the student and all teachers, resulting in a principled, task-agnostic distillation objective.
3. **High-quality generalized embedders.** Our method enhances distillation capabilities across three application domains: molecular modeling, natural language processing and computer vision. We release trained students achieving competitive performance on a wide range of downstream tasks, e.g., classification, regression, clustering, and sentence similarity.

2 Related Work

Task-oriented distillation. KD is widely used for transferring knowledge from one or a set of teachers to a student model (Gou et al., 2021) to improve the performance of the student on a given task (Zhang et al., 2019; Yim et al., 2017). This is typically done by transferring logits (Sun et al., 2024); *i.e.* the models’ output, features (Wang et al., 2023; Sarkar & Etemad, 2024), relational information (Dong et al., 2024, 2021), or a mixture of them (Liu et al., 2021a). Similarly, (Qiu et al., 2024) uses a regularization term to distill the task-relevant information from the large teacher to the small student. We depart from these methods by focusing on distilling task-agnostic representations.

Task-oriented multi-teacher distillation. A common method for multi-teacher KD is averaging the teachers’ logits and transferring the result to the student (Dvornik et al., 2019; Hinton et al., 2015).

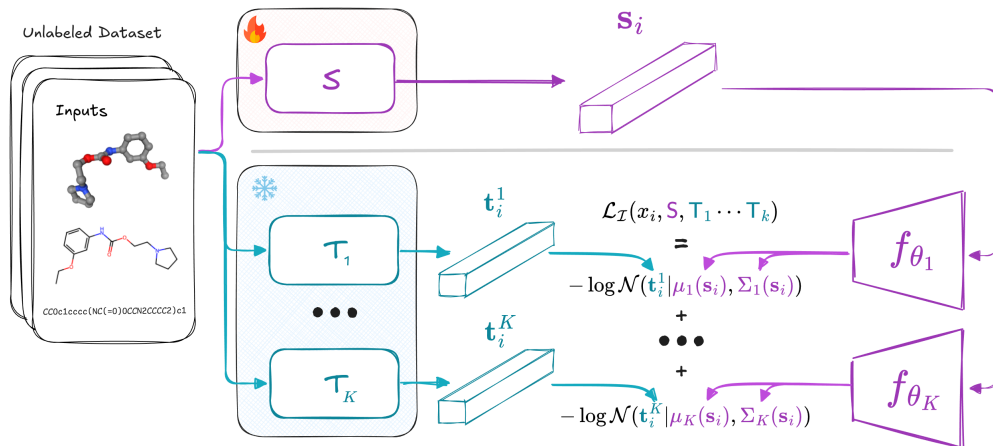


Figure 1: **Unsupervised training of our student through task-agnostic distillation.** The student embedder S is trained to minimize the negative log-likelihood of multiple teachers’ outputs conditioned on the student’s predictions. During this multi-teacher distillation procedure, both the student’s weights and those of the teacher-specific Gaussian kernels $\{f_{\theta_k}\}_{k \leq K}$ are updated in an end-to-end fashion. Post-training, we discard the Gaussian kernels and evaluate the student embedders by freezing their weights and training a feed-forward network on their embeddings for an unseen dataset.

However, this approach is not ideal when the performance of the teachers is uncertain. Alternative methods include using gate networks (Zhu et al., 2020), reinforcement learning agents (Yuan et al., 2020), and other methods (Ma et al., 2024a; Borza et al., 2022; Zhang et al., 2023) to perform teacher selection or evaluation. Due to challenges in distilling knowledge among diverse architectures, multi-teacher KD research mainly focuses on logit distillation. Other techniques were also explored, such as multi-teacher feature ensemble (Ye et al., 2024), contrasting feature distillation (Li et al., 2024), and cosine similarity-based methods for various tasks (Ma et al., 2024b; Aslam et al., 2024, 2023). Ensemble-based methods have also been proposed to mitigate over-smoothing and leverage teacher diversity, such as by aggregating structured predictions before distillation (Shayegh et al., 2024). Although successful, most multi-teacher feature distillation methods remain oriented to only one or a few tasks.

Task-agnostic and self-supervised features distillation. To the best of our knowledge, few works address task-agnostic representation distillation. Several approaches assume strong limitations, such as requiring the student to have the same architecture as the teachers (Liang et al., 2023; Xu et al., 2022b), or requiring fine-tuning the teachers to then distill their representations (Liu et al., 2023). Other methods induce requirements on the students, limiting their extension to a general multi-teacher setting. Notably (Gao et al., 2022) relies on vision-specific data augmentation, RoB (Duval et al., 2023) focuses on the distillation of joint-embedding approaches, AttnDist (Wang et al., 2022b) is only applicable to single teacher, (Song et al., 2023) need the teacher and student to have the same architecture, and SEED (Fang et al., 2021) requires the student and the teacher to have the same embedding dimension. Finally, CompRESS (Abbasi Koohpayegani et al., 2020) introduced a distillation method ensuring that the embeddings of the student and the teacher encode a similar nearest-neighbor graph, which would be unstable in a multi-teacher setting. Other approaches such as contrastive learning (Feng et al., 2024; Liu et al., 2022; Xu et al., 2022a) focus on distilling relational relationships between the samples, such as nearest neighbors preservation (Noroozi et al., 2018) or angle preserving distillation (Park et al., 2019a). SimReg (Navaneet et al., 2022), however, trains the student jointly with cross-encoding heads to directly reconstruct the teacher’s features using an MSE loss.

Interval estimation. While SimReg performs its distillation through pointwise estimation with MSE, it is well known in the reinforcement learning literature that these standard regression methods are difficult to train (Farebrother et al., 2024). On the other hand, replacing traditional regression scheme by maximum-likelihood training of Gaussian kernels appears to be more stable (Stewart et al., 2023) and effective in Value learning (Bellemare et al., 2017). We extend this idea in the

context of embedder distillation by using Gaussian kernels to estimate the conditional distribution of the teachers' embeddings given the student embedding and show that it is directly connected to maximizing the mutual information between the student and the teacher.

3 Distilling Representation Through Gaussian Kernels

We denote the input space by $\mathcal{X}$ and the corresponding input distribution by $P_{\mathbf{X}}$. We assume we have access to a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$, where samples are drawn i.i.d. according to $P_{\mathbf{X}}$. We consider a set of K different teacher embedders, $\mathbf{T}_k : \mathcal{X} \rightarrow \mathbb{R}^{d_k}$, for $k \in \{1, \dots, K\}$, each mapping inputs to potentially different embedding spaces of dimension d_k .

3.1 From a task-oriented setting to a task-agnostic loss

Our goal is to train a representation model capable of effectively handling any downstream task, by leveraging diverse representations from diverse pretrained teachers (Figure 1). To do so, we first measure the agreement between the student's Bayes classifier and the teachers' for any given task. First, we demonstrate that it can be bounded by the conditional entropy of the teacher's embedding given the student's, which does not depend on the considered task.

Let us consider a task characterized by a target set $\mathcal{Y}$ of discrete concepts and the feature space $\mathcal{X}$ with joint probability measure $P_{\mathbf{Y}\mathbf{X}} \in \mathcal{P}(\mathcal{Y} \times \mathcal{X})$. For every projection of the features through the different teachers, the Bayes decision rule is given by $c_{\mathbf{T}_k}^* \triangleq \arg \max_{c: \mathbb{R}^{d_k} \rightarrow \mathcal{Y}} \mathbb{E}_{\mathbf{X}, \mathbf{Y}} [\mathbb{1} [c(\mathbf{T}_k(\mathbf{X})) = \mathbf{Y}]]$ and

for the student: $c_S^* \triangleq \arg \max_{c: \mathbb{R}^d \rightarrow \mathcal{Y}} \mathbb{E}_{\mathbf{X}, \mathbf{Y}} [\mathbb{1} [c(\mathbf{S}(\mathbf{X})) = \mathbf{Y}]]$.

Our goal is to minimize the probability that the student's Bayesian classifier deviates from the predictions of the teachers'. This approach has been shown to enhance performance in most cases by reducing both bias and variance, while improving robustness and generalizability (Dietterich, 2000; Scimeca et al., 2023; Allen-Zhu & Li, 2020; Theisen et al., 2024). In other words, we aim to minimize the probability that the student's decision differs from that of each teacher:

$$\mathcal{L}^*(\mathbf{X}, \mathbf{Y}, \mathbf{S}, \mathbf{T}_1, \dots, \mathbf{T}_K) = \frac{1}{K} \sum_{k=1}^K \underbrace{\Pr(c_S^*(\mathbf{S}(\mathbf{X})) \neq c_{\mathbf{T}_k}^*(\mathbf{T}_k(\mathbf{X})))}_{\text{Probability that the student Bayesian classifier's output is different from the } k^{\text{th}} \text{ teacher's}}. \quad (1)$$

where the loss depends on the joint distribution $(\mathbf{X}, \mathbf{Y})$, through the definition of the Bayesian classifiers.

We leverage recent results on the performance of the Bayes classifiers to bound the probability of getting two different outcomes using the Bayes classifiers operating on two different projections of the input space.

Proposition 3.1 (Darrin et al. (2024)). *Let $C_{\mathbf{T}_k} = c_{\mathbf{T}_k}^*(\mathbf{T}_k(\mathbf{X}))$ and $C_S = c_S^*(\mathbf{S}(\mathbf{X}))$ denote the outcome of the Bayes classifier observing the output of the teacher $\mathbf{T}_k$ and the student $\mathbf{S}$ on a given task $\mathbf{Y}$, respectively.*

$$\Pr(C_S \neq C_{\mathbf{T}_k}) \leq 1 - \exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X})))$$

Corollary 3.2 (Training objective). *By applying Prop. 3.1 to Eq. 1 for any given joint distribution $P_{\mathbf{X}\mathbf{Y}}$, we have*

$$\mathcal{L}^*(\mathbf{X}, \mathbf{Y}, \mathbf{S}, \mathbf{T}_1, \dots, \mathbf{T}_K) \leq 1 - \exp\left(-\underbrace{\frac{1}{K} \sum_{k=1}^K h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))}_{\text{Negative log likelihood}}\right). \quad (2)$$

This corollary directly follows from the concavity of $t \rightarrow 1 - \exp(-t)$ (see Appendix A).

Remark 3.3. This bound over our ideal loss $\mathcal{L}^*$ is independent of the specific task and depends solely on the conditional entropy of the teacher embeddings given the student embeddings. Therefore, optimizing the student to minimize this loss provides a task-agnostic approach to aligning its Bayesian classifier predictions with the ensemble of teachers' predictions, regardless of the downstream task.

3.2 Student training

Estimation of the conditional entropy. To evaluate the conditional entropy of the teachers’ embeddings given the student’s embedding, we need a kernel to learn their conditional distribution $\hat{p}(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))$ as presented in Figure 1. To this end, we use a parametric Gaussian model whose parameters $\mu_k(\mathbf{S}(\mathbf{X}))$ and $\Sigma_k(\mathbf{S}(\mathbf{X}))$ are learned during the student’s training (Pichler et al., 2022).

Loss function. Following the above reasoning, we propose to train the student embedder $\mathbf{S}$ by minimizing the negative log-likelihood of the teachers’ embeddings given the student’s embedding, where the likelihood is estimated using Gaussian Kernels as follows:

$$\begin{aligned}\hat{\mathcal{L}}(\mathbf{X}, \mathbf{S}, \mathbf{T}_1, \dots, \mathbf{T}_K) &= \frac{1}{K} \sum_{k=1}^K h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X})) \\ &\leq \frac{1}{K} \sum_{k=1}^K \mathbb{E}_{\mathbf{X}} \left[-\log \mathcal{N}(\mathbf{T}_k(\mathbf{X}) \mid \mu_k(\mathbf{S}(\mathbf{X})), \Sigma_k(\mathbf{S}(\mathbf{X}))) \right], \quad (3)\end{aligned}$$

where $\mathcal{N}(\cdot|\mu, \Sigma)$ is the Gaussian distribution with mean μ and covariance Σ . In our setting, minimizing the conditional entropy $h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))$, exactly corresponds to maximizing the mutual information $I(\mathbf{T}_k(\mathbf{X}); \mathbf{S}(\mathbf{X})) = h(\mathbf{T}_k(\mathbf{X})) - h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))$ since for each teacher $h(\mathbf{T}_k(\mathbf{X}))$ is constant w.r.t of the student. This also applies to the bound in Eq. 2.

Training procedure. We train both the student and the different kernels in an end-to-end fashion by minimizing the loss function $\hat{\mathcal{L}}$. It boils down to minimizing the negative log-likelihood of the teachers’ embeddings given the student’s embedding. We use the Adam optimizer to minimize the loss function. See Appendix E for the detailed training algorithm. To reduce the computational cost, we first embedded the entirety of the training set using the teachers and store them. We can then build training batches by sampling from the pre-computed embeddings.

Baselines and Evaluation. We consider two widely used multi-teacher feature distillation methods, MSE, used in SimReg (Navaneet et al., 2022) and Cosine similarity (see Appendix G for more information). To evaluate the representations learned by the student, for each modality, we run different benchmarks evaluating its performance on a wide variety of downstream tasks. For classification and regression tasks, we train a small feedforward network on top of the embeddings (the backbones are considered frozen) on different tasks and evaluate its performance.

4 Text Embedders

4.1 Experimental setting

We focus on distilling high-performing and large models into significantly smaller ones. Indeed, modern models in NLP are extremely large and costly to train². Thus, we aim to produce the best possible models for a given weight category, pushing the size/performance of the Pareto frontier (Figure 2a), and not necessarily competing with the largest models. We distill from four teachers ranging from 433M parameters to 7B into students ranging from 20M to 335M parameters based on the nowflakes (Merrick et al., 2024) embedders.

Teachers and student. We select four freely available embedding models from the Huggingface hub (Wolf et al., 2020) (See Sec. C.1.2 for a detailed list of the teachers) whose evaluations are available in the MTEB benchmark (Muennighoff et al., 2023). To ensure having a point of comparison, we select teachers of different sizes and performances. Notably, SFR-Embeddings-R.2 is more than ten points stronger than the other three (smaller) teachers. As students we use snowflakes (Merrick, 2024; Merrick et al., 2024) models xs (22M), s (33M), m (109M) and l (335M) and we further train them using our distillation method (See Sec. C.1.4).

Embedder evaluation. Evaluating NLP models is notably challenging, and the common practice of evaluating a model using multi-task benchmarks may not be indicative of model capabilities (Liu et al.,

²<https://github.com/ills-montreal/nlp-distill>

Table 1: Performance of our distilled models compared to the strongest models of similar sizes from the MTEB Benchmark on classification tasks. Our 109M parameters model outperform significantly models 3 times bigger exhibiting exceptional information density.

	Task Model	Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTPDomain	MTPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
xs	Bas. GIST	23M	<u>72.9</u>	87.2	42.6	<u>84.2</u>	52.1	78.5	<u>94.8</u>	<u>77.7</u>	73.2	76.7	72.9	<u>59.9</u>	72.7
	Ivysaur	23M	72.1	<u>86.7</u>	42.7	81.9	45.4	80.8	92.1	71.9	70.3	74.9	65.5	58.7	70.2
	gte-tiny	23M	71.8	86.6	<u>42.6</u>	81.7	44.7	80.5	91.8	69.9	70.1	74.9	<u>71.0</u>	58.6	70.3
	MSE Student-xs	23M	71.6	86.2	42.3	83.6	<u>57.5</u>	83.5	94.5	75.4	<u>74.3</u>	80.4	66.3	59.3	<u>72.9</u>
s	NLL Student-xs	23M	76.5	84.9	42.4	85.8	58.0	<u>81.1</u>	95.2	79.9	75.8	80.4	68.1	60.1	74.0
	Bas. bge-small-en-v1.5	33M	73.8	92.8	47.0	85.7	47.8	90.6	93.4	74.8	74.8	78.7	<u>69.9</u>	60.5	74.1
	GIST	33M	75.3	<u>93.2</u>	<u>49.7</u>	<u>86.7</u>	55.9	89.5	<u>95.5</u>	79.1	75.5	79.2	72.8	<u>61.0</u>	76.1
	NoInstruct	33M	<u>75.8</u>	93.3	50.0	86.4	55.1	<u>90.2</u>	95.3	<u>79.6</u>	<u>76.0</u>	79.3	69.4	61.3	<u>76.0</u>
m	MSE Student-s	33M	72.6	90.3	44.3	84.2	<u>56.5</u>	88.8	94.9	77.2	75.4	81.2	64.9	60.4	74.2
	NLL Student-s	33M	77.3	89.2	43.8	86.7	58.0	88.3	95.5	81.9	76.7	<u>80.7</u>	66.1	60.6	75.4
	Bas. bge-base-en-v1.5	109M	76.2	<u>93.4</u>	<u>48.9</u>	87.0	51.9	90.8	94.2	76.9	76.2	80.2	71.6	59.4	75.5
	GIST	109M	76.0	93.5	50.5	87.3	54.7	<u>89.7</u>	95.3	78.1	76.0	79.6	72.4	59.3	76.0
l	e5-base-4k	112M	<u>77.8</u>	92.8	46.7	83.5	47.0	86.2	93.7	75.3	73.0	77.7	<u>72.1</u>	60.4	73.8
	e5-base-v2	110M	<u>77.8</u>	92.8	46.7	83.5	47.0	86.2	93.7	75.3	73.0	77.7	<u>72.1</u>	60.4	73.8
	MSE Student-m	109M	76.6	89.1	44.7	87.2	60.8	88.0	<u>95.7</u>	<u>81.6</u>	<u>77.7</u>	<u>82.2</u>	67.3	<u>60.5</u>	76.0
	NLL Student-m	109M	79.6	89.5	45.8	88.0	<u>59.7</u>	88.3	96.2	83.9	78.6	82.7	67.1	61.3	76.7
l	Bas. bge-large-en-v1.5	335M	75.8	92.4	48.2	87.8	51.5	92.8	94.6	79.5	77.6	80.5	70.9	59.9	76.0
	GIST	335M	75.6	93.4	49.1	88.1	54.7	91.2	<u>95.2</u>	78.2	76.2	79.3	71.9	59.2	76.0
	UAE-Large-V1	335M	75.5	92.8	48.3	87.7	51.8	92.8	94.0	76.9	76.5	79.8	71.1	59.8	75.6
	ember-v1	335M	76.1	92.0	47.9	<u>87.9</u>	52.0	92.8	94.6	79.3	77.4	80.5	71.4	60.0	76.0
l	mxbai-embed-large-v1	335M	75.0	93.8	49.2	87.8	50.9	92.8	94.0	76.8	76.2	80.0	<u>71.5</u>	59.7	75.6
	MSE Student-l	335M	<u>77.3</u>	84.5	43.4	86.0	<u>60.0</u>	82.7	95.1	<u>79.8</u>	76.3	<u>81.3</u>	65.8	<u>60.2</u>	74.4
	NLL Student-l	335M	81.5	88.1	45.9	86.9	60.4	88.2	95.6	83.2	<u>77.5</u>	81.4	67.7	62.2	76.5

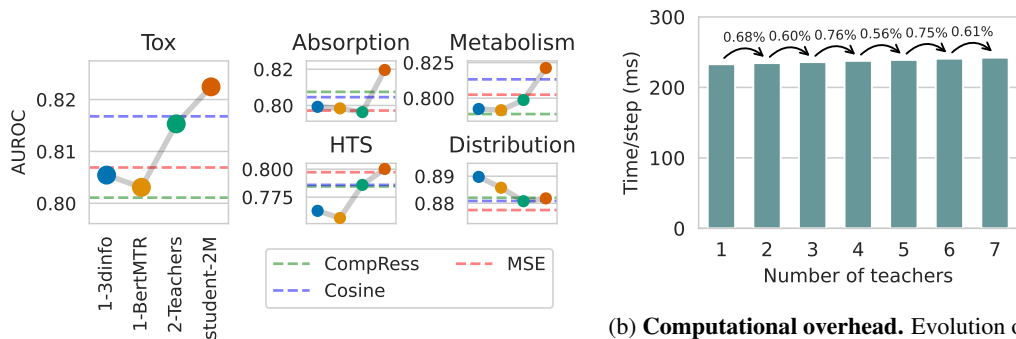
2024). For lack of better options and because it is currently the most widely accepted benchmark, we rely on the evaluation provided by the MTEB benchmark (Muennighoff et al., 2023) on 33 tasks encompassing clustering (11 datasets), sentence similarity (10 datasets) and classification tasks (12 datasets). We compare our models with distilled and non-distilled ones from the MTEB leaderboard.

Training set. We gathered different common datasets used for training embedders and collected 6 million entries from the Huggingface Hub, including Specter (Cohan et al., 2020), T5 (Ni et al., 2021), Amazon QA (McAuley & Leskovec, 2013), IMDB (Maas et al., 2011), SNLI (Bowman et al., 2015), QQP triplets from Quora, AG News (Zhang et al., 2015), MEDI dataset (Su et al., 2023) and the DAIL Emotion dataset (Saravia et al., 2018). We provide the dataset statistics in Sec. C.1.1. The datasets are all flattened, such that if the original had two columns (e.g., sentence 1 and 2 in the SNLI dataset), we end up with twice the number of entries, one for each sentence, and we deduplicated the dataset. Models are trained for two epochs with batch size 16 on NVIDIA V100.

4.2 Distillation performance

Task performance. Our method produces models that exhibit strong performance on a large variety of tasks, ranking first amongst all models of similar size in the MTEB benchmark on most of the tasks (Figure 2b). Notably, we observe that our method produces models that are competitive for almost all the tasks, whereas other models appear more specialized. We provide the actual accuracy of our models on classification tasks in Tab. 1. We provide the full results for all model sizes in Sec. C.2.1.

Pareto frontier. Our goal with distillation is to increase information density of models to reduce computational costs and memory footprint, we show in Figure 2a that our method can pack more information into fixed-size models. Interestingly, our medium-sized model (109M parameters) outperforms all the models three times its size and even our 335M model under the same training setting. In addition, our small models outperform all previous model of their weight category, notably yielding a 2-point gain on average classification accuracy on the MTEB over the previous *state-of-the-art* efficient GIST-based embedders (Solatorio, 2024).



(a) **Number of teachers and classification performances.** Performance of students trained with 1 teacher “1-”, 2 teachers, and 8 teachers (student-2M). We compare our NLL distillation method to MSE, Cosine, and CompRes for eight teachers.

(b) **Computational overhead.** Evolution of runtime for a training step as a function of the number of teachers. The computational overhead induced by an additional teacher represents less than 1% of the total runtime on a batch.

the teachers with (t) such as ChemBERTaMTR $^{(t)}$, and use a 2D-GNN (Graph Isomorphism Network: GIN (Hu et al., 2020)) for our student (for more details see Sec. B.1)³.

Evaluation setting. We evaluated all models on the ADMET (Absorption, Distribution, Metabolism, Excretion, Toxicity) tasks of the Therapeutic Data Commons platform (TDC) (Huang et al., 2021) and on a high-throughput screening task (HTS), (HIV (Wu et al., 2018)). We record the test performance over five runs (details on the evaluation procedure in Sec. B.3). We trained our models on MOSES, a processed version of the ZINC Clean Leads dataset (Polykovskiy et al., 2018), containing 2 million samples, and on ZINC-250k (Irwin & Shoichet, 2005), consisting of 250,000 samples. The performances of the model trained on 250k samples can be found in Sec. B.1. Both are public datasets of commercially available compounds designed to be used in various therapeutic projects.

5.2 Results

Overall performance. We compare the performance of the student model with the teachers and other baseline embedders on the different tasks. The results (average rank) for each task are presented in Figure 3. Our student model achieves the best performance on both the regression and classification tasks, delivering the most accurate predictions across a majority of tasks. This suggests that our method generates informative representations, providing high-quality molecular descriptors.

Single teacher vs. multi-teachers. To assess the impact of training a student with multiple teachers, we trained students to distill the knowledge of a single teacher and two teachers, and compared the results to those of our student trained with eight teachers. We selected two of the best-performing baselines as teachers: ChemBERTaMTR-77M (Ahmad et al., 2022) and 3D-infomax (Stärk et al., 2021). We then trained student models on the 2M-molecules dataset. Figure 4a displays the performances of each of these student models on the regression tasks. Training with multiple teachers consistently outperforms training with a single teacher, except on the Blood-Brain Barrier (BBB) task (the only Distribution classification task), which is also one of the tasks our model struggles the most with. For the BBB benchmark, we noticed it is one of the datasets where all results are among the most tightly packed (variations within 1.45 times the average standard deviation of the results), and whose data distribution differs the most from the training set, which could explain the slightly lower average performance of the 8-teacher student compared to the 1 or 2-teacher students. Overall, using multiple teachers significantly improves performance, with the best performance achieved when training with all eight teachers (additional results are available in Sec. B.4).

Comparison to baselines. Figure 4a also compares the performance of our NLL distillation method to MSE, cosine, and CompRes distillation for eight teachers. Overall, in the evaluation of classification tasks, our NLL distillation method outperformed the Cosine and MSE distillation methods. This observation goes beyond the results of classification tasks, as we also observed that the NLL distillation method consistently outperforms the other two methods on all evaluated task categories (see Sec. B.1.3 for more details).

³<https://github.com/ills-montreal/mol-distill>

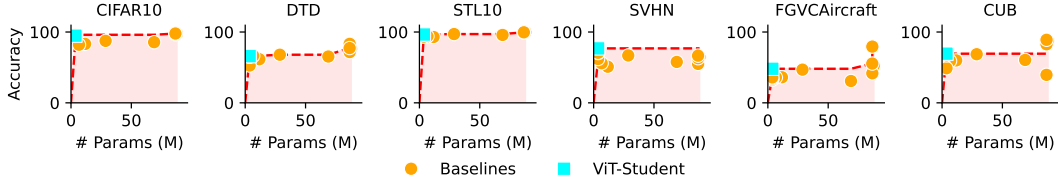


Figure 5: **Pareto frontier of vision models.** The figure compares the performance of student model distilled using our method (named ViT-Student shown with color blue) with baselines (shown in yellow) across various datasets. The distilled student consistently lies on the Pareto frontier.

Computational complexity. Training our molecular embedders on the largest dataset (2 M molecules) takes approximately 50 hours on 6 A6000 GPUs. We evaluated the computational overhead induced by the multi-teacher setting in Figure 4b. The runtime of a training step increases linearly with the number of teachers: $+1.57ms$ per teacher, representing less than 1% of the total runtime.

6 Image Embedders

For our final modality, vision, we aim to assess whether our method can deliver competitive performance compared to other baseline models (teachers, and MSE, Cosine, and CompRes student), especially on fine-grained vision classification tasks. In the following subsections, we outline the experimental setup used to investigate these questions and present the results. Additional details, including hyperparameter tuning and the augmentations applied, can be found in Sec. D.3.

6.1 Experimental setting

Teachers and evaluations. Given the increasing use of Vision Transformers, we used large transformer models (Swin (Liu et al., 2021b), DINOv2 (Oquab et al., 2023), ViT (Dosovitskiy et al., 2021), and BEiT (Bao et al., 2022), with around 87 million parameters) as teachers, and selected a smaller Vision Transformer, PVTv2 (Wang et al., 2022c), with 3.7 million parameters, as the student. We also use some CNN based modes with different sizes as baselines to have a more comprehensive comparison of our student’s representation abilities (refer to Sec. D.1 for more details).

Training set. We include fine-grained datasets such as DTD (Cimpoi et al., 2014), FGVCAircraft (Maji et al., 2013), and CUB (Welinder et al., 2010), alongside CIFAR10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011), STL10 (Coates et al., 2011) for the vision experiment. These allows us to assess the performance of our approach on a variety of challenging and detailed classification tasks. Refer to Sec. D.2 for details of the datasets meta-data⁴.

6.2 Results on Vision Transformer

To further evaluate our method, we conducted experiments using Vision Transformer (ViT) teachers. As shown in Figure 5, the distilled student model trained with our approach consistently lies on the Pareto frontier, for each task, showing a superior trade-off between accuracy and model size. Notably, our distilled student achieves the best performance among other distillation methods and other baseline models within its respective size categories, with results comparable to large ViT teachers ($20\times$ more parameters). This demonstrates our method’s ability to effectively transfer knowledge from large, complex teacher models to smaller, more efficient student models, while maintaining comparable performance. Additional results in Sec. D.4 show that our method generalizes well to unseen vision datasets, improving other distillation baselines, and effectively integrates diverse task-specific teachers without performance conflicts, confirming its robustness across domains.

⁴<https://github.com/ills-montreal/vision-distill/>

7 Limitations

Our method focuses on training student embedding models for diverse, unknown tasks; for single, pre-defined tasks, task-specific distillation may be more effective. As with any distillation approach—especially multi-teacher distillation—there is an overhead, either computational (if teacher embeddings are generated on-the-fly) or memory-intensive (if precomputed). We mitigate this by precomputing and storing embeddings, requiring approximately 100GB of disk space for our largest text-based teacher. The quality of our student embeddings depends on the relevance of the teachers to the downstream tasks. While task-specific teachers provide limited benefits outside their domain, they do not degrade performance when combined with task-relevant teachers (Sec. D.4). Our optimization metric maximizes mutual information between student and teachers but does not explicitly structure the embedding space, potentially limiting performance in tasks like clustering. For textual embeddings, we observe significant gains in classification (where embeddings train a small classifier) but more modest improvements in clustering and STS tasks, which rely on embedding dot products for similarity assessment (Sec. C.2.2).

8 Conclusions and Future Work

We proposed a theoretically grounded task-agnostic distillation mechanism that leverages interval estimation through Gaussian kernels in high dimensions to distill a more informative representation from multiple teachers to a single student. We demonstrated that our objective serves as a proxy for maximizing the mutual information and reconstructive capacity of the student model in relation to the teachers. We experimentally validated that our method is more efficient than point estimation-based multi-teacher feature distillation methods such as MSE or cosine-based distillation mechanisms. We demonstrated the superior performance of our method compared to others across three different modalities and numerous downstream tasks. In future work, we aim to extend this distillation approach to cross-modal distillation, enhancing the model’s capabilities by leveraging task-agnostic cross-modal information.

References

- Abbasi Koohpayegani, S., Tejankar, A., and Pirsiavash, H. Compress: Self-supervised learning by compressing representations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 12980–12992. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/975a1c8b9aee1c48d32e13ec30be7905-Paper.pdf.
- Ahmad, W., Simon, E., Chithrananda, S., Grand, G., and Ramsundar, B. Chemberta-2: Towards chemical foundation models, 2022.
- Allen-Zhu, Z. and Li, Y. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.
- Aslam, M. H., Zeeshan, M. O., Pedersoli, M., Koerich, A. L., Bacon, S., and Granger, E. Privileged knowledge distillation for dimensional emotion recognition in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3338–3347, 2023.
- Aslam, M. H., Pedersoli, M., Koerich, A. L., and Granger, E. Multi teacher privileged knowledge distillation for multimodal expression recognition, 2024. URL <https://arxiv.org/abs/2408.09035>.
- Axelrod, S. and Gómez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022. doi: 10.1038/s41597-022-01288-4. URL <https://doi.org/10.1038/s41597-022-01288-4>.
- Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers, 2022. URL <https://arxiv.org/abs/2106.08254>.
- Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning, 2017. URL <https://arxiv.org/abs/1707.06887>.
- Bhalla, U., Oesterling, A., Srinivas, S., Calmon, F. P., and Lakkaraju, H. Interpreting clip with sparse linear concept embeddings (splice), 2024.
- Borza, D.-L., Darabant, A., Ileni, T., and Marinescu, A.-I. Effective online knowledge distillation via attention-based model ensembling. *Mathematics*, 10:4285, 11 2022. doi: 10.3390/math10224285.
- Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T. (eds.), *Computer Vision – ECCV 2014*, pp. 446–461, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In Márquez, L., Callison-Burch, C., and Su, J. (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL <https://aclanthology.org/D15-1075>.
- Brody, S., Alon, U., and Yahav, E. How attentive are graph attention networks?, 2022. URL <https://arxiv.org/abs/2105.14491>.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.
- Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.
- Cohan, A., Feldman, S., Beltagy, I., Downey, D., and Weld, D. S. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*, 2020.

- Darrin, M., Formont, P., Cheung, J. C. K., and Piantanida, P. COSMIC: Mutual information for task-agnostic summarization evaluation, 2024. URL <https://arxiv.org/abs/2402.19457>.
- Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.
- Dong, C., Li, Y., Shen, Y., and Qiu, M. HRKD: Hierarchical relational knowledge distillation for cross-domain language model compression. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3126–3136, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.250. URL <https://aclanthology.org/2021.emnlp-main.250>.
- Dong, Y., Miller, K., Lei, Q., and Ward, R. Cluster-aware semi-supervised learning: relational knowledge distillation provably learns clustering. *Advances in Neural Information Processing Systems*, 36, 2024.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL <https://arxiv.org/abs/2010.11929>.
- Duval, Q., Misra, I., and Ballas, N. A simple recipe for competitive low-compute self supervised vision models, 2023. URL <https://arxiv.org/abs/2301.09451>.
- Dvornik, N., Schmid, C., and Mairal, J. Diversity with cooperation: Ensemble methods for few-shot classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3723–3731, 2019.
- Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., and Liu, Z. Seed: Self-supervised distillation for visual representation, 2021. URL <https://arxiv.org/abs/2101.04731>.
- Farebrother, J., Orbay, J., Vuong, Q., Taïga, A. A., Chebotar, Y., Xiao, T., Irpan, A., Levine, S., Castro, P. S., Faust, A., Kumar, A., and Agarwal, R. Stop regressing: Training value functions via classification for scalable deep rl, 2024. URL <https://arxiv.org/abs/2403.03950>.
- Feng, S., Ni, Y., Lan, Y., Ma, Z.-M., and Ma, W.-Y. Fractional denoising for 3D molecular pre-training. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9938–9961. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/feng23c.html>.
- Feng, S., Ni, Y., Li, M., Huang, Y., Ma, Z.-M., Ma, W.-Y., and Lan, Y. Unicorn: A unified contrastive learning approach for multi-view molecular representation learning, 2024. URL <https://arxiv.org/abs/2405.10343>.
- Frosst, N. and Hinton, G. Distilling a neural network into a soft decision tree, 2017.
- Gao, Y., Zhuang, J.-X., Lin, S., Cheng, H., Sun, X., Li, K., and Shen, C. Disco: Remedy self-supervised learning on lightweight models with distilled contrastive learning, 2022. URL <https://arxiv.org/abs/2104.09124>.
- Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021. ISSN 1573-1405. doi: 10.1007/s11263-021-01453-z. URL <http://dx.doi.org/10.1007/s11263-021-01453-z>.
- Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs, 2018. URL <https://arxiv.org/abs/1706.02216>.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Herbold, S. Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48):2173, 2020. doi: 10.21105/joss.02173. URL <https://doi.org/10.21105/joss.02173>.

- Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks, 2020. URL <https://arxiv.org/abs/1905.12265>.
- Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. Ogb-lsc: A large-scale challenge for machine learning on graphs, 2021.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- Huang, K., Fu, T., Gao, W., Zhao, Y., Roohani, Y., Leskovec, J., Coley, C. W., Xiao, C., Sun, J., and Zitnik, M. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5mb model size, 2016. URL <https://arxiv.org/abs/1602.07360>.
- Irwin, J. J. and Shoichet, B. K. ZINC – A Free Database of Commercially Available Compounds for Virtual Screening. *Journal of chemical information and modeling*, 45(1):177–182, 2005. ISSN 1549-9596. doi: 10.1021/ci049714. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1360656/>.
- Isert, C., Atz, K., Jiménez-Luna, J., and Schneider, G. Qmugs: Quantum mechanical properties of drug-like molecules, 2021.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.
- Kim, S., Chen, J., Cheng, T., Gindulyte, A., He, J., He, S., Li, Q., Shoemaker, B. A., Thiessen, P. A., Yu, B., Zaslavsky, L., Zhang, J., and Bolton, E. E. PubChem 2023 update. *Nucleic Acids Research*, 51(D1):D1373–D1380, 10 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac956. URL <https://doi.org/10.1093/nar/gkac956>.
- Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi: 10.1109/ICCVW.2013.77.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images, 2009.
- Kubota, Y., Haraguchi, D., and Uchida, S. Impression-clip: Contrastive shape-impression embedding for fonts, 2024.
- Li, S., Yang, X., Cheng, G., Liu, W., and Hu, H. Sa-mdrad: sample-adaptive multi-teacher dynamic rectification adversarial distillation. *Multimedia Systems*, 30(4), July 2024. ISSN 1432-1882. doi: 10.1007/s00530-024-01416-7. URL <http://dx.doi.org/10.1007/s00530-024-01416-7>.
- Li, X. and Li, J. Angle-optimized text embeddings, 2023.
- Liang, C., Jiang, H., Li, Z., Tang, X., Yin, B., and Zhao, T. Homodistil: Homotopic task-agnostic distillation of pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.
- Liu, L., Huang, Q., Lin, S., Xie, H., Wang, B., Chang, X., and Liang, X. Exploring inter-channel correlation for diversity-preserved knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8271–8280, 2021a.
- Liu, S., Wang, H., Liu, W., Lasenby, J., Guo, H., and Tang, J. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2022. URL <https://openreview.net/forum?id=xQUe1pOKPam>.

- Liu, W., Chen, X., Liu, J., Feng, S., Sun, Y., Tian, H., and Wu, H. Ernie 3.0 tiny: Frustratingly simple method to improve task-agnostic distillation generalization. *arXiv preprint arXiv:2301.03416*, 2023.
- Liu, Y. L., Blodgett, S. L., Cheung, J., Liao, Q. V., Olteanu, A., and Xiao, Z. ECBD: Evidence-centered benchmark design for NLP. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16349–16365, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.861. URL <https://aclanthology.org/2024.acl-long.861>.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b.
- Ma, D., Zhang, K., Cao, Q., Li, J., and Gao, X. Coordinate attention guided dual-teacher adaptive knowledge distillation for image classification. *Expert Systems with Applications*, 250:123892, 2024a. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2024.123892>. URL <https://www.sciencedirect.com/science/article/pii/S0957417424007589>.
- Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.
- Ma, Z., Dong, J., Ji, S., Liu, Z., Zhang, X., Wang, Z., He, S., Qian, F., Zhang, X., and Yang, L. Let all be whitened: Multi-teacher distillation for efficient visual retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):4126–4135, March 2024b. ISSN 2159-5399. doi: 10.1609/aaai.v38i5.28207. URL <http://dx.doi.org/10.1609/aaai.v38i5.28207>.
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P11-1015>.
- Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.
- McAuley, J. and Leskovec, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys ’13, pp. 165–172, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324090. doi: 10.1145/2507157.2507163. URL <https://doi.org/10.1145/2507157.2507163>.
- Merrick, L. Embedding and clustering your data can improve contrastive pretraining, 2024. URL <https://arxiv.org/abs/2407.18887>.
- Merrick, L., Xu, D., Nuti, G., and Campos, D. Arctic-embed: Scalable, efficient, and accurate text embedding models, 2024. URL <https://arxiv.org/abs/2405.05374>.
- Mobley, D. L. and Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design*, 28(7):711–720, July 2014. ISSN 1573-4951. doi: 10.1007/s10822-014-9747-x. URL <https://doi.org/10.1007/s10822-014-9747-x>.
- Morgan, H. L. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965. doi: 10.1021/c160017a018. URL <https://doi.org/10.1021/c160017a018>.
- Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks, 2021. URL <https://arxiv.org/abs/1810.02244>.

- Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. Mteb: Massive text embedding benchmark, 2023.
- Muralidharan, S., Sreenivas, S. T., Joshi, R., Chochowski, M., Patwary, M., Shoneybi, M., Catanzaro, B., Kautz, J., and Molchanov, P. Compact language models via pruning and knowledge distillation. *arXiv preprint arXiv:2407.14679*, 2024.
- Murphy, K. P. *Machine learning : a probabilistic perspective*. MIT Press, Cambridge, Mass. [u.a.], 2013. ISBN 9780262018029 0262018020. URL https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.
- Navaneet, K. L., Koohpayegani, S. A., Tejankar, A., and Pirsiavash, H. Simreg: Regression as a simple yet effective tool for self-supervised knowledge distillation, 2022. URL <https://arxiv.org/abs/2201.05131>.
- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 4. Granada, 2011.
- Ni, J., Ábrego, G. H., Constant, N., Ma, J., Hall, K. B., Cer, D., and Yang, Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models, 2021.
- Noroozi, M., Vinjimoor, A., Favaro, P., and Pirsiavash, H. Boosting self-supervised learning via knowledge transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9359–9367, 2018.
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- Pan, Z., Zhou, X., and Tian, H. Extreme generative image compression by learning text embedding from diffusion models. *arXiv preprint arXiv:2211.07793*, 2022.
- Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019a.
- Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation, 2019b. URL <https://arxiv.org/abs/1904.05068>.
- Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.
- Peng, B., Jin, X., Liu, J., Li, D., Wu, Y., Liu, Y., Zhou, S., and Zhang, Z. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- Pichler, G., Colombo, P., Boudiaf, M., Koliander, G., and Piantanida, P. A differential entropy estimator for training neural networks, 2022.
- Pimentel, T., Meister, C., and Cotterell, R. On the usefulness of embeddings, clusters and strings for text generator evaluation, 2023.
- Polykovskiy, D., Zhebrak, A., Sánchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Nikolenko, S. I., Aspuru-Guzik, A., and Zhavoronkov, A. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *CoRR*, abs/1811.12823, 2018. URL <http://arxiv.org/abs/1811.12823>.
- Qiu, S., Han, B., Maddix, D. C., Zhang, S., Wang, Y., and Wilson, A. G. Transferring knowledge from large foundation models to small downstream models. *arXiv preprint arXiv:2406.07337*, 2024.

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
- Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3687–3697, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL <https://www.aclweb.org/anthology/D18-1404>.
- Sarkar, P. and Etemad, A. Xkd: Cross-modal knowledge distillation with domain alignment for video representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14875–14885, 2024.
- Scimeca, L., Rubinstein, A., Teney, D., Oh, S. J., Nicolicioiu, A. M., and Bengio, Y. Short-cut bias mitigation via ensemble diversity using diffusion probabilistic models. *arXiv preprint arXiv:2311.16176*, 2023.
- Shayegh, B., Cao, Y., Zhu, X., Cheung, J. C., and Mou, L. Ensemble distillation for unsupervised constituency parsing. In *International Conference on Learning Representations (ICLR)*, 2024.
- Solatorio, A. V. Gistembed: Guided in-sample selection of training negatives for text embedding fine-tuning. *arXiv preprint arXiv:2402.16829*, 2024. URL <https://arxiv.org/abs/2402.16829>.
- Song, K., Xie, J., Zhang, S., and Luo, Z. Multi-mode online knowledge distillation for self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11848–11857, 2023.
- Stewart, L., Bach, F., Berthet, Q., and Vert, J.-P. Regression as classification: Influence of task formulation on neural network features, 2023. URL <https://arxiv.org/abs/2211.05641>.
- Stärk, H., Beaini, D., Corso, G., Tossou, P., Dallago, C., Günnemann, S., and Liò, P. 3d infomax improves gnn for molecular property prediction. *arXiv preprint arXiv:2110.04126*, 2021.
- Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., tau Yih, W., Smith, N. A., Zettlemoyer, L., and Yu, T. One embedder, any task: Instruction-finetuned text embeddings, 2023. URL <https://arxiv.org/abs/2212.09741>.
- Sun, S., Ren, W., Li, J., Wang, R., and Cao, X. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15731–15740, 2024.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.
- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2820–2828, 2019.
- Theisen, R., Kim, H., Yang, Y., Hodgkinson, L., and Mahoney, M. W. When are ensembles really effective? *Advances in Neural Information Processing Systems*, 36, 2024.
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardaş, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023.

- Vilnis, L. and McCallum, A. Word representations via gaussian embedding. In Bengio, Y. and LeCun, Y. (eds.), *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/1412.6623>.
- Wang, H., Li, W., Jin, X., Cho, K., Ji, H., Han, J., and Burke, M. D. Chemical-reaction-aware molecule representation learning. In *International Conference on Learning Representations*, 2022a. URL <https://openreview.net/forum?id=6sh3pIzKS->.
- Wang, H., Ma, C., Zhang, J., Zhang, Y., Avery, J., Hull, L., and Carneiro, G. Learnable cross-modal knowledge distillation for multi-modal learning with missing modality. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 216–226. Springer, 2023.
- Wang, K., Yang, F., and van de Weijer, J. Attention distillation: self-supervised vision transformer students need more guidance. *arXiv preprint arXiv:2210.00944*, 2022b.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022c.
- Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.
- Wenlock, M. and Tomkinson, N. Experimental in vitro dmpk and physicochemical data on a set of publicly disclosed compounds, 2021. URL https://www.ebi.ac.uk/chembl/document_report_card/CHEMBL3301361/.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Huggingface’s transformers: State-of-the-art natural language processing, 2020. URL <https://arxiv.org/abs/1910.03771>.
- Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning, October 2018. URL <http://arxiv.org/abs/1703.00564>.
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.
- Xu, H., Fang, J., Zhang, X., Xie, L., Wang, X., Dai, W., Xiong, H., and Tian, Q. Bag of instances aggregation boosts self-supervised distillation, 2022a. URL <https://arxiv.org/abs/2107.01691>.
- Xu, H., Koehn, P., and Murray, K. The importance of being parameters: An intra-distillation method for serious gains. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 170–183, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.13. URL <https://aclanthology.org/2022.emnlp-main.13>.
- Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=ryGs6iA5Km>.
- Xu, M., Wang, H., Ni, B., Guo, H., and Tang, J. Self-supervised graph-level representation learning with local and global structure. *arXiv preprint arXiv:2106.04113*, 2021.
- Ye, X., Jiang, R., Tian, X., Zhang, R., and Chen, Y. Knowledge distillation via multi-teacher feature ensemble. *IEEE Signal Processing Letters*, 2024.

- Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7130–7138, 2017. doi: 10.1109/CVPR.2017.754.
- You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5812–5823. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/3fe230348e9a12c13120749e3f9fa4cd-Paper.pdf>.
- Yuan, F., Shou, L., Pei, J., Lin, W., Gong, M., Fu, Y., and Jiang, D. Reinforced multi-teacher selection for knowledge distillation, 2020.
- Zagoruyko, S. and Komodakis, N. Wide residual networks, 2017. URL <https://arxiv.org/abs/1605.07146>.
- Zhang, H., Chen, D., and Wang, C. Adaptive multi-teacher knowledge distillation with meta-learning. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1943–1948. IEEE, 2023.
- Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., and Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3713–3722, 2019.
- Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *NIPS*, 2015.
- Zhu, J., Liu, J., Li, W., Lai, J., He, X., Chen, L., and Zheng, Z. Ensembled ctr prediction via knowledge distillation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2941–2958, 2020.

Appendix

Table of Contents

A Proofs	20
A.1 Proof of Theorem 3.2	20
B Molecular Modelling	21
B.1 Model architecture	21
B.1.1 Chosen Teachers	21
B.1.2 Architecture influence	21
B.1.3 Additional results on the TDC datasets	22
B.2 Kernel’s predictive power	23
B.3 Evaluation details	24
B.3.1 Benchmark Choice	24
B.3.2 Evaluation Procedure	24
B.3.3 Evaluation Metrics	24
B.4 Single-Teacher setting	24
B.5 Comprehensive results	26
C Natural Language Processing	29
C.1 Training set and hyperparameters	29
C.1.1 Training set	29
C.1.2 Teachers and based students performance	30
C.1.3 Single teacher distillation	30
C.1.4 Hyperparameters	30
C.2 Detailed evaluation results	30
C.2.1 Evaluation on classification tasks	30
C.2.2 Evaluation on similarity and clustering tasks	30
C.2.3 Analysis and compare with the most recent embedders	31
D Vision	37
D.1 Model architecture	37
D.2 Training Set	37
D.3 Vision Details	37
D.4 Complementary Results	37
E Detailed Method	40
F Computaional ressources	40
G Baselines	40
H Discussion On MSE distillation	41
I Funding	47

A Proofs

We denote $\mathbf{X}$ as the random variable over $\mathcal{X}$ that describes the input distribution. We suppose we have access to a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$ of inputs drawn following $p_{\mathbf{X}}$ and different embedders $\mathbf{T}_k : \mathcal{X} \rightarrow \mathbb{R}^{d_k}$, $k \in \{1, \dots, K\}$, that map the inputs to different embedding spaces. We denote $\mathbf{Z}_k = \mathbf{T}_k(\mathbf{X})$ as the random variable over $\mathbb{R}^{d_k}$ that describes the embedding of the input distribution in the k -th embedding space and by $\mathbf{U} = \mathbf{S}(\mathbf{X})$ the random variable over $\mathbb{R}^d$ that describe the embedding of the input distribution in the student embedding space. We denote by $\mathbf{z}_i^k = \mathbf{T}_k(\mathbf{x}_i)$ the embedding of $\mathbf{x}_i$ in the k -th embedding space. We are interested in learning a representation that captures the information contained in all the embeddings.

Let us consider a task characterized by a target set $\mathcal{Y}$ of discrete concepts and the feature space $\mathcal{X}$ with joint probability measure $P_{\mathbf{Y}\mathbf{X}} \in \mathcal{P}(\mathcal{Y} \times \mathcal{X})$. For every projection of the features through the different teachers, the Bayes decision rule $c_{\mathbf{T}_k}^* \triangleq \arg \max_{c: \mathbb{R}^{d_k} \rightarrow \mathcal{Y}} \mathbb{E}_{\mathbf{X}\mathbf{Y}} [\mathbb{1}[c(\mathbf{T}_k(\mathbf{X})) = \mathbf{Y}]]$ and similarly for the student: $c_S^* \triangleq \arg \max_{c: \mathbb{R}^d \rightarrow \mathcal{Y}} \mathbb{E}_{\mathbf{X}\mathbf{Y}} [\mathbb{1}[c(\mathbf{S}(\mathbf{X})) = \mathbf{Y}]]$.

We leverage the following recent result from (Darrin et al., 2024):

Proposition A.1. *Let $C_{\mathbf{T}_k} = c_{\mathbf{T}_k}^*(\mathbf{T}_k(\mathbf{X}))$ and $C_S = c_S^*(\mathbf{S}(\mathbf{X}))$ denote the outcome of the Bayes classifier observing the output of the teacher $\mathbf{T}_k$ and the student $\mathbf{S}$, respectively*

$$\Pr(C_S \neq C_{\mathbf{T}_k}) \leq 1 - \exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))). \quad (4)$$

A.1 Proof of Theorem 3.2

By applying the above proposition to all the terms in Eq. 1, we obtain the following bound on the loss function:

Proposition 1 (Upper bound).

$$\mathcal{L}^*(\mathbf{X}\mathbf{Y}, \mathbf{S}, \mathbf{T}_1, \dots, \mathbf{T}_K) \leq \frac{1}{K} \sum_{k=1}^K (1 - \exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X})))) \quad (5)$$

$$\leq 1 - \exp\left(-\underbrace{\frac{1}{K} \sum_{k=1}^K h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))}_{\text{Negative log likelihood}}\right). \quad (6)$$

Proof.

$$\begin{aligned} \mathcal{L}^*(\mathbf{X}\mathbf{Y}, \mathbf{S}, \mathbf{T}_1, \dots, \mathbf{T}_K) &\leq \frac{1}{K} \sum_{k=1}^K (1 - \exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X})))) \\ &\leq 1 - \frac{1}{K} \sum_{k=1}^K \exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))) \\ &\leq 1 + \frac{1}{K} \sum_{k=1}^K -\exp(-h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))) \\ &\leq 1 - \exp\left(-\frac{1}{K} \sum_{k=1}^K h(\mathbf{T}_k(\mathbf{X})|\mathbf{S}(\mathbf{X}))\right). \end{aligned}$$

We simply rearrange the terms and use the fact that $x \mapsto -\exp(-x)$ is concave to interchange the sum and the exponential. $\square$

B Molecular Modelling

B.1 Model architecture

We trained a 10-layer GINE (Hu et al., 2020) neural network with a 512 hidden dimension, using a 2-layer network for the message passing process. We use the atomic number of each node as input, as well as possible chirality information, and the nature of the bond between each pair of nodes. We use a batch size of 256 and a learning rate of $1e-4$ to train the model for 400 epochs on the 250k dataset and 200 epochs on the 2M dataset. For the teacher-specific kernels, we used a 3-layer MLP with a hidden size of 1024.

B.1.1 Chosen Teachers

The teachers used to train our molecular modeling students are summed up in Tab. 2. We gathered various representation models for molecular modeling, with different pre-training objectives, input modalities, architectures, and training datasets.

Table 2: Description of all teachers used in our experiments.

Model name	SMILES	2D-GNN	3D-GNN	Architecture	Out size	Dataset (size)
GraphCL(You et al., 2020)		✓		GIN	300	GEOM (Axelrod & Gómez-Bombarelli, 2022) (50k)
GraphLog(Xu et al., 2021)		✓		GIN	300	GEOM (Axelrod & Gómez-Bombarelli, 2022) (50k)
GraphMVP(Liu et al., 2022) ¹		✓		GIN	300	GEOM (Axelrod & Gómez-Bombarelli, 2022) (50k)
3D-infomax(Stärk et al., 2021) ¹		✓		PNA	800	QMugs (Isert et al., 2021) (620k)
ChemBERT MTR(Ahmad et al., 2022) ²	✓			RoBERTa	384	PubChem (Kim et al., 2022) (5M, 10M, 77M)
3D-fractional(Feng et al., 2023)			✓	TorchMD-net	256	PCQM4Mv2(Hu et al., 2021) (3.7M)

B.1.2 Architecture influence

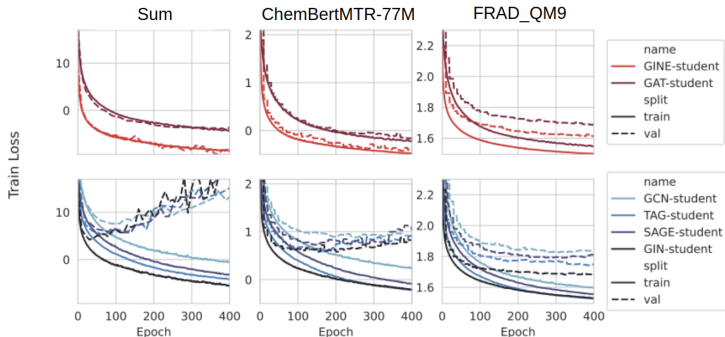


Figure 6: Training loss of different students using different GNN architectures on the ZINC-250k dataset.

Figure 6 shows the training loss of the student model with different GNN architectures on the ZINC-250k dataset. In particular, we compared the GINE architecture with a Graph Convolutional Network (GCN) (Morris et al., 2021), a Graph Attention Network (GAT) (Brody et al., 2022), a GraphSAGE (SAGE) (Hamilton et al., 2018), a Topology Adaptive Graph Convolutional Network (TAG) (Brody et al., 2022), and a GIN Network, that separates from the GINE architecture by the fact that it does not take edge features into account (Xu et al., 2019). We observe that the GINE architectures outperform the other architectures, with a lower training loss, a faster convergence, and a lower validation loss. The Graph attention network (GAT) is the second best performing architecture, but it is still outperformed by the GINE architecture. These two architectures are the only ones to use the edge embeddings in the message passing process, which could explain their better performance.

¹Models aiming at incorporating 3D information into 2D-GNNs models.

²We used the three versions of ChemBERT-MTR models trained on 5M, 10M, and 77M.

Indeed, all other architectures perform worse, especially when considering their validation loss computed on 10% of the training set. Specifically, the GIN architecture, not using edge feature, performs significantly worse than the GINE architecture, while having a similar architecture.

For our experiments, we decided to use the GINE architecture, as it performs the best during training and converges faster than the other architectures.

B.1.3 Additional results on the TDC datasets

Table 3: Average rank of each model on the ADMET and HTS downstream tasks from the TDC (Huang et al., 2021) platform. Our student outperforms all baselines, including teachers, on average.

	Absorption	Distribution	Metabolism	Excretion	Tox	HTS	Avg
InfoGraph	13.50	13.27	13.32	11.40	11.98	9.40	12.14
ChemBertMLM-10M	10.65	11.00	10.70	13.80	11.11	14.60	11.98
FRAD QM9 ^(t)	10.57	11.13	10.38	8.33	10.04	7.80	9.71
ChemGPT-1.2B	9.55	11.73	11.75	10.73	10.86	11.20	10.97
GROVER	10.43	8.33	11.25	8.53	10.38	11.00	9.99
GraphCL ^(t)	10.89	8.53	9.45	10.13	8.70	9.80	9.58
GraphLog ^(t)	11.05	7.80	9.07	10.53	8.93	14.00	10.23
GraphMVP ^(t)	7.20	6.20	7.85	9.80	7.49	8.80	7.89
MolR gat	6.95	7.60	8.30	8.53	6.49	3.40	6.88
ThreeDInfomax ^(t)	4.17	6.00	7.58	7.13	6.16	10.40	6.91
ChemBertMTR-77M ^(t)	3.50	4.27	5.75	5.00	6.03	4.20	4.79
MSE	8.07	6.40	5.55	6.33	7.55	3.00	6.15
Cosine	5.51	6.13	3.60	4.33	4.97	6.20	5.13
student-250k	3.55	6.20	2.70	2.40	4.99	3.80	3.94
student-2M	4.40	5.40	2.75	3.00	4.34	2.40	3.72

The average rank of each model in each task category can be found in Tab. 3. Surprisingly, the performances of the "student-250k" and "student-2M" models are similar on average. Specifically, the student-250k model outperforms the student-2M model on regression datasets notably, by achieving the best performances on the FreeSolv (Mobley & Guthrie, 2014) and Lipophilicity (Wenlock & Tomkinson, 2021) tasks. This suggests that our method can leverage the diversity of the teachers to learn more informative representations, even when trained on a smaller dataset of 250k datapoints.

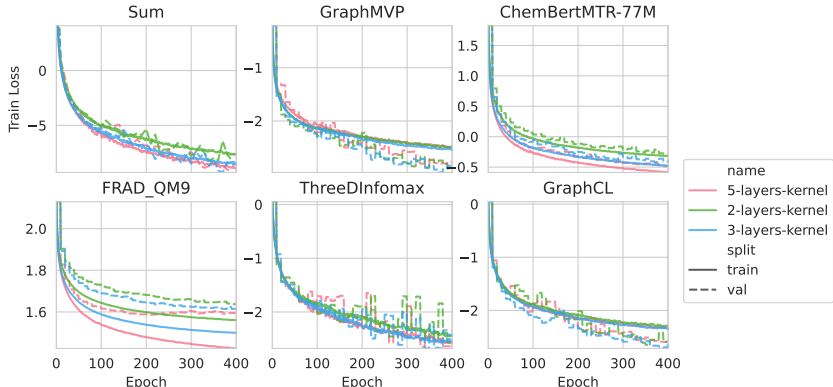


Figure 7: Training loss of the student model along the training with different kernel-size on the ZINC-250k dataset.

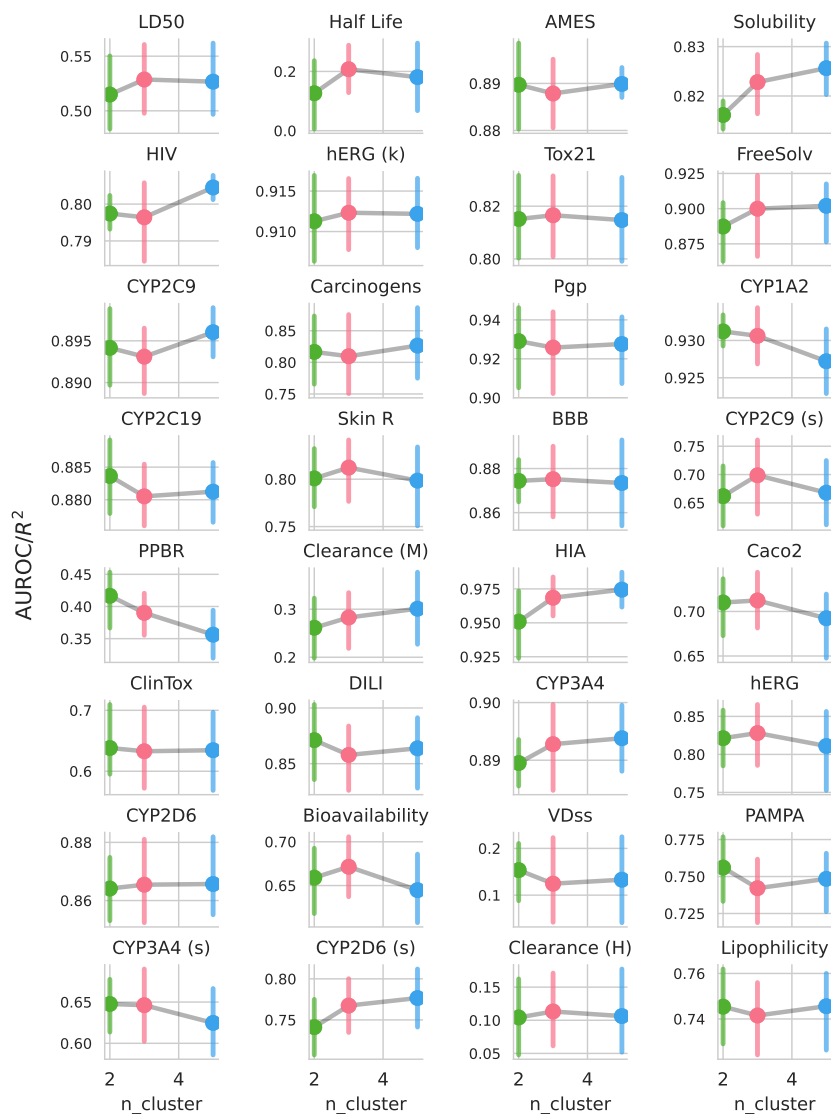


Figure 8: Test AUROC/ R^2 score of the students on the classification/regression tasks, trained with different kernel-size on the ZINC-250k dataset.

B.2 Kernel’s predictive power

Our method relies on teacher-specific heads to distill the knowledge of each teacher. In this section, we wish to evaluate the impact of the choice of these kernels and their predictive power (in terms of depth) on the performance and training of the student model.

We performed this experiment with kernels of depth 2, 3, and 5, and we trained the student model with these kernels on the ZINC-250k dataset and evaluated the performance of the student model on the ADMET and HTS downstream tasks.

First, during the training, as expected, the more powerful the kernel, the lower the training loss is (see Figure 7), even though the difference is significant, especially between the students using kernels of depth 3 and 5. Overall, the performances of each student on the downstream tasks are similar, underlining the robustness of our method regarding the choice of the kernel’s depth (see Figure 8). For our experiments in the main paper, we used a kernel of depth 3, as it enables the best trade-off

between computational complexity, and training convergence while providing competitive results on the downstream tasks.

B.3 Evaluation details

B.3.1 Benchmark Choice

We selected a total of 32 tasks, extracted from the Therapeutic Data Commons (Huang et al., 2021) platform, 8 absorption tasks, 3 distribution tasks, 8 metabolism tasks, 3 excretion tasks, 9 toxicity tasks and 1 high-throughput screening task. A summary of the tasks considered can be found in Tab. 4, with their corresponding size (total number of samples) and type (classification or regression). For all tasks, we computed 5 conformations for each molecule, and used the least energetic as an input of our 3D models.

B.3.2 Evaluation Procedure

For every task, we opted for a random split since we obtained similar results to a scaffold split, with a faster computation time, with a ratio of 70/10/20 for the train/validation/test sets. For all tasks, we compute the embeddings generated by each model on the task. We then train a 2 layer perceptron with a hidden size of 128 on the task for $\min(100, 200 * \frac{5000}{\text{task size}})$ epochs (to limit the compute time on large tasks) with a learning rate of $1e - 3$. We then select the best checkpoint according to the validation performances and report the test metrics of this checkpoint.

B.3.3 Evaluation Metrics

We repeat this process five times with different seeds in the train-val-test splits in order to enable the establishment of robust rankings using autorank (Herbold, 2020). We decided to report the ranks of the models to enable the comparison of the models on both classification and regression by simply averaging the rank. To compute the rank on all tasks, we rely on the AUROC score for classification tasks and the R^2 score for regression tasks. For the excretion tasks, since the regression labels have a large variance, we decided to apply the regression on the log-values and report the R^2 score on the log-values.

B.4 Single-Teacher setting

To assess the impact of the multi-teacher setting on the performance of the student model, we trained students to distill the knowledge of a single teacher. We used only the two best performing teachers, 3D-infomax (Stärk et al., 2021) and ChemBERTaMTR (Ahmad et al., 2022), to train the student model on the 2M datapoints dataset. We also train a student with both teachers, to see if those two teachers are sufficient to achieve the same performance as the models we presented in the core of the paper.

Figure 9 shows how these students underperform compared to a student trained with all teachers, in terms of AUROC for classification tasks and R^2 for regression tasks respectively. These tables also show that the student trained with both teachers performs better than each student trained with only one teacher. All results are aggregated in Tab. 6 and Tab. 5.

Table 4: Tasks extracted from the Therapeutic Data Commons platform considered in our experiments.

Category	Model	Task	cls	reg
Absorption	P-glycoprotein Inhibition	1212	✓	
	AqSolDB	9982		✓
	Lipophilicity	4200		✓
	Caco-2 Permeability	906		✓
	Human Intestinal Absorption	578	✓	
	FreeSolv	642		✓
	PAMPA Permeability	2035	✓	
	Oral Bioavailability	640	✓	
Distribution	Plasma-Protein BDR	1614		✓
	Blood-Brain barrier	1975	✓	
	VDss	1130		✓
Metabolism	CYP450 3A4 Inhib.	12328	✓	
	CYP450 1A2 Inhib.	12579	✓	
	CYP450 2C19 Inhib.	12665	✓	
	CYP450 2C9 Inhib.	12092	✓	
	CYP450 2D6 Inhib.	13130	✓	
	CYP450 2D6 Substrate	664	✓	
	CYP450 3A4 Substrate	667	✓	
	CYP450 2C9 Substrate	666	✓	
Excretion	Clearance hepatocyte	1020		✓
	Half Life	667		✓
	Clearance microsome	1102		✓
Toxicity	Tox21	7831	✓	
	hERG	13445	✓	
		648	✓	
	Acute Toxicity LD50	7385		✓
	Ames Mutagenicity	7255	✓	
	ClinTox	1484	✓	
	Carcinogens	278	✓	
	Drug Induced Liver Injury	475	✓	
	Skin Reaction	404	✓	
HTS	HIV	40000	✓	

Table 5: Performance of the student models trained with only the best teacher ("1-ChemBertMTR"), the second-best teacher ("1-3dinfo"), both teachers together ("2-teachers"), and "student-2M" on regression tasks (R2).

	avg	Absorption Caco2	Absorption FreeSolv	Absorption Lipophilicity	Absorption Solubility	Tox LD50
1-3dinfo	0.392± 0.317	0.654± 0.041	0.822± 0.044	0.583± 0.047	0.798± 0.010	0.471± 0.048
1-BertMTR	0.405± 0.309	0.660± 0.026	0.829± 0.031	0.582± 0.044	0.803± 0.010	0.480± 0.023
2-Teachers	0.449± 0.312	0.692± 0.043	0.882± 0.034	0.688± 0.028	0.812± 0.012	0.497± 0.033
student-2M	0.476± 0.301	0.687± 0.045	0.878± 0.036	0.739± 0.021	0.822± 0.005	0.543± 0.041
	Distribution PPBR	Distribution VDss	Excretion Clearance (H)	Excretion Clearance (M)	Excretion Half Life	
1-3dinfo	0.316± 0.062	0.130± 0.146	0.048± 0.095	0.137± 0.083	-0.037± 0.254	
1-BertMTR	0.347± 0.070	0.145± 0.072	0.051± 0.148	0.136± 0.110	0.017± 0.195	
2-Teachers	0.419± 0.032	0.172± 0.098	0.066± 0.075	0.199± 0.075	0.061± 0.135	
student-2M	0.389± 0.050	0.138± 0.115	0.069± 0.060	0.348± 0.062	0.144± 0.205	

Table 6: Performance of the student models trained with only the best teacher ("1-ChemBertMTR"), the second-best teacher ("1-3dinfo"), both teachers together ("2-teachers"), and "student-2M" on classification tasks (AUROC).

	avg	Absorption Bioavailability	Absorption HIA	Absorption PAMPA	Absorption Pgp	Distribution BBB	HTS HIV	
1-BertMTR	0.801± 0.101	0.631± 0.059	0.910± 0.054	0.719± 0.020	0.933± 0.021	0.886± 0.022	0.756± 0.005	
1-3dinfo	0.803± 0.100	0.631± 0.050	0.899± 0.056	0.737± 0.021	0.930± 0.024	0.890± 0.024	0.763± 0.010	
2-Teachers	0.808± 0.097	0.641± 0.055	0.868± 0.051	0.749± 0.009	0.927± 0.026	0.881± 0.026	0.786± 0.012	
student-2M	0.825± 0.096	0.653± 0.055	0.959± 0.026	0.730± 0.024	0.936± 0.024	0.882± 0.020	0.800± 0.014	
	Metabolism CYP1A2	Metabolism CYP2C19	Metabolism CYP2C9 (s)	Metabolism CYP2C9	Metabolism CYP2D6 (s)	Metabolism CYP2D6	Metabolism CYP3A4 (s)	Metabolism CYP3A4
1-BertMTR	0.916± 0.008	0.866± 0.007	0.622± 0.088	0.874± 0.006	0.729± 0.021	0.839± 0.010	0.638± 0.017	0.848± 0.014
1-3dinfo	0.916± 0.005	0.870± 0.006	0.630± 0.069	0.874± 0.005	0.721± 0.028	0.836± 0.008	0.638± 0.015	0.855± 0.007
2-Teachers	0.924± 0.006	0.871± 0.009	0.627± 0.086	0.883± 0.005	0.725± 0.043	0.848± 0.010	0.632± 0.047	0.881± 0.006
student-2M	0.933± 0.006	0.882± 0.007	0.697± 0.093	0.893± 0.002	0.766± 0.058	0.868± 0.010	0.639± 0.054	0.892± 0.004
	Tox AMES	Tox Carcinogens	Tox ClnTox	Tox DILI	Tox Skin R	Tox Tox21	Tox hERG	Tox hERG (k)
1-BertMTR	0.862± 0.014	0.831± 0.074	0.601± 0.069	0.831± 0.060	0.815± 0.057	0.801± 0.056	0.826± 0.023	0.877± 0.012
1-3dinfo	0.864± 0.011	0.826± 0.083	0.603± 0.074	0.841± 0.049	0.835± 0.051	0.802± 0.055	0.831± 0.019	0.875± 0.007
2-Teachers	0.875± 0.007	0.827± 0.062	0.657± 0.102	0.865± 0.024	0.770± 0.053	0.814± 0.057	0.826± 0.029	0.898± 0.006
student-2M	0.891± 0.014	0.839± 0.095	0.662± 0.072	0.856± 0.045	0.801± 0.026	0.819± 0.054	0.834± 0.019	0.911± 0.005

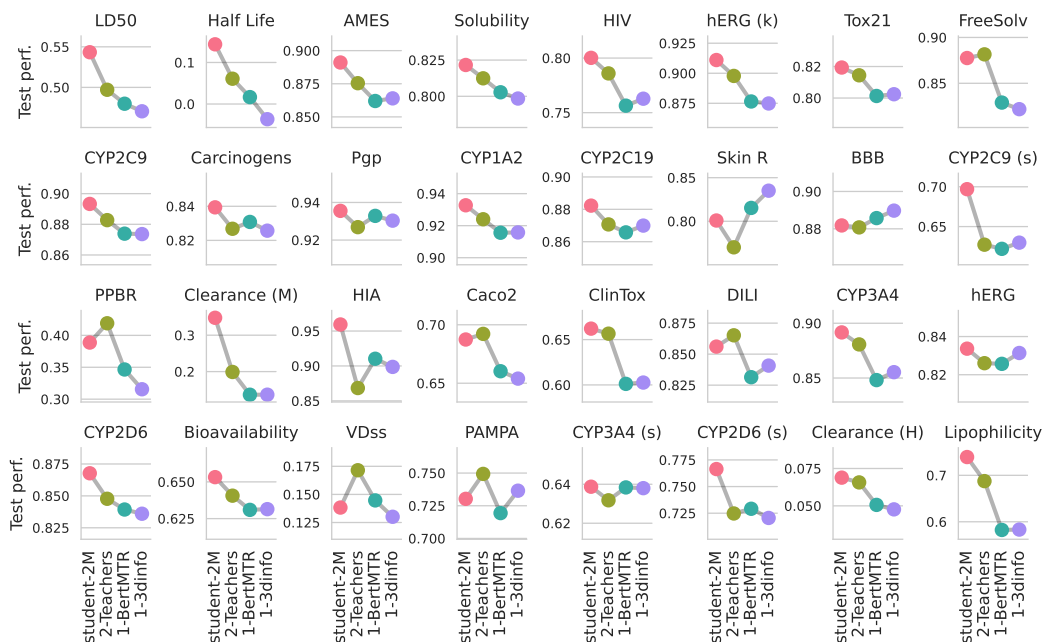


Figure 9: Test AUROC/ R^2 score of the students on the classification/regression tasks, trained with all teachers (student-2M), two teachers (2-Teachers) and one teacher (1-ChemBertMTR for the model trained with ChemBertMTR-77M and 1-teacher-3dinfomax for the model trained with 3D-infomax).

B.5 Comprehensive results

The following tables provide the raw results of the different evaluated models on the ADMET and HTS downstream tasks. Tab. 7 and Tab. 8 display the test performances of the models on the classification and regression tasks respectively. All regression tasks are evaluated using the R^2 score, while the classification tasks are evaluated using the AUROC score. We report the mean values of the metrics over 5 runs for each task, as well as the standard deviation.

We display in Figure 10 the evolution of the average rank of the embedders when separating the tasks based on the amount of samples, and the class imbalance (for classification tasks). Our student appears robust in both setups, even though as the class imbalance becomes more important, or as the amount of samples in the task decreases, the difference between the top-performing embedders becomes less significant.

Table 7: AUROC of each model on the ADMET and HTS downstream classification tasks. The best embedder for each task is highlighted in bold and underlined, and the second best is highlighted in bold.

	avg	Distribution BBB	HTS HIV	Metabolism CYP1A2	Metabolism CYP2C19	Metabolism CYP2C9 (s)	Metabolism CYP2D6	Metabolism CYP3A4 (s)	Metabolism CYP3A4
InfoGraph	0.768 ± 0.097	0.843 ± 0.022	0.769 ± 0.018	0.878 ± 0.003	0.832 ± 0.004	0.840 ± 0.007	0.813 ± 0.013	0.567 ± 0.042	0.817 ± 0.003
ChemGPT-1.2B	0.779 ± 0.084	0.853 ± 0.012	0.770 ± 0.014	0.886 ± 0.008	0.835 ± 0.004	0.852 ± 0.008	0.816 ± 0.020	0.608 ± 0.026	0.844 ± 0.006
FRAD QM9 ⁽ⁱ⁾	0.785 ± 0.111	0.769 ± 0.005	0.769 ± 0.005	0.906 ± 0.004	0.845 ± 0.010	0.860 ± 0.009	0.815 ± 0.020	0.607 ± 0.060	0.855 ± 0.003
ChemBERTMLM-10M	0.785 ± 0.089	0.868 ± 0.009	0.763 ± 0.012	0.886 ± 0.007	0.843 ± 0.007	0.843 ± 0.007	0.814 ± 0.014	0.606 ± 0.029	0.846 ± 0.005
GROVER	0.787 ± 0.086	0.869 ± 0.016	0.770 ± 0.012	0.880 ± 0.006	0.843 ± 0.008	0.853 ± 0.008	0.824 ± 0.013	0.599 ± 0.028	0.827 ± 0.007
GraphCL ⁽ⁱ⁾	0.792 ± 0.093	0.865 ± 0.011	0.765 ± 0.019	0.898 ± 0.004	0.858 ± 0.007	0.862 ± 0.006	0.828 ± 0.009	0.607 ± 0.044	0.846 ± 0.004
GraphLog ⁽ⁱ⁾	0.790 ± 0.096	0.867 ± 0.017	0.748 ± 0.008	0.886 ± 0.006	0.855 ± 0.004	0.860 ± 0.009	0.832 ± 0.007	0.624 ± 0.049	0.847 ± 0.011
GraphMPV ⁽ⁱ⁾	0.800 ± 0.097	0.874 ± 0.016	0.771 ± 0.016	0.899 ± 0.004	0.865 ± 0.006	0.862 ± 0.008	0.830 ± 0.016	0.619 ± 0.034	0.847 ± 0.009
MolR ^{gat}	0.808 ± 0.098	0.867 ± 0.025	0.797 ± 0.012	0.909 ± 0.004	0.859 ± 0.010	0.865 ± 0.009	0.841 ± 0.017	0.598 ± 0.045	0.873 ± 0.005
ThreeDifomax ⁽ⁱ⁾	0.815 ± 0.100	0.885 ± 0.019	0.762 ± 0.005	0.917 ± 0.005	0.868 ± 0.008	0.876 ± 0.009	0.842 ± 0.010	0.604 ± 0.015	0.858 ± 0.009
ChemBERTMTR-77M ⁽ⁱ⁾	0.816 ± 0.096	0.894 ± 0.026	0.797 ± 0.014	0.919 ± 0.007	0.873 ± 0.005	0.877 ± 0.014	0.845 ± 0.015	0.645 ± 0.081	0.883 ± 0.008
MSE	0.806 ± 0.094	0.878 ± 0.022	0.797 ± 0.005	0.917 ± 0.003	0.864 ± 0.008	0.873 ± 0.009	0.849 ± 0.012	0.641 ± 0.063	0.871 ± 0.008
Cosine	0.816 ± 0.096	0.881 ± 0.024	0.798 ± 0.011	0.925 ± 0.003	0.879 ± 0.005	0.887 ± 0.008	0.855 ± 0.015	0.637 ± 0.025	0.892 ± 0.005
student-250k	0.823 ± 0.095	0.875 ± 0.033	0.796 ± 0.013	0.931 ± 0.006	0.881 ± 0.005	0.893 ± 0.007	0.865 ± 0.017	0.646 ± 0.082	0.893 ± 0.004
student-2M	0.825 ± 0.096	0.882 ± 0.020	0.800 ± 0.014	0.933 ± 0.005	0.882 ± 0.007	0.893 ± 0.002	0.868 ± 0.010	0.639 ± 0.054	0.892 ± 0.004
InfoGraph	0.631 ± 0.015	0.872 ± 0.085	0.685 ± 0.031	0.896 ± 0.002	0.853 ± 0.009	0.837 ± 0.006	0.770 ± 0.058	0.778 ± 0.027	0.849 ± 0.009
ChemGPT-1.2B	0.668 ± 0.046	0.859 ± 0.058	0.665 ± 0.040	0.926 ± 0.007	0.843 ± 0.012	0.857 ± 0.011	0.762 ± 0.066	0.789 ± 0.009	0.867 ± 0.007
FRAD QM9 ⁽ⁱ⁾	0.626 ± 0.032	0.945 ± 0.014	0.699 ± 0.043	0.914 ± 0.027	0.871 ± 0.009	0.843 ± 0.044	0.713 ± 0.083	0.817 ± 0.032	0.873 ± 0.004
ChemBERTMLM-10M	0.664 ± 0.060	0.892 ± 0.064	0.715 ± 0.066	0.911 ± 0.029	0.858 ± 0.012	0.776 ± 0.093	0.792 ± 0.060	0.779 ± 0.052	0.867 ± 0.007
GROVER	0.663 ± 0.038	0.931 ± 0.038	0.703 ± 0.027	0.918 ± 0.029	0.867 ± 0.012	0.779 ± 0.084	0.747 ± 0.081	0.779 ± 0.077	0.856 ± 0.004
GraphCL ⁽ⁱ⁾	0.643 ± 0.027	0.863 ± 0.052	0.709 ± 0.042	0.920 ± 0.010	0.869 ± 0.008	0.835 ± 0.086	0.780 ± 0.059	0.774 ± 0.054	0.864 ± 0.004
GraphLog ⁽ⁱ⁾	0.622 ± 0.071	0.897 ± 0.035	0.707 ± 0.034	0.937 ± 0.004	0.873 ± 0.006	0.827 ± 0.055	0.787 ± 0.058	0.799 ± 0.058	0.864 ± 0.005
GraphMPV ⁽ⁱ⁾	0.694 ± 0.055	0.944 ± 0.025	0.718 ± 0.009	0.921 ± 0.006	0.874 ± 0.011	0.853 ± 0.015	0.801 ± 0.062	0.797 ± 0.049	0.849 ± 0.006
MolR ^{gat}	0.672 ± 0.049	0.957 ± 0.024	0.705 ± 0.041	0.928 ± 0.008	0.871 ± 0.014	0.867 ± 0.049	0.793 ± 0.065	0.823 ± 0.045	0.872 ± 0.005
ThreeDifomax ⁽ⁱ⁾	0.670 ± 0.033	0.986 ± 0.014	0.745 ± 0.026	0.929 ± 0.009	0.791 ± 0.034	0.830 ± 0.048	0.802 ± 0.061	0.844 ± 0.02	0.881 ± 0.011
ChemBERTMTR-77M ⁽ⁱ⁾	0.668 ± 0.047	0.960 ± 0.040	0.763 ± 0.026	0.936 ± 0.010	0.881 ± 0.006	0.838 ± 0.037	0.833 ± 0.041	0.829 ± 0.035	0.874 ± 0.010
MSE	0.626 ± 0.076	0.914 ± 0.040	0.735 ± 0.027	0.914 ± 0.030	0.876 ± 0.033	0.734 ± 0.094	0.817 ± 0.061	0.832 ± 0.018	0.895 ± 0.009
Cosine	0.629 ± 0.043	0.908 ± 0.062	0.755 ± 0.024	0.926 ± 0.021	0.884 ± 0.008	0.822 ± 0.084	0.814 ± 0.066	0.830 ± 0.038	0.908 ± 0.006
student-250k	0.671 ± 0.043	0.969 ± 0.025	0.926 ± 0.025	0.930 ± 0.024	0.888 ± 0.009	0.830 ± 0.079	0.817 ± 0.062	0.828 ± 0.038	0.911 ± 0.006
student-2M	0.653 ± 0.028	0.959 ± 0.028	0.936 ± 0.024	0.936 ± 0.024	0.891 ± 0.014	0.839 ± 0.095	0.801 ± 0.026	0.834 ± 0.09	0.912 ± 0.006

Table 8: R^2 score of each model on the ADMET downstream regression tasks. The best embedder for each task is highlighted in bold and underlined, and the second best is highlighted in bold.

	avg	Absorption Caco2	Absorption FreeSolv	Absorption Lipophilicity	Absorption Solubility
InfoGraph	0.275 $\pm$ 0.284	0.491 $\pm$ 0.031	0.639 $\pm$ 0.058	0.341 $\pm$ 0.035	0.700 $\pm$ 0.007
ChemBertMLM-10M	0.264 $\pm$ 0.364	0.543 $\pm$ 0.076	0.776 $\pm$ 0.038	0.363 $\pm$ 0.063	0.774 $\pm$ 0.007
FRAD QM9 ^(t)	0.332 $\pm$ 0.284	0.564 $\pm$ 0.051	0.686 $\pm$ 0.082	0.483 $\pm$ 0.029	0.758 $\pm$ 0.011
ChemGPT-1.2B	0.340 $\pm$ 0.329	0.567 $\pm$ 0.079	0.831 $\pm$ 0.048	0.487 $\pm$ 0.020	0.798 $\pm$ 0.009
GROVER	0.350 $\pm$ 0.274	0.575 $\pm$ 0.058	0.708 $\pm$ 0.024	0.470 $\pm$ 0.043	0.733 $\pm$ 0.027
GraphLog ^(t)	0.350 $\pm$ 0.311	0.545 $\pm$ 0.055	0.811 $\pm$ 0.017	0.486 $\pm$ 0.037	0.765 $\pm$ 0.010
GraphCL ^(t)	0.355 $\pm$ 0.292	0.559 $\pm$ 0.051	0.764 $\pm$ 0.038	0.467 $\pm$ 0.067	0.745 $\pm$ 0.021
GraphMVP ^(t)	0.397 $\pm$ 0.320	0.592 $\pm$ 0.064	0.861 $\pm$ 0.036	0.590 $\pm$ 0.064	0.791 $\pm$ 0.009
MolR gat	0.394 $\pm$ 0.307	0.651 $\pm$ 0.089	0.804 $\pm$ 0.075	0.518 $\pm$ 0.037	0.822 $\pm$ 0.010
ThreeDInfomax ^(t)	0.425 $\pm$ 0.322	0.700 $\pm$ 0.038	0.852 $\pm$ 0.055	0.624 $\pm$ 0.031	0.848$\pm$ 0.004
ChemBertMTR-77M ^(t)	0.459 $\pm$ 0.308	0.725$\pm$ 0.027	0.874 $\pm$ 0.037	0.670 $\pm$ 0.025	0.839$\pm$ 0.007
MSE	0.420 $\pm$ 0.299	0.642 $\pm$ 0.060	0.851 $\pm$ 0.063	0.605 $\pm$ 0.021	0.792 $\pm$ 0.018
Cosine	0.460 $\pm$ 0.311	0.699 $\pm$ 0.056	0.893$\pm$ 0.034	0.721 $\pm$ 0.028	0.815 $\pm$ 0.009
student-250k	0.482$\pm$ 0.298	0.712$\pm$ 0.040	0.900$\pm$ 0.035	0.742$\pm$ 0.019	0.823 $\pm$ 0.007
student-2M	0.476$\pm$ 0.301	0.687 $\pm$ 0.045	0.878 $\pm$ 0.036	0.739$\pm$ 0.021	0.822 $\pm$ 0.005

	Distribution PPBR	Distribution VDss	Excretion Clearance (H)	Excretion Clearance (M)	Excretion Half Life	Tox LD50
InfoGraph	0.093 $\pm$ 0.073	0.018 $\pm$ 0.190	-0.048 $\pm$ 0.133	0.070 $\pm$ 0.046	-0.011 $\pm$ 0.161	0.458 $\pm$ 0.039
ChemBertMLM-10M	0.112 $\pm$ 0.035	0.066 $\pm$ 0.091	-0.185 $\pm$ 0.122	0.040 $\pm$ 0.178	-0.240 $\pm$ 0.279	0.390 $\pm$ 0.044
FRAD QM9 ^(t)	0.180 $\pm$ 0.031	-0.004 $\pm$ 0.050	0.006 $\pm$ 0.095	0.124 $\pm$ 0.059	0.104 $\pm$ 0.129	0.415 $\pm$ 0.039
ChemGPT-1.2B	0.175 $\pm$ 0.036	0.046 $\pm$ 0.173	-0.018 $\pm$ 0.071	0.117 $\pm$ 0.099	-0.047 $\pm$ 0.182	0.442 $\pm$ 0.043
GROVER	0.185 $\pm$ 0.056	0.186$\pm$ 0.079	-0.034 $\pm$ 0.095	0.197 $\pm$ 0.082	0.035 $\pm$ 0.161	0.447 $\pm$ 0.058
GraphLog ^(t)	0.240 $\pm$ 0.082	0.202$\pm$ 0.111	-0.094 $\pm$ 0.053	0.068 $\pm$ 0.120	0.018 $\pm$ 0.192	0.457 $\pm$ 0.054
GraphCL ^(t)	0.237 $\pm$ 0.048	0.158 $\pm$ 0.075	-0.022 $\pm$ 0.127	0.123 $\pm$ 0.108	0.007 $\pm$ 0.165	0.508 $\pm$ 0.026
GraphMVP ^(t)	0.327 $\pm$ 0.036	0.168 $\pm$ 0.081	-0.009 $\pm$ 0.135	0.144 $\pm$ 0.071	-0.017 $\pm$ 0.226	0.527 $\pm$ 0.042
MolR gat	0.284 $\pm$ 0.093	0.155 $\pm$ 0.180	-0.024 $\pm$ 0.091	0.174 $\pm$ 0.050	0.059 $\pm$ 0.232	0.496 $\pm$ 0.040
ThreeDInfomax ^(t)	0.314 $\pm$ 0.053	0.152 $\pm$ 0.061	0.071 $\pm$ 0.049	0.195 $\pm$ 0.114	-0.004 $\pm$ 0.264	0.500 $\pm$ 0.040
ChemBertMTR-77M ^(t)	0.393$\pm$ 0.055	0.138 $\pm$ 0.127	0.011 $\pm$ 0.048	0.250 $\pm$ 0.078	0.196$\pm$ 0.190	0.491 $\pm$ 0.031
MSE	0.362 $\pm$ 0.077	0.135 $\pm$ 0.097	0.034 $\pm$ 0.097	0.244 $\pm$ 0.062	0.060 $\pm$ 0.116	0.470 $\pm$ 0.030
Cosine	0.382 $\pm$ 0.032	0.108 $\pm$ 0.084	0.079$\pm$ 0.102	0.275 $\pm$ 0.054	0.111 $\pm$ 0.158	0.515 $\pm$ 0.039
student-250k	0.390$\pm$ 0.042	0.125 $\pm$ 0.111	0.113$\pm$ 0.070	0.283$\pm$ 0.076	0.207$\pm$ 0.101	0.529$\pm$ 0.039
student-2M	0.389 $\pm$ 0.050	0.138 $\pm$ 0.115	0.069 $\pm$ 0.060	0.348$\pm$ 0.062	0.144 $\pm$ 0.205	0.543$\pm$ 0.041

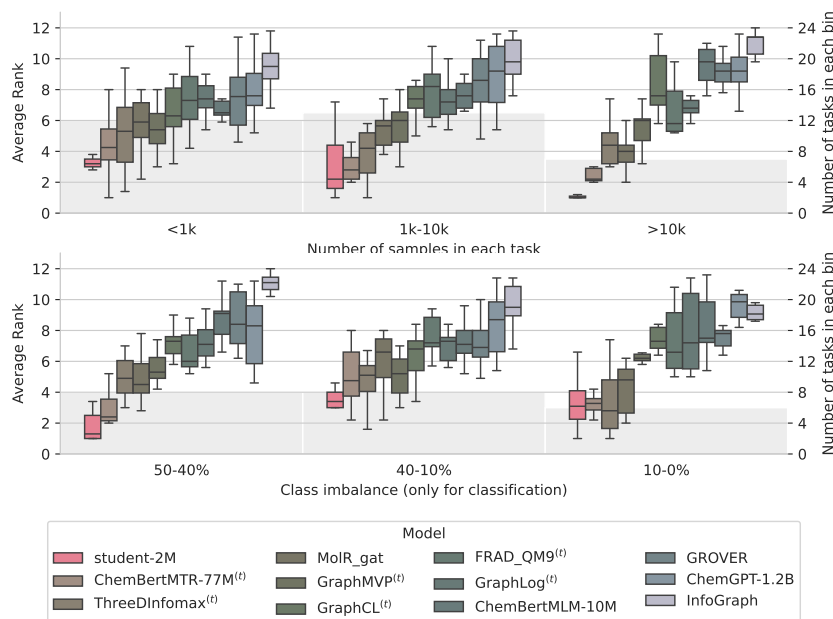


Figure 10: Average ranking of our models when grouping tasks based on the number of samples in the task and the class imbalance (for classification tasks).

C Natural Language Processing

C.1 Training set and hyperparameters

C.1.1 Training set

Dataset sources. We ran experiments with two training sets a home-made dataset combining different training sets of different embedders and the GISTEmbed dataset. We provide the statistics of our dataset in Tab. 9 and the GISTEmbed dataset is described in (Solatorio, 2024).

Dataset construction. Most embedding datasets consists of positive and negative samples, questions and answers, or sentences and their labels. We flattened the datasets to have only one column of sentences and deduplicated the dataset. For the MEDI () dataset for example, given query, positive and negative samples we build a dataset with three times the number of entries, one for each sentence. We then deduplicated the dataset to remove any duplicate entries.

Table 9: Number of samples in each dataset

URL	Number of samples
https://huggingface.co/datasets/embedding-data/SPECTER	190872
https://huggingface.co/datasets/embedding-data/Amazon-QA	3264474
https://huggingface.co/datasets/embedding-data/simple-wiki	203755
https://huggingface.co/datasets/embedding-data/QQP_triplets	328188
https://huggingface.co/datasets/embedding-data/sentence-compression	356409
https://huggingface.co/datasets/embedding-data/altlex	223901
https://huggingface.co/datasets/fancyzhx/ag_news	120000
https://huggingface.co/datasets/stanfordnlp/sst2	67349
https://huggingface.co/datasets/dair-ai/emotion	416809
https://huggingface.co/datasets/stanfordnlp/snli	1100304
https://huggingface.co/datasets/cardiffnlp/tweet_eval	45000
https://huggingface.co/datasets/stanfordnlp/imdb	25000
	6342061

Table 10: Performance of the 4 teachers we used and of the base students. Experiments with single teacher distillation were performed with the stronger teacher SFR-Embedding-2_R.

		Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTOPDomain	MTOPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
Teacher	SFR-Embedding-2_R	7111.0	92.7	97.3	61.0	90.0	93.4	96.8	98.6	91.3	86.0	90.6	91.1	79.7	89.0
	stella_en_400M_v5	435.0	92.4	97.2	59.5	89.3	78.8	96.5	98.8	92.3	85.2	89.6	86.9	73.6	86.7
	UAE-Large-V1	335.0	75.5	92.8	48.3	87.7	51.8	92.8	94.0	76.9	76.5	79.8	71.1	59.8	75.6
	sf_model_e5	335.0	70.8	91.8	48.9	84.6	54.9	93.1	93.6	66.0	73.5	77.4	71.2	61.5	74.0
Student (Base)	snowflake-arctic-embed-m	109.0	76.8	82.8	38.9	80.3	46.5	74.1	92.7	65.2	66.9	72.8	64.9	56.7	68.2
	snowflake-arctic-embed-s	33.0	71.2	78.8	38.3	79.1	45.8	69.5	90.9	58.6	64.8	70.0	62.0	58.9	65.7
	snowflake-arctic-embed-xs	23.0	65.1	70.0	35.3	76.4	41.8	62.8	90.8	58.0	63.5	71.0	64.3	56.2	62.9

C.1.2 Teachers and based students performance

Teachers. We selected 4 teachers from the MTEB benchmark (Muennighoff et al., 2023) as teachers for our distillation method. We provide the list of the teachers and their performance in Tab. 10. The 4 teachers of widely different sizes (335M, 435M and 7B) have display strong but different performances on the MTEB benchmark.

C.1.3 Single teacher distillation

Single teacher vs. Multi-Teachers. Since some teachers yield strong performance on their own, distilling only from the strongest could yield similar results as the multi-teacher setting involving weaker teachers. We applied our method in a single-teacher setting using the strongest teacher by far (SF-Embeddings-R.2) as a teacher and compared the results to the multi-teacher setting. Consistently with results in computer vision and molecular representations, we found that adding weaker teachers did improve our results (Figure 11), supporting our hypothesis that enforcing reconstruction capabilities for a diversity of models indeed leads to more informative representations.

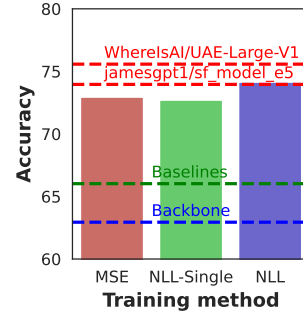


Figure 11: Comparison of distilled small model with the performance of the initial backbone, baselines in the MTEB, with our teachers’ performance.

C.1.4 Hyperparameters

Training hyperparameters. We trained our models using the Adam optimizer with a constant learning rate of 5.10^{-5} and an effective batch size of 16 for all our models.

C.2 Detailed evaluation results

We ran different parts of the MTEB benchmarks and report the overall results for all our models in this section.

C.2.1 Evaluation on classification tasks

Small models’ performance. In Tab. 11 and Tab. 12, we provide the classification accuracy of our distilled models on the MTEB classification benchmark for our smaller models xs (22M) and s (33M). Our smallest model significantly improves SOTA performance for models of its size by increasing the average score of 2 points compared to the previous best model.

C.2.2 Evaluation on similarity and clustering tasks

Limited structure of our embedding spaces. Our method only seeks to pack as much (statistical) information into the embeddings as possible without any constraints on the underlying structure of the embedding space. It is therefore not surprising that methods that relies on metrics on the embedding space such as similarity tasks do not perform as well as the classification tasks. However,

Table 11: Performance of our distilled models compared to models of similar sizes 16M to 30M parameters from the MTEB Benchmark on classification tasks.

Task Model	Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTOPDomain	MTOPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
GIST	23M	72.9	87.2	42.6	84.2	52.1	78.5	94.8	77.7	73.2	76.7	72.9	59.9	72.7
Bulbasaur	17M	71.9	78.8	39.3	80.6	44.8	71.5	90.8	68.7	68.8	73.8	66.3	59.5	67.9
Ivysaur	23M	72.1	<u>86.7</u>	42.7	81.9	45.4	80.8	92.1	71.9	70.3	74.9	65.5	58.7	70.2
Squirtle	16M	69.6	82.1	41.9	67.1	45.8	75.0	87.3	54.7	61.5	67.0	64.5	<u>61.8</u>	64.9
Venusaur	16M	<u>73.2</u>	80.0	39.7	78.0	44.4	73.0	89.9	71.0	67.8	72.4	64.4	59.7	67.8
Wartortle	17M	70.4	82.0	42.4	71.1	46.8	74.6	88.2	54.9	62.3	68.2	65.2	62.5	65.7
gte-micro	17M	68.8	77.1	40.9	69.6	46.2	62.2	86.7	49.7	59.0	66.6	66.1	60.8	62.8
gte-micro-v2	17M	71.4	77.7	39.0	80.4	44.5	70.6	90.5	67.5	68.5	73.5	66.7	59.3	67.5
gte-micro-v4	19M	71.8	80.0	39.8	80.9	44.9	72.0	90.9	68.5	69.1	74.2	66.0	59.4	68.1
snowflake-arctic-embed-xs	23M	65.1	70.0	35.3	76.4	41.8	62.8	90.8	58.0	63.5	71.0	64.3	56.2	62.9
bge-micro	17M	66.3	75.4	35.8	80.6	42.5	70.7	90.2	68.0	67.8	73.0	69.2	56.7	66.3
bge-micro-v2	17M	67.8	79.8	37.5	81.2	44.5	76.5	90.7	68.3	68.6	73.9	70.2	57.6	68.0
gte-tiny	23M	71.8	86.6	<u>42.6</u>	81.7	44.7	80.5	91.8	69.9	70.1	74.9	<u>71.0</u>	58.6	70.3
slx-v0.1	23M	61.5	64.3	30.3	80.0	40.5	61.8	92.0	63.3	67.9	73.9	62.1	54.0	62.6
multi-qa-MiniLM-L6-cos-v1	23M	61.8	62.4	29.6	78.6	39.6	61.2	90.0	59.6	66.8	73.8	65.1	51.6	61.7
all-MiniLM-L6-v2	23M	63.6	64.3	30.9	80.0	40.8	61.8	91.7	61.5	66.9	73.8	62.1	54.0	62.6
MSE Student-xs	23M	71.6	86.2	42.3	83.6	<u>57.5</u>	83.5	94.5	75.4	<u>74.3</u>	80.4	66.3	59.3	72.9
NLL Student-xs	23M	76.5	84.9	42.4	85.8	58.0	<u>81.1</u>	95.2	79.9	75.8	80.4	68.1	60.1	74.0

Table 12: Performance of our distilled models compared to models of similar sizes 30M to 50M parameters from the MTEB Benchmark on classification tasks.

Task Model	Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTOPDomain	MTOPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
bge-small-en-v1.5	33M	73.8	92.8	47.0	85.7	47.8	90.6	93.4	74.8	74.8	78.7	69.9	60.5	74.1
GIST	33M	75.3	<u>93.2</u>	49.7	<u>86.7</u>	55.9	89.5	<u>95.5</u>	79.1	75.5	79.2	72.8	61.0	76.1
NoInstruct	33M	75.8	93.3	50.0	86.4	55.1	<u>90.2</u>	95.3	<u>79.6</u>	<u>76.0</u>	79.3	69.4	61.3	<u>76.0</u>
snowflake-arctic-embed-s	33M	71.2	78.8	38.3	79.1	45.8	69.5	90.9	58.6	64.8	70.0	62.0	58.9	65.7
bge-small-4096	35M	68.8	81.3	38.6	80.0	40.1	80.1	90.4	66.5	67.6	73.5	69.3	57.6	67.8
LASER	43M	76.8	61.0	28.7	57.8	24.8	57.6	75.4	49.5	47.9	55.9	54.0	48.7	53.2
e5-small	33M	76.2	87.5	42.6	81.9	46.9	75.5	92.0	73.2	72.2	75.8	<u>72.8</u>	63.3	71.7
e5-small-v2	33M	77.6	91.3	45.9	81.6	47.1	86.0	92.7	72.6	71.6	76.4	71.1	<u>61.5</u>	72.9
jina-embedding-s-en-v1	35M	64.8	64.3	30.6	74.6	36.1	58.7	88.8	58.6	64.7	71.8	59.4	54.3	60.6
jina-embeddings-v2-small-en	33M	71.4	82.9	40.9	78.2	44.0	73.6	94.0	72.5	67.6	69.8	71.5	59.4	68.8
all-MiniLM-L12-v2	33M	65.3	63.0	30.8	80.4	41.2	59.8	91.9	62.8	67.2	74.6	67.5	54.2	63.2
gte-small	33M	73.2	91.8	48.0	84.1	46.6	86.8	93.0	69.7	70.3	75.6	70.3	58.2	72.3
MSE Student-s	33M	72.6	90.3	44.3	84.2	<u>56.5</u>	88.8	94.9	77.2	75.4	81.2	64.9	60.4	74.2
NLL Student-s	33M	<u>77.3</u>	89.2	43.8	86.7	58.0	88.3	95.5	81.9	76.7	<u>80.7</u>	66.1	60.6	75.4

our embedder are still competitive on these tasks achieving average performance for their respective size categories.

Clustering with very small model. In Tab. 15, we show that our very small model actually outperforms baselines and sits on the pareto frontier for clustering tasks. This is a surprising result as we did not optimize our models for clustering tasks and the embeddings are not designed to have a meaningful structure.

C.2.3 Analysis and compare with the most recent embedders

The results at Tab. 22 show that our medium model (STUDENT-M-NLL, 109M) achieves an average of 80.2 on the selected MTEB classification tasks, tracking much larger recent embedders within single-digit margins. In particular, QWEN3-EMBEDDING-0.6B (595M) reaches 85.8, a +5.6 point gain at $\sim 5.5\times$ the parameters. Substantially larger improvements appear only beyond $\sim 1\text{B}$ parameters (JASPER_EN_VISION_LANGUAGE_V1, 1.0B: 90.3; STELLA_EN_1.5B_V5, 1.5B: 89.4; QWEN3-EMBEDDING-4B, 4.0B: 89.8). Overall, the 109M model delivers competitive accuracy relative to 4–6 $\times$ larger embedders, supporting our claim that multi-teacher distillation yields high information density at compact scales.

Table 13: Performance of our distilled models compared to models of similar sizes 100M to 120M parameters from the MTEB Benchmark on classification tasks.

Task Model	Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTOPDomain	MTOPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
bge-base-en-v1.5	109M	76.2	93.4	48.9	87.0	51.9	90.8	94.2	76.9	76.2	80.2	71.6	59.4	75.5
GIST	109M	76.0	93.5	50.5	87.3	54.7	89.7	95.3	78.1	76.0	79.6	72.4	59.3	76.0
bilingual-embedding-small	118M	74.3	82.2	40.2	80.3	40.8	73.7	89.7	66.5	68.9	74.5	62.5	59.6	67.8
multilingual-e5-small	118M	73.8	88.7	44.7	79.4	42.5	80.8	91.1	71.1	70.3	74.5	69.4	62.6	70.7
snowflake-arctic-embed-m	109M	76.8	82.8	38.9	80.3	46.5	74.1	92.7	65.2	66.9	72.8	64.9	56.7	68.2
snowflake-arctic-embed-m-v1.5	109M	68.3	90.3	46.3	80.0	43.7	84.4	91.4	60.6	66.7	73.1	66.8	53.9	68.8
ml-nlp-elser.html	110M	74.2	61.9	32.1	82.0	46.6	65.0	93.2	71.1	68.5	75.0	68.2	53.6	65.9
e5-base-4k	112M	77.8	92.8	46.7	83.5	47.0	86.2	93.7	75.3	73.0	77.7	72.1	60.4	73.8
instructor-base	110M	86.2	88.4	44.6	77.0	51.8	81.2	93.7	70.3	67.5	72.6	71.8	63.3	72.4
bert-base-uncased	110M	74.2	71.3	33.6	63.4	35.3	65.3	82.6	68.1	59.9	64.3	70.0	51.8	61.7
e5-base	109M	79.7	88.0	42.6	83.3	49.4	76.0	93.2	74.8	72.2	76.8	74.1	61.4	72.6
e5-base-v2	110M	77.8	92.8	46.7	83.5	47.0	86.2	93.7	75.3	73.0	77.7	72.1	60.4	73.8
jina-embedding-b-en-v1	110M	66.7	67.6	31.2	84.1	44.7	63.9	91.5	72.8	71.1	76.2	66.2	56.9	66.1
contriever-base-msmarco	110M	72.2	68.6	37.4	80.0	44.8	67.0	93.2	69.3	67.8	76.0	67.8	56.1	66.7
sup-simcse-bert-base-uncased	110M	75.8	82.5	39.6	75.8	44.8	73.5	84.3	63.1	66.0	70.8	72.0	59.7	67.3
unsup-simcse-bert-base-uncased	110M	67.1	74.5	33.9	73.5	42.2	69.6	81.7	59.2	59.8	66.2	68.8	53.4	62.5
all-mpnet-base-v2	110M	65.0	67.1	31.4	81.7	42.2	71.2	91.9	68.3	69.8	75.7	61.0	55.0	65.0
allenai-specter	110M	58.7	57.8	26.3	66.7	24.8	56.4	74.5	50.0	51.7	58.6	57.4	45.5	52.4
gtr-t5-base	110M	69.3	67.8	38.5	79.3	42.2	66.0	92.4	62.4	67.0	75.4	66.6	56.0	65.3
msmarco-bert-co-condensor	110M	64.1	66.9	34.9	82.3	41.9	60.2	91.3	71.1	70.4	73.7	64.0	55.7	64.7
paraphrase-multilingual-MiniLM-L12-v2	118M	71.5	69.2	35.1	79.8	42.3	60.5	87.0	65.5	66.9	71.5	60.1	56.1	63.8
sentence-t5-base	110M	75.8	85.1	44.9	76.5	51.4	77.3	90.3	63.3	69.7	72.3	68.2	62.7	69.8
text2vec-base-multilingual	118M	71.0	66.1	33.1	78.1	43.4	59.4	81.0	62.8	63.8	67.0	66.0	55.2	62.2
Angle.BERT	109M	77.9	76.0	37.2	75.5	45.2	68.8	85.4	64.5	66.3	70.6	67.1	57.6	66.0
gte-base	109M	74.2	91.8	49.0	85.1	48.6	86.0	93.0	72.0	71.5	76.4	71.6	57.0	73.0
ALL_862873	118M	50.8	52.6	22.6	36.4	22.8	50.8	61.0	29.7	34.3	44.1	54.9	40.8	41.7
MSE Student-m	109M	76.6	89.1	44.7	87.2	60.8	88.0	95.7	81.6	77.7	82.2	67.3	60.5	76.0
NLL Student-m	109M	79.6	89.5	45.8	88.0	59.7	88.3	96.2	83.9	78.6	82.7	67.1	61.3	76.7

Table 14: Performance of our distilled models compared to models of similar sizes 200M to 420M parameters from the MTEB Benchmark on classification tasks.

Task Model	Size	Amazon Counterfactual	Amazon Polarity	Amazon Reviews	Banking77	Emotion	Imdb	MTOPDomain	MTOPIntent	Massive Intent	Massive Scenario	Toxic Conversations	Tweet Sentiment Extraction	Avg.
gte-multilingual-base	305M	76.0	80.7	43.6	85.4	48.0	74.9	92.5	72.6	72.1	76.3	71.0	57.6	70.9
bge-large-en-v1.5	335M	75.8	92.4	48.2	87.8	51.5	92.8	94.6	79.5	77.6	80.5	70.9	59.9	76.0
GIST	335M	75.6	93.4	49.1	88.1	54.7	91.2	95.2	78.2	76.2	79.3	71.9	59.2	76.0
MUG-B-1.6	335M	72.4	93.7	50.9	85.4	55.9	93.6	94.2	67.5	73.9	77.4	67.3	61.8	74.5
bilingual-embedding-base	278M	77.4	89.5	46.1	78.5	47.1	87.4	92.9	64.8	68.9	75.2	63.4	62.5	71.1
snowflake-arctic-embed-l	334M	74.8	78.4	36.7	80.1	46.5	72.9	92.6	64.5	65.8	71.1	64.7	56.7	67.1
UAE-Large-V1	335M	75.5	92.8	48.3	87.7	51.8	92.8	94.0	76.9	76.5	79.8	71.1	59.8	75.6
embedder-100p	278M	67.1	70.4	33.2	82.7	43.5	67.3	91.8	74.7	71.8	77.8	67.5	55.6	67.0
instructor-large	335M	88.1	91.5	47.9	78.5	52.7	88.3	93.9	68.0	68.9	73.3	71.0	64.1	73.9
e5-large	335M	77.7	90.0	43.0	84.1	48.0	82.1	93.9	76.4	73.2	77.4	70.6	61.2	73.1
e5-large-v2	335M	79.2	93.8	48.6	84.5	49.5	91.7	94.6	77.1	73.8	78.1	70.9	60.9	75.2
multilingual-e5-base	278M	77.4	91.8	47.5	73.5	45.7	84.3	90.9	61.6	65.7	71.6	64.3	62.8	69.8
sf_model_e5	335M	70.8	91.8	48.9	84.6	54.9	93.1	93.6	66.0	73.5	77.4	71.2	61.5	74.0
jina-embedding-l-en-v1	335M	68.9	69.1	31.4	85.3	45.8	66.4	92.8	76.1	72.7	77.1	69.1	58.2	67.8
ember-v1	335M	76.1	92.0	47.9	87.9	52.0	92.8	94.6	79.3	77.4	80.5	71.4	60.0	76.0
mxbai-embed-2d-large-v1	335M	74.8	93.3	46.2	86.7	49.3	90.4	93.1	73.2	73.9	78.2	71.5	59.2	74.1
mxbai-embed-large-v1	335M	75.0	93.8	49.2	87.8	50.9	92.8	94.0	76.8	76.2	80.0	71.5	59.7	75.6
paraphrase-multilingual-mpnet-base-v2	278M	75.8	76.4	38.5	81.1	45.8	64.6	89.2	68.7	69.3	75.3	71.0	59.0	67.9
gte-large	335M	72.6	92.5	49.1	86.1	47.9	88.5	93.5	73.2	72.6	76.8	70.6	56.6	73.3
b1ade-embed	335M	75.2	93.1	48.4	88.0	51.9	91.9	94.3	76.6	75.9	79.4	67.9	59.2	75.2
MSE Student-l	335M	77.3	84.5	43.4	86.0	60.0	82.7	95.1	79.8	76.3	81.3	65.8	60.2	74.4
NLL Student-l	335M	81.5	88.1	45.9	86.9	60.4	88.2	95.6	83.2	77.5	81.4	67.7	62.2	76.5

Table 15: Performance of our distilled models compared of models of similar sizes 16M to 30M parameters from the MTEB Benchmark on clustering tasks.

	Task Model	Size	Arxiv Clustering P2P	Arxiv Clustering S2S	Reddit Clustering P2P	Reddit Clustering	Stack Exchange Clustering P2P	Stack Exchange Clustering	Twenty Newsgroups Clustering	Avg.
MTEB	Bulbasaur	17M	40.3	31.1	51.4	45.9	30.7	52.2	39.4	41.6
	Ivysaur	23M	46.4	35.4	56.0	47.5	33.6	53.9	40.8	44.8
	Squirtle	16M	33.0	24.7	43.7	31.4	29.2	39.2	28.2	32.8
	Venusaur	16M	31.8	21.1	44.1	26.7	27.5	32.8	26.1	30.0
	Wartortle	17M	35.8	27.3	46.1	35.9	29.9	45.3	31.7	36.0
	gte-micro	17M	35.2	31.1	47.9	45.6	30.1	52.6	40.8	40.5
	gte-micro-v4	19M	42.9	32.5	53.6	48.3	31.9	55.1	41.4	43.6
	snowflake-arctic-embed-xs	23M	43.5	32.1	<u>57.8</u>	48.3	34.6	57.5	36.3	44.3
	bge-micro	17M	44.6	34.5	54.5	45.3	<u>34.7</u>	53.1	39.4	43.7
	bge-micro-v2	17M	44.5	33.2	55.2	45.5	34.1	54.5	40.2	43.9
	gte-tiny	23M	46.6	36.0	56.5	50.2	35.7	<u>57.5</u>	43.3	<u>46.6</u>
	GIST-all-MiniLM-L6-v2	23M	45.3	35.5	48.7	44.1	33.9	53.1	41.1	43.1
	slx-v0.1	23M	46.5	<u>37.7</u>	54.8	50.7	34.2	53.1	46.5	46.2
	multi-qa-MiniLM-L6-cos-v1	23M	37.8	27.7	51.0	46.3	33.4	48.1	40.8	40.7
	all-MiniLM-L6-v2	23M	<u>46.5</u>	37.9	54.8	<u>50.7</u>	34.3	53.1	46.5	46.3
	rubert-tiny-turbo	29M	24.8	16.7	40.5	26.3	28.0	33.5	19.9	27.1
MSE	Student-xs	23M	42.4	30.9	55.2	49.2	32.7	53.5	41.9	43.7
NLL	Student-xs	23M	45.2	33.9	58.1	52.1	33.1	59.9	44.3	46.7

Table 16: Performance of our distilled models compared of models of similar sizes 30M to 50M parameters from the MTEB Benchmark on clustering tasks.

	Task Model	Size	Arxiv Clustering P2P	Arxiv Clustering S2S	Reddit Clustering P2P	Reddit Clustering	Stack Exchange Clustering P2P	Stack Exchange Clustering	Twenty Newsgroups Clustering	Avg.
MTEB	bge-small-en-v1.5	33M	47.4	40.0	60.6	52.3	35.3	60.8	48.5	49.3
	snowflake-arctic-embed-s	33M	44.9	35.9	60.5	50.5	34.0	60.7	38.3	46.4
	bge-small-4096	35M	43.9	29.6	54.3	43.7	33.3	51.8	36.6	41.9
	GIST-small-Embedding-v0	33M	47.6	39.9	60.6	<u>55.5</u>	36.2	61.9	50.0	50.2
	NoInstruct-small-Embedding-v0	33M	<u>47.8</u>	<u>40.1</u>	<u>61.2</u>	55.4	36.6	<u>62.0</u>	49.9	<u>50.4</u>
	e5-small	33M	44.1	37.1	57.2	43.3	30.8	59.6	37.6	44.3
	e5-small-v2	33M	42.1	34.8	59.7	45.7	32.0	58.5	41.1	44.8
	jina-embedding-s-en-v1	35M	34.2	24.0	49.9	38.0	31.5	46.4	34.4	36.9
	jina-embeddings-v2-small-en	33M	44.0	35.2	57.1	49.3	34.4	55.4	41.6	45.3
	all-MiniLM-L12-v2	33M	46.1	37.5	54.8	51.2	33.1	53.0	47.5	46.2
	gte-small	33M	47.9	40.3	61.4	55.6	<u>36.3</u>	62.6	50.0	50.6
MSE	Student-s	33M	43.1	33.3	57.1	50.8	32.3	55.7	42.8	45.0
NLL	Student-s	33M	45.9	35.2	60.3	51.9	32.3	61.5	45.1	47.4

Table 17: Performance of our distilled models compared of models of similar sizes 100M to 120M parameters from the MTEB Benchmark on clustering tasks.

Task Model	Size	Arxiv Clustering P2P	Arxiv Clustering S2S	Reddit Clustering P2P	Reddit Clustering	Stack Exchange Clustering P2P	Stack Exchange Clustering	Twenty Newsgroups Clustering	Avg.	
MTEB	bge-base-en-v1.5	109M	48.8	42.8	62.7	56.6	35.2	66.1	50.8	51.8
	bilingual-embedding-small	118M	41.8	31.6	58.4	47.4	33.6	52.5	40.5	43.7
	multilingual-e5-small	118M	39.2	30.8	59.0	39.1	32.1	53.5	33.2	41.0
	snowflake-arctic-embed-m	109M	47.2	37.4	62.8	47.5	39.4	59.5	37.7	47.4
	snowflake-arctic-embed-m-v1.5	109M	45.0	34.1	61.8	51.9	33.8	61.2	38.1	46.6
	GIST-Embedding-v0	109M	48.3	42.7	62.4	59.1	35.6	<u>66.1</u>	<u>52.2</u>	<u>52.4</u>
	ml-nlp-elser.html	110M	35.3	23.2	51.9	38.7	28.7	42.7	27.8	35.5
	e5-base-4k	112M	46.1	39.7	63.4	56.2	32.5	65.2	48.2	50.2
	instructor-base	110M	39.7	29.2	63.2	<u>59.3</u>	35.3	65.0	51.3	49.0
	bert-base-uncased	110M	35.2	27.5	43.3	27.2	26.6	43.6	23.4	32.4
	e5-base	109M	44.6	40.5	62.2	48.2	32.6	63.9	42.6	47.8
	e5-base-v2	110M	46.1	39.7	63.2	56.5	33.0	64.6	49.9	50.4
	jina-embedding-b-en-v1	110M	39.2	29.1	<u>52.5</u>	42.9	31.4	48.1	38.1	40.2
	contriever-base-msmarco	110M	42.6	32.3	57.6	54.9	32.2	63.1	46.8	47.1
	sup-simcse-bert-base-uncased	110M	35.2	27.5	47.7	40.2	29.4	47.5	34.9	37.5
	unsup-simcse-bert-base-uncased	110M	32.6	24.7	45.1	32.2	28.5	43.1	23.2	32.8
	all-mpnet-base-v2	110M	48.4	39.7	56.8	54.8	34.3	53.8	49.7	48.2
	allenai-specter	110M	44.8	35.3	35.1	24.1	31.5	39.0	24.2	33.4
	gtr-t5-base	110M	35.5	27.2	58.5	56.1	33.0	64.2	46.7	45.9
	msmarco-bert-co-condensor	110M	36.9	29.0	53.5	48.0	30.5	59.5	38.7	42.3
	paraphrase-multilingual-MiniLM-L12-v2	118M	38.3	31.6	50.1	42.6	31.7	49.3	40.0	40.5
	sentence-t5-base	110M	39.3	27.3	59.7	52.9	35.7	63.1	48.1	46.6
	text2vec-base-multilingual	118M	32.3	25.5	43.3	31.2	30.6	34.4	31.6	32.7
	Angle_BERT	109M	35.3	27.7	46.0	40.3	28.9	48.3	33.1	37.1
	gte-base	109M	<u>48.6</u>	43.0	62.6	59.3	<u>36.0</u>	66.6	52.3	52.6
	ALL_862873	118M	14.8	12.2	27.1	18.4	27.3	23.7	20.2	20.5
MSE	Student-m	109M	46.5	37.1	60.4	54.5	33.4	62.0	46.1	48.6
NLL	Student-m	109M	47.7	38.7	61.5	56.3	33.8	64.7	46.6	49.9

Table 18: Performance of our distilled models compared of models of similar sizes 16M to 30M parameters from the MTEB Benchmark on STS tasks.

		Size	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS17	STS22	STSBenchmark	Avg.
MTEB	Bulbasaur	17M	85.0	76.0	69.5	81.0	77.1	85.4	82.3	88.0	64.1	83.3	79.2
	Ivysaur	23M	87.3	75.6	68.6	80.5	77.6	86.2	82.8	<u>88.6</u>	<u>67.4</u>	84.2	79.9
	Squirtle	16M	71.8	77.3	70.2	78.4	74.8	82.0	78.3	85.8	61.2	79.2	75.9
	Venusaur	16M	77.6	74.7	54.4	74.2	70.0	75.7	73.7	84.8	62.6	76.7	72.4
	Wartortle	17M	80.8	78.2	75.2	79.3	76.6	84.7	81.4	86.6	63.4	81.8	78.8
	snowflake-arctic-embed-xs	23M	84.0	69.3	65.9	77.9	72.8	83.5	80.6	84.5	66.3	79.2	76.4
	bge-micro	17M	83.4	72.4	71.9	80.9	76.6	84.9	80.7	85.6	65.9	81.3	78.4
	bge-micro-v2	17M	82.9	73.6	71.9	79.8	76.9	84.8	81.9	86.8	65.4	82.5	78.7
	gte-tiny	23M	<u>86.6</u>	75.8	72.6	<u>82.4</u>	<u>78.0</u>	<u>86.5</u>	83.3	88.3	66.7	<u>84.4</u>	<u>80.5</u>
	GIST-all-MiniLM-L6-v2	23M	81.3	79.1	<u>75.0</u>	83.3	78.6	87.0	<u>83.0</u>	87.4	68.1	84.4	80.7
	multi-qa-MiniLM-L6-cos-v1	23M	79.8	70.0	64.4	76.4	69.3	80.2	79.6	81.2	65.5	76.0	74.2
	all-MiniLM-L6-v2	23M	81.6	77.6	72.4	80.6	75.6	85.4	79.0	87.6	67.2	82.0	78.9
MSE	Student-xs	23M	76.8	79.2	72.2	80.3	75.9	85.0	83.0	87.1	66.4	82.9	78.9
NLL	Student-xs	23M	78.8	77.8	71.6	80.2	77.0	85.8	82.8	89.3	65.8	83.5	79.3

Table 19: Performance of our distilled models compared of models of similar sizes 30M to 50M parameters from the MTEB Benchmark on STS tasks.

		STSBenchmark										Avg.	
Task Model	Size	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS17	STS22			
MTEB	bge-small-en-v1.5	33M	83.8	79.4	77.4	83.0	81.8	87.3	84.9	87.2	65.3	85.9	81.6
	snowflake-arctic-embed-s	33M	86.3	69.7	68.8	79.6	75.6	84.6	82.4	86.7	69.5	81.2	78.4
	bge-small-4096	35M	81.6	74.2	72.2	80.5	76.2	85.2	81.9	86.6	65.5	81.9	78.6
	GIST-small-Embedding-v0	33M	87.0	80.5	75.6	86.3	82.3	88.7	85.3	89.0	68.5	87.1	83.0
	NoInstruct-small-Embedding-v0	33M	<u>87.2</u>	<u>80.3</u>	75.8	<u>86.1</u>	<u>82.3</u>	88.9	<u>85.2</u>	88.7	<u>68.5</u>	<u>87.0</u>	<u>83.0</u>
	e5-small	33M	84.2	78.9	75.2	81.8	78.5	87.5	84.6	87.9	63.8	86.4	80.9
	e5-small-v2	33M	79.4	78.5	<u>76.2</u>	82.4	79.0	87.8	83.8	87.7	63.1	86.0	80.4
	jina-embedding-s-en-v1	35M	83.0	76.3	74.3	78.5	73.8	83.7	80.0	87.5	64.2	79.2	78.1
	jina-embeddings-v2-small-en	33M	80.5	76.7	73.7	83.3	79.2	87.3	83.6	88.2	63.5	84.0	80.0
	all-MiniLM-L12-v2	33M	83.6	79.3	73.1	82.1	76.7	85.6	80.2	88.6	65.7	83.1	79.8
gte-small	33M	88.2	77.9	75.1	85.1	81.0	88.3	83.9	87.6	68.0	85.6	82.1	
MSE	Student-s	33M	78.9	79.5	70.6	79.7	75.4	84.1	81.8	86.7	66.6	83.1	78.6
NLL	Student-s	33M	81.5	79.3	73.0	81.4	78.2	86.3	84.2	90.0	66.0	84.8	80.5

Table 20: Performance of our distilled models compared of models of similar sizes 100M to 120M parameters from the MTEB Benchmark on STS tasks.

Task Model	Size	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS17	STS22	STSBenchmark	Avg.	
MTEB	bge-base-en-v1.5	109M	86.9	80.3	78.0	84.2	82.3	88.0	85.5	86.4	66.0	86.4	82.4
	bilingual-embedding-small	118M	84.0	74.7	79.4	85.3	83.9	88.5	84.4	85.8	67.2	86.1	81.9
	multilingual-e5-small	118M	82.3	77.5	76.6	77.0	75.5	87.1	83.6	86.4	60.9	84.0	79.1
	snowflake-arctic-embed-m	109M	86.6	69.1	67.0	79.1	68.5	79.9	78.7	81.5	65.8	74.1	75.0
	snowflake-arctic-embed-m-v1.5	109M	86.4	69.9	61.8	82.7	69.0	75.5	77.3	75.0	69.1	69.7	73.6
	GIST-Embedding-v0	109M	88.0	81.3	76.2	87.8	83.4	89.4	85.3	88.6	67.8	87.3	83.5
	ml-nlp-elser.html	110M	83.8	68.8	64.8	80.1	75.0	83.7	80.5	85.7	67.5	79.5	76.9
	e5-base-4k	112M	81.4	78.3	75.8	83.6	80.0	88.8	84.5	87.6	64.1	86.5	81.0
	instructor-base	110M	82.3	80.3	77.0	86.6	81.3	88.2	84.9	89.5	66.5	86.4	82.3
	bert-base-uncased	110M	54.7	58.6	30.9	59.9	47.7	60.3	63.7	64.1	56.4	47.3	54.4
	e5-base	109M	85.1	79.7	74.2	83.3	78.5	88.3	84.2	87.2	62.9	86.2	81.0
	e5-base-v2	110M	81.4	78.3	75.8	83.6	80.0	88.8	84.5	87.6	64.1	86.5	81.0
	jina-embedding-b-en-v1	110M	83.6	79.1	75.1	80.9	76.1	85.5	81.2	89.0	66.2	82.6	79.9
	contriever-base-msmarco	110M	83.3	70.2	64.3	80.0	74.5	83.3	79.7	86.3	64.6	78.8	76.5
	sup-simcse-bert-base-uncased	110M	68.4	80.8	75.3	84.7	80.2	85.4	80.8	89.4	62.0	84.2	79.1
	unsup-simcse-bert-base-uncased	110M	72.3	72.2	66.0	81.5	73.6	79.7	78.1	83.6	59.6	76.5	74.3
	all-mpnet-base-v2	110M	80.4	80.6	72.6	83.5	78.0	85.7	80.0	90.6	68.0	83.4	80.3
	allenai-specter	110M	65.0	56.4	62.5	58.7	54.9	62.5	64.3	69.6	55.1	61.3	61.0
	gtr-t5-base	110M	79.0	71.5	68.6	79.1	74.6	84.8	81.6	85.8	66.2	79.6	77.1
	msmarco-bert-co-condensor	110M	77.3	72.0	68.2	80.4	74.0	82.6	79.8	85.9	67.5	77.0	76.5
	paraphrase-multilingual-MiniLM-L12-v2	118M	74.2	79.6	76.0	80.7	78.8	85.8	81.0	86.9	62.1	84.4	79.0
	sentence-t5-base	110M	75.9	80.2	78.0	85.8	82.2	87.5	84.0	89.6	62.7	85.5	81.1
	text2vec-base-multilingual	118M	66.2	80.0	80.9	82.9	87.4	88.3	81.6	85.8	63.0	86.5	80.2
	gte-base	109M	87.6	78.9	75.7	85.7	81.5	88.8	83.8	87.9	67.3	85.7	82.3
	ALL-862873	118M	21.3	48.5	55.6	18.4	28.8	29.2	39.0	61.2	44.5	44.4	39.1
MSE	Student-m	109M	83.4	80.9	74.5	82.8	79.0	86.6	85.2	88.4	66.4	85.2	81.2
NLL	Student-m	109M	85.2	80.2	75.2	83.4	80.4	88.3	86.0	89.9	66.2	86.4	82.1

Table 21: Performance of our distilled models compared of models of similar sizes 200M to 400M parameters from the MTEB Benchmark on STS tasks.

			STSBenchmark										
Task Model	Size	BIOSSES	SICK-R	STS12	STS13	STS14	STS15	STS16	STS17	STS22		Avg.	
MTEB	gte-multilingual-base	305M	81.2	79.3	77.5	85.5	81.7	89.0	84.3	88.9	67.2	86.5	82.1
	bge-large-en-v1.5	335M	84.7	81.7	79.0	86.4	82.8	88.0	86.5	87.5	67.0	87.5	83.1
	MUG-B-1.6	335M	88.4	83.0	<u>79.2</u>	89.4	84.8	89.5	86.7	89.6	70.3	89.0	85.0
	bilingual-embedding-base	278M	87.1	<u>79.5</u>	79.6	84.7	83.9	<u>89.9</u>	84.9	88.7	<u>64.3</u>	87.4	83.0
	snowflake-arctic-embed-l	334M	86.3	69.3	67.8	77.5	69.8	80.2	77.9	82.3	68.0	75.7	75.5
	UAE-Large-V1	335M	86.1	82.6	79.1	89.6	85.0	89.5	86.6	89.0	68.8	89.1	84.5
	GIST-large-Embedding-v0	335M	89.2	82.8	77.1	89.3	83.8	89.7	86.4	89.7	69.6	88.3	84.6
	embedder-100p	278M	75.3	80.9	77.0	82.6	77.8	85.9	80.7	89.0	68.3	84.2	80.2
	instructor-large	335M	84.4	81.3	76.3	88.2	81.9	89.0	85.5	90.3	67.7	86.9	83.1
	e5-large	335M	84.7	80.5	75.9	85.2	80.5	88.8	85.3	89.4	63.0	87.2	82.1
	e5-large-v2	335M	83.6	79.3	77.0	84.1	80.5	89.8	85.5	89.0	64.1	87.7	82.1
	multilingual-e5-base	278M	85.0	78.5	76.7	78.0	76.6	88.2	84.3	87.8	62.3	85.6	80.3
	sf_model_e5	335M	86.8	82.3	77.6	88.0	83.8	88.5	86.5	88.7	68.0	88.3	83.8
	jina-embedding-l-en-v1	335M	84.4	79.2	74.5	83.2	78.1	86.9	83.7	<u>90.2</u>	64.9	84.6	81.0
	ember-v1	335M	85.8	81.8	78.5	86.6	83.1	88.4	<u>86.8</u>	87.9	66.8	87.8	83.3
	mxbai-embed-2d-large-v1	335M	88.1	82.0	78.8	90.4	<u>85.5</u>	90.0	87.4	88.8	68.8	89.2	84.9
	mxbai-embed-large-v1	335M	88.4	82.9	78.8	<u>90.3</u>	85.5	89.6	86.6	89.5	69.3	<u>89.1</u>	<u>85.0</u>
	paraphrase-multilingual-mpnet-base-v2	278M	76.3	79.6	77.9	85.1	80.8	87.5	83.2	87.0	63.5	86.8	80.8
	gte-large	335M	88.7	79.8	76.8	88.1	82.7	88.9	84.2	88.5	<u>69.7</u>	86.1	83.3
	blade-embed	335M	<u>89.2</u>	82.8	78.7	90.0	85.0	89.8	86.7	89.8	69.7	88.8	85.0
MSE	Student-l	335M	79.1	80.6	<u>73.7</u>	82.1	78.1	87.4	84.2	89.1	67.0	85.3	80.7
NLL	Student-l	335M	83.8	79.5	74.4	83.0	79.6	88.0	85.2	90.1	65.3	86.2	81.5

Table 22: Head-to-head comparison on selected MTEB classification tasks, with large embedders (over x5 times the number of parameters).

		AmazonCtf	Banking77	IMDB	MTOP Dom.	Massive Int.	Massive Scen.	Toxic Conv.	Tweet Sent.		
Model	Size									Avg.	
Qwen3-Embedding-4B	4.0B	93.7	86.3	97.2	97.8	85.0	88.8	91.4	78.4	89.8	
stella_en.1.5B_v5	1.5B	94.1	89.8	96.7	98.7	84.5	89.7	86.8	74.8	89.4	
jasper_en.vision.language.v1	1.0B	93.8	87.2	97.0	99.2	85.3	91.2	91.3	77.2	90.3	
Qwen3-Embedding-0.6B	595M	91.5	81.0	95.4	96.0	80.4	83.6	82.1	76.0	85.8	
jina-embeddings-v3	572M	90.9	84.1	91.9	–	75.2	84.1	91.3	71.4	84.1	
snowflake-arctic-embed-l-v2.0	568M	65.6	81.8	72.8	93.5	71.5	76.2	65.9	59.6	73.4	
KaLM-embed-mini-instr-v2	494M	95.3	89.5	95.2	98.9	77.8	86.0	89.3	78.6	88.8	
KaLM-embed-mini-instr-v1	494M	81.5	84.9	95.0	92.2	69.8	74.2	89.0	76.5	82.9	
KaLM-embed-mini-v1	494M	76.4	79.2	91.6	92.5	70.9	76.1	70.8	62.7	77.5	
stella_en.400M_v5	435M	94.3	89.3	96.5	98.3	80.5	89.6	84.0	73.6	88.2	
NLL	Student-m-nll	109M	79.6	88.0	88.3	96.2	78.6	82.7	67.1	61.3	80.2
	Student-s-nll	32M	77.3	86.7	88.3	95.5	76.7	80.7	66.1	60.6	79.0

D Vision

D.1 Model architecture

The models we used for vision as teachers and student are presented in Tab. 23, including the number of parameters of each of them.

D.2 Training Set

Tab. 24 presents the statistics, *i.e.* the number of training and testing samples, of the datasets we used for vision.

D.3 Vision Details

Data processing details: We use the official train sets of the datasets for the knowledge distillation part. We split the official training part, if there are no official validation sets, to train and validation set with 80 and 20 percents of the data, consequently. For the augmentation we used color jitter with brightness, contrast, saturation and hue equal to 0.2, and random horizontal flip (except for the SVHN dataset).

Distillation details: For training the distillation, we extract the embeddings of the train set of each dataset, for each teacher and divide the embeddings to 80 train set and 20 percent validation set. For the optimizer we use Adam, with learning rate of 0.001, a batch size of 128, trained for 50 epochs.

Down-stream task fine-tuning: For fine-tuning of down-stream tasks, we add a classifier on the frozen embedders. We again use Adam optimizer for the fine-tuning of downstream tasks. We perform hyperparameter tuning using grid search to optimize the performance of our models. Our search space includes the learning rate with values (1e-2, 1e-3), the number of fully connected layer units with values (0, 128), and the type of normalization after the fully connected layer, considering (no optimization, batch normalization, layer normalization). The models are trained for a maximum of 1000 epochs with a batch size of 128, but we apply early stopping with a patience of 20 to prevent over-fitting and reduce unnecessary computation.

D.4 Complementary Results

Tab. 25 shows the detailed results of the Vision Transformer teachers and students. The best among the students are shown with an underline, showing that on average and most of the cases our method improves the baseline. In addition to the main results, we added additional experiments to answer further informative question:

Table 23: Number of parameters for each model (in million parameters)

Model	# Parameters
Swin (Liu et al., 2021b)	87.77M
DINOv2 (Oquab et al., 2023)	86.58M
ViT (Dosovitskiy et al., 2021)	86.57M
BEiT (Bao et al., 2022)	86.53M
PVTv2 (Wang et al., 2022c)	3.67M
WideResNet (Zagoruyko & Komodakis, 2017)	68.88M
DenseNet (Huang et al., 2017)	28.68M
ResNext (Xie et al., 2017)	25.03M
ResNet18 (He et al., 2016)	11.69M
GoogLeNet (Szegedy et al., 2015)	6.62M
MNASNet (Tan et al., 2019)	4.38M
MobileNet (Sandler et al., 2018)	3.50M
ShuffleNet (Ma et al., 2018)	2.28M
SqueezeNet (Iandola et al., 2016)	1.25M

Table 24: Number of classes, training, validation (if any) and testing samples in each vision dataset

Dataset	classes	training samples	validation samples	test samples
CIFAR10 (Krizhevsky et al., 2009)	10	50000	-	10000
STL10 (Coates et al., 2011)	10	5000	-	8000
SVHN (Netzer et al., 2011)	10	73257	-	26032
CUB (Welinder et al., 2010)	200	5,994	-	5,794
DTD (Cimpoi et al., 2014)	47	1880	1880	1880
FGVCAircraft (Maji et al., 2013)	100	3334	3333	3333
Oxford Pets (Parkhi et al., 2012)	37	3680	-	8041
Food101 (Bossard et al., 2014)	101	750	-	250
Stanford Cars (Krause et al., 2013)	196	8144	-	8041

Table 25: Comparison of Vision Transformer teachers, CNN baselines and the ViT student, with their corresponding parameter size, with the underline showing the best students.

Method	Model	# Parameters	CIFAR10	DTD	STL10	SVHN	FGVCAircraft	CUB
NoKD	Swin	87.77	97.67	76.33	99.60	64.42	52.45	87.11
	ViT	86.57	96.90	71.65	99.40	54.97	41.71	82.67
	DINOv2	86.58	98.57	83.30	99.45	63.01	79.40	89.02
	BEiT	86.53	97.89	77.34	99.60	66.61	55.45	39.52
	PVTv2	3.67	89.27	65.05	95.80	62.03	38.58	68.97
	wide resnet	68.88	85.65	65.37	95.85	57.77	30.82	60.55
	densenet	28.68	87.49	67.93	97.11	66.91	46.84	68.62
	resnet18	11.69	83.22	61.54	92.98	51.01	36.09	59.89
	googlenet	6.62	82.07	66.38	93.95	55.90	35.85	59.09
CompRes	PVTv2	3.67	94.6	52.7	93.5	61.9	32.7	48.8
MSE	PVTv2	3.67	<u>96.1</u>	65.1	96.4	70.3	34.4	67.7
Cosine	PVTv2	3.67	95.89	65.4	<u>96.7</u>	70.7	35.9	67.1
RKD	PVTv2	3.67	87.64	52.23	89.63	61.66	30.54	47.85
CC grbf	PVTv2	3.67	84.07	61.86	93.03	59.96	33.48	57.55
CC bilinear	PVTv2	3.67	92.95	61.22	95.42	63.71	35.16	64.70
NLL	PVTv2	3.67	94.76	<u>65.85</u>	96.45	76.91	<u>48.13</u>	<u>69.37</u>

How will our method work in vision for unseen datasets? Tab. 26 shows the accuracy of our student compared to various distillation baselines: MSE distillation, Cosine distillation, Correlation Congruence (CC rbf and CC dot) Peng et al. (2019), CompRes Abbasi Koohpayegani et al. (2020) and relational KD Park et al. (2019b).

for three unseen datasets. As we can see, our method improved the baselines considerably for unseen datasets.

How our method works for a setting with diverse teachers specialized in different task, and if it will be able to avoid conflicts? We evaluated the student model’s classification performance using three specialized vision teachers: ViT (classification), DETR (Carion et al., 2020), object detection),

Table 26: Comparison of ViT student of our method (NLL), and various distillation baselines for the unseen datasets.

Method	Oxford Pets	Food101	Stanford Cars
CompRes	70.23	45.48	19.43
MSE	85.58	58.04	31.96
Cosine	84.38	56.37	30.92
RKD	69.99	43.48	18.24
CC rbf	85.09	58.47	30.08
CC dot	67.42	45.93	20.88
NLL	87.46	62.62	41.29

and SegFormer (Xie et al., 2021), segmentation). We also included DINOv2, a general-purpose embedding model known for strong performance across multiple benchmarks. As shown in Tab. 27, adding DETR or SegFormer alongside ViT did not significantly improve or degrade classification performance compared to using ViT alone. This suggests that while task-specific teachers may offer limited benefit outside their domain, they do not negatively impact the student’s learning.

To further validate this, we incorporated DINOv2 into the teacher set (Tab. 28). This addition improved overall performance, while the inclusion of DETR and SegFormer continued to have minimal effect, confirming that our earlier observations hold even in a more competitive setting with a strong general-purpose teacher. These results are consistent with Sec. 5.2 and Figure C.1.3, where we observe that adding teachers typically boosts student performance. In molecular and text domains, where all teachers are general-purpose embedders, improvements are more uniform. However, in vision tasks, specialized teachers contribute gains primarily in their area of expertise, yet without harming performance elsewhere. Overall, these findings suggest that our method can effectively integrate knowledge from both specialized and generalist teachers without conflict.

Table 27: Performance of different teacher combinations across datasets (accuracy %).

Teachers	CIFAR-10	DTD	STL-10	SVHN	FGVC	CUB	Average
ViT + Segformer + DETR	94.03	63.62	95.86	65.63	38.79	67.67	70.93
ViT + Segformer	94.23	63.24	95.91	65.79	38.31	67.35	70.81
ViT + DETR	94.71	61.28	95.80	64.14	37.89	65.90	69.95
ViT	94.69	61.70	95.75	64.13	39.42	69.23	70.82
DETR + Segformer	87.87	63.72	94.81	54.71	37.89	62.43	66.91

Table 28: Comparison of ViT-based teacher combinations including DINO on multiple datasets (accuracy %). Bolded values indicate best per column.

Teachers	CIFAR-10	DTD	STL-10	SVHN	FGVC	CUB	Average
ViT + Segformer + DETR + DINO	95.39	64.31	96.14	72.88	50.38	69.69	74.80
ViT + DINO	95.83	61.92	96.06	73.60	50.59	69.21	74.54

As another additional experiment, we use CNN based teachers for resnet18, for different relevant datasets. Tab. 29 shows the performance improvements, and the effectiveness of using our distillation method, compared to other.

Table 29: Comparison of the performance with CNN-based teacher (accuracy %). Bolded values indicate best per column.

Method	Model	CIFAR10	FMNIST	MNIST	STL10	SVHN	QMNIST	KMNIST	CelebA
NoKD	resnet18	81.89	86.94	96.6	92.98	51.01	96.89	80.43	90.82
	squeezenet	79.23	86.65	97.51	85.82	47.77	97.59	84.05	61.35
	densenet	87.49	88.69	96.80	97.11	66.91	97.72	86.33	93.98
	googlenet	81.94	86.38	96.71	93.95	55.9	97.2	79.27	92.93
	shufflenet	81.61	87.57	95.77	71.51	49.08	95.96	76.97	92.42
	mobilenet	81.67	88.07	96.05	92.26	48.57	97.5	85.64	91.02
	mnasnet	81.41	88.76	96.09	92.79	57.63	97.00	82.35	89.01
	resnext50-32x4d	83.42	87.32	95.37	<u>95.97</u>	52.87	96.65	83.37	91.74
	wide-resnet50-2	84.30	87.40	95.16	95.85	57.77	96.74	76.23	90.22
Cosine	resnet18	84.57	<u>89.90</u>	<u>98.58</u>	88.34	<u>76.34</u>	<u>98.95</u>	<u>91.97</u>	<u>95.00</u>
L2	resnet18	82.90	89.75	98.25	88.15	74.84	98.61	88.21	94.89
NLL	resnet18	87.51	90.64	99.15	88.45	81.99	99.15	95.21	95.47

E Detailed Method

Algorithm 1 Distillation through Gaussian Kernels

Input: Dataset $D = \{\mathbf{x}_i\}$, Embedders $(\mathbf{T}_k)_{1 \leq k \leq K}$, Student embedder S , Number of iterations T , Learning rate η
Initialize the parameters θ_s of the student embedder E_s and the parameters θ_k of the parametric Gaussian kernels
for $t = 1$ to T **do**
 Sample a batch of inputs $\{\mathbf{x}_i\}$
 Compute the embeddings $\{\mathbf{t}_i^k = \mathbf{T}_k(\mathbf{x}_i)\}_{1 \leq k \leq K}$
 Compute the student embeddings $\{\mathbf{s}_i = S(\mathbf{x}_i)\}$
 Compute the loss $\mathcal{L}_{NLL} = -\sum_{k=1}^K \sum_{i=1}^N \log \mathcal{N}(\mathbf{t}_i^k | \mu_k(\mathbf{s}_i), \Sigma_k(\mathbf{s}_i))$
 Update the parameters θ_s and θ_k using the Adam optimizer.
end for

F Computaional ressources

Our experiments were conducted in single GPUs settings. We used NVIDIA V100 GPUs for about 3000 GPUs hours to train our different models.

G Baselines

For the MSE, we will optimize the following loss function following SimReg strategy (Navaneet et al., 2022).

$$\mathcal{L}_{MSE} = -\sum_{k=1}^K \sum_{i=1}^N \|\mathbf{S}(\mathbf{x}_i) - \mathbf{T}_k(\mathbf{x}_i)\|^2, \quad (7)$$

where it calculates the summation of MSE between the representation produced by each teacher and the student, for each instance of the batch.

Variant of SimReg can be implemented for Cosine multi-teacher feature distillation(Gao et al., 2022; Navaneet et al., 2022), we optimize the summation of cosine of teachers and the students representations of each instance of the batch, *i.e.*:

$$\mathcal{L}_{Cosine} = -\sum_{k=1}^K \sum_{i=1}^N \frac{\mathbf{S}(\mathbf{x}_i) \cdot \mathbf{T}_k(\mathbf{x}_i)}{\max(\|\mathbf{S}(\mathbf{x}_i)\|_2 \cdot \|\mathbf{T}_k(\mathbf{x}_i)\|_2, \epsilon)}. \quad (8)$$

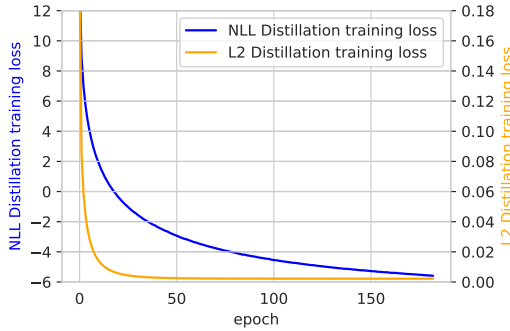


Figure 12: Training curves for the MSE baseline and the NLL student for the molecular experiments.

H Discussion On MSE distillation

We observed that when training with the MSE loss, the loss reaches a minimum in only a few epochs (40), but the distilled students achieve lower performances on downstream tasks. This could be due to the fact that the NLL loss is more expressive, and harder to optimize (see below). As a result the student learns more informative features compared to when trained with the MSE loss (Figure 12).

We can provide a theoretical insight to explain this phenomenon. Training using the negative log-likelihood over a Gaussian kernel is a simple generalization of the MSE. For a given multivariate Gaussian kernel parameterized by μ and Σ , we have:

$$-\log(p_{\mu,\Sigma}(x)) = \log(C) + \frac{1}{2} \log \det \Sigma + \frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)$$

Minimizing the MSE loss boils down to minimizing this equation over only, with $\Sigma = I$. Therefore, minimizing the negative log-likelihood of a Gaussian kernel is strictly more expressive than minimizing the MSE directly, which could account for the performance gains we observe.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

Answer: [\[Yes\]](#)

Justification: We propose an objective to train model through multi teacher distillation, give theoretical grounding to it and validate it experimentally that it works by training embedders in 3 modalities for a wide range of size and domains.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, see for example "Embedding space structure" Sec. 4.2, and "Computational complexity." in Sec. 5.2.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We provide proofs for our corollary in the supplementary materials. See Sec. 3.1 and Appendix E.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: All datasets, checkpoints and architectures are open source and under permissive licences. The code for our particular experiments is available in different repositories for each modalities with the specific hyperparameters and how to rerun the experiments. Our code is publicly available through anonymous links provided in the core of the paper. Regardless of the code we provided the details of datasets, models, hyperparameters and tuning setting (see Sec. D.2, Sec. D.3, Sec. C.1.4, Sec. C.1.1, Sec. B.3, and Sec. B.1).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may

be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: We leveraged datasets and models available on the huggingface hub and torchvision datasets for computer vision and NLP (apache 2.0, CC BY-NC-SA 4.0, CC BY-SA, CC BY-NC 4.0, CC BY 4.0 license), in molecules we gathered publicly available datasets under the apache 2.0 license and MIT license. All models and datasets are correctly cited in the paper. Code and data are publicly available in different repositories available in the main body of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: For every modality we share hyperparameter choices and specifics of the model in the specific sections as well as the detailed experiments in appendices (see Sec. D.2, Sec. D.3, Sec. C.1.4, and Sec. B.3).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[Yes\]](#)

Justification: We report comprehensive results in appendices for every task with error bars and subset specific evaluations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: We both discuss the method complexity in the main paper and used resources in Appendix F and Sec. 5.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: We attribute all artefacts used and provide direct links to them for reproducibility purposes. All the resources we used were publicly available under permissive licences.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Source code is shared through anonymous git repositories and trained models will be released on the Huggingface hub.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [NA]

Justification: [NA]

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

I Funding

This work was granted access to the HPC resources of IDRIS under the allocation AD011013290R3, and enabled by support provided by Calcul Quebec and the Digital Research Alliance of Canada. This work was funded through scholarships by "École de Technologie Supérieure Montreal", "Université Paris-Saclay" and "McGill University".