RAPID: INCREASING EFFICIENCY OF INTERNAL DISTILLATION VIA RANDOMIZED METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation is a technique used to transfer the expertise of large teacher models into small, efficient student models. Recent advances have integrated the distillation of internal activations, more directly training students to learn similar representations as the teacher. One of the core challenges of internal distillation is the dimensional incompatibility between the student and teacher internal representations. Existing methods add learnable matrices to project the larger teacher activations into the smaller representation space of the student. This is a notable point of inefficiency, as learning additional parameters for the projectors increases training costs, only for them to be discarded immediately after training. We introduce RAPID (RAndomized Projection for Internal Distillation) a novel distillation methodology which eliminates the wasteful projection learning by leveraging randomized methods. RAPID utilizes an efficient transform construction based on the Johnson-Lindenstrauss lemma to compress the teacher representations into the student embedding space with strong theoretical guarantees of structural preservation. RAPID also employs a combination of exponential moving average (EMA) and homoscedastic uncertainty weighting in a novel multi-objective function designed for internal distillation. Our experiments in post-training for reasoning demonstrate that RAPID increases training efficiency by as much as 21.1% while maintaining performance comparable to existing methods.

1 INTRODUCTION

Recent open-weight large language models (LLMs), including Kimi K3 Team et al. (2026), DeepSeek V4 Pro DeepSeek-AI (2026), Qwen 3.8 Max Qwen Team (2026), and GLM 5.3 GLM-5-Team et al. (2026), have approached proprietary frontier performance across scientific reasoning, software engineering, multimodal understanding, long-context processing, and tool use. However, their substantial deployment costs make them impractical for many applications, particularly in healthcare, law, finance, cybersecurity, and autonomous vehicles, where compute, latency, and privacy constraints are stringent. Open weights enable these capabilities to be transferred into smaller, more efficient models.

Knowledge distillation transfers knowledge from a large *teacher* model to a smaller *student*. It has been used to pretrain models in Google's Gemma 2 and 3 families Team et al. (2024; 2025) and Meta's Llama 3.2 family Meta (2024). It is also effective for post-training Guo et al. (2025), improving reasoning, planning, and tool-use capabilities in lightweight agentic systems.

However, text-based distillation is highly sensitive to the quality and diversity of teacher trajectories and their alignment with the student's pretraining distribution Gudibande et al. (2023). *Internal distillation* supplements these trajectories with teacher activations, providing richer supervision and encouraging the student to process inputs similarly to the teacher.

Existing internal-distillation methods learn projections between teacher and student representation spaces before computing losses such as mean-squared error (MSE) or KL divergence Liang et al. (2023); Jiao et al. (2020); Sun et al. (2019). These methods were developed when large models contained roughly 150 million parameters, over four orders of magnitude fewer than today's approximately two-trillion-parameter frontier models. As shown in Table 2, at modern scales, learned projectors may require training billions of additional parameters beyond the student itself. Consequently, the cost of internal distillation remains a major barrier to further exploration and adoption.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

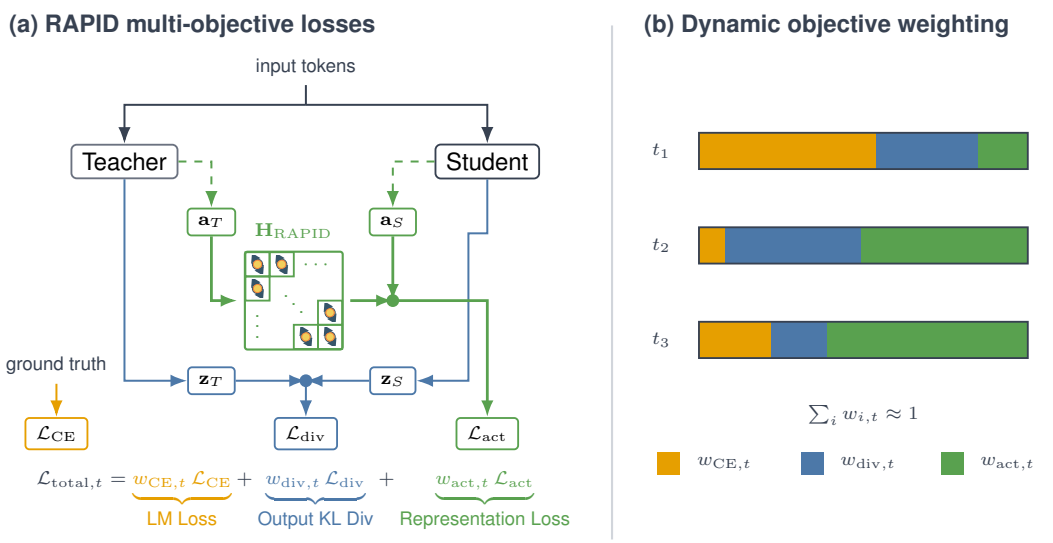


Figure 1: Overview of the RAPID internal distillation methodology. Figure 1(a) demonstrates how teacher activations are passed through a randomized projector to produce a representation in the student's space. The multi-objective function is comprised of the language modeling loss, KL divergence on the output logits (z_T, z_S), and the representation loss on the internal activations (a_T, a_S). Figure 1(b) shows how RAPID's dynamic weighting scheme allows the student to prioritize different forms of knowledge at different times while preventing any singlular objective from dominating or being ignored.

To efficiently distill the capabilities of large modern LLMs into smaller, more efficient models, we turn to randomization. Randomization has an extensive history in accelerating numerical computation and machine learning Mahoney (2011); Martinsson & Tropp (2021), including specifically in dimension projection and information compression Drinea et al. (2004); Johnson & Lindenstrauss (1984); Achlioptas (2003). We also draw from work in dynamic weighting and multi-objective optimization Li et al. (2021); Wang et al. (2023a); Kendall et al. (2018) to formulate a novel objective function designed for the noisy signals of internal distillation.

Our core contributions are:

- In Section 4 we introduce RAPID (**RA**ndomized **P**rojections for **I**nternal **D**istillation), a novel methodology for conducting internal distillation which eliminates the need for training projection matrices by replacing them with randomized matrices and leverages a novel dynamic multi-objective loss function
- In Sections 6 and 7.3 we demonstrate that RAPID achieves task performance comparable to existing methods while improving parameter efficiency by as much as 8.1% in the residual-only distillation scheme, and up to 21.1% with an expanded set of distilled activations
- In Section 7.1 we detail how RAPID enforces a separation of responsibilities between projection and representation learning, and Section 7.2 demonstrates that RAPID produces similar or improved representational alignment with the teacher

2 Preliminaries

2.1 Knowledge Distillation

Knowledge distillation (KD) was first introduced by Buciluǎ et al. (2006) and Hinton et al. (2015) was the landmark work that adapted this approach to neural networks. Recent advancements have built upon these foundational works to distill post-training behaviors such as reasoning and instruction-following using synthetic data traces Guha et al. (2025); Guo et al. (2025); Wang et al.

(2023b) and to distill internal activations in addition to output logits Liang et al. (2023); Jiao et al. (2019); Sun et al. (2019).

Logit and internal-activation distillation require alignment between teacher and student tokenizers. Because LLMs are sensitive to tokenization and cross-family vocabularies are difficult to align, these methods are typically restricted to models within the same family. Internal distillation must also reconcile differences in hidden dimensionality,[1] since losses such as mean-squared error and cosine similarity require equal-dimensional vectors. Teacher and student activations must therefore be projected into a shared representation space.

2.2 Randomized Methods for Efficient Computation

Randomization accelerates numerical computation and machine learning by replacing expensive exact operations with efficient approximations whose errors are probabilistically controlled Mahoney (2011); Martinsson & Tropp (2021). One of the foundational works in randomization for information compression is Johnson & Lindenstrauss (1984) which introduced the Johnson-Lindenstrauss lemma (restated here in Lemma 1). Simply put, the JL lemma states that any set of n points in high-dimensional Euclidean space can be embedded into a smaller dimensional space while approximately preserving all pairwise distances.

2.3 Mechanistic Interpretability

Mechanistic interpretability seeks to develop human-interpretable accounts of how LLMs store, recognize, and process information. Although individual neurons and model components are difficult to interpret in isolation Elhage et al. (2022); Ameisen* et al.; Templeton et al. (2024), recent work suggests that the resulting activation spaces are not unstructured. Instead, LLMs represent concepts and semantic relationships through organized geometric patterns, including identifiable directions and low-dimensional subspaces Elhage et al. (2022); Bricken et al. (2023); Park et al. (2024); Gurnee & Tegmark (2024); Park et al. (2025).

3 Related Work

3.1 Knowledge Distillation

Response-trace distillation has proven effective for transferring post-training behaviors Wang et al. (2023b); Guo et al. (2025) and supports both proprietary teachers and cross-family transfer. However, its effectiveness is sensitive to the scope, quality, and diversity of the trace distribution Gudibande et al. (2023). Logit distillation instead trains students on teacher soft targets, following Hinton et al. (2015). Google and Meta have applied this approach during pretraining for models up to 27B parameters Meta (2024); Team et al. (2024; 2025), while Alibaba used it to consolidate multiple teacher post-training stages into a single student-training stage Team (2025).

3.1.1 Internal Distillation

Internal distillation goes beyond traditional output logit distillation to additionally distill internal activations into the student Liang et al. (2023); Jiao et al. (2019); Sun et al. (2019). While logit distillation attempts to improve the generalization of the knowledge being distilled by training students to replicate the teacher's output distribution, internal distillation also trains students to replicate the internal processing steps which produce the output logits. The intuition behind this approach is that the teacher is able to provide step-by-step guidance to how to student arrives at the final answer, a more granular and information-rich signal than output logits alone.

TinyBERT Jiao et al. (2019) distills the embeddings, all residual stream activations, and all attention distribution matrices. Patient KD Sun et al. (2019) reduces the computational cost of internal distillation by only distilling the internal activations of the `[CLS]` token[2] from a subset of layers. TED Liang et al. (2023) performs task-specific distillation by training filters on top of residual stream activations to reduce the redundant information transferred to the student.

[1] We use *representation space*, *embedding space*, *internal dimension*, and *hidden dimension* interchangeably.

[2] the `[CLS]` token in BERT models which contains the full representation of the entire sequence

3.2 Randomized Projections for Information Compression

The original transform constructed by Johnson and Lindenstrauss applied a suitably scaled random orthogonal projection onto a k-dimensional subspace. Later work such as Achlioptas (2003) improved efficiency by replacing conventional Gaussian projection matrix with one of two discrete random matrices. In the dense "binary coin" construction, each entry is sampled independently as

$$R_{ij} = \begin{cases} +1, & \text{with probability } \frac{1}{2}, \\ -1, & \text{with probability } \frac{1}{2}. \end{cases} \tag{1}$$

In the sparse ternary construction, each entry is sampled independently as

$$R_{ij} = \sqrt{3} \begin{cases} +1, & \text{with probability } \frac{1}{6}, \\ 0, & \text{with probability } \frac{2}{3}, \\ -1, & \text{with probability } \frac{1}{6}. \end{cases} \tag{2}$$

Both distributions have zero mean and unit variance, allowing the full data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ to be projected as

$$\mathbf{Z} = \frac{1}{\sqrt{k}} \mathbf{X}\mathbf{R}, \qquad \mathbf{Z} \in \mathbb{R}^{n \times k}, \tag{3}$$

where $\mathbf{R} \in \mathbb{R}^{d \times k}$ is the random projection matrix. The dense construction replaces general multiplications with additions and subtractions, while the sparse construction additionally eliminates approximately two-thirds of the operations because two-thirds of its entries are zero.

3.3 Dynamic Objective Weighting

Wang et al. (2023a) demonstrated that knowledge distillation can benefit from different types of knowledge at different times during training. Kendall et al. (2018) introduced uncertainty-based weighting for multi-objective optimization in deep learning, and Li et al. (2021) employed it in internal distillation to allow for downweighting of representational alignment when student can already mimic teacher outputs. Lakkapragada et al. (2022) leveraged exponential moving average (EMA) to dynamically normalize loss magnitudes for stability despite noisy updates.

4 Methodology

4.1 Intuition

There are three core features of RAPID that distinguish itself from existing internal distillation methodologies. The first is that, by using a fixed projector, we strictly enforce that the full responsibility of representation learning falls solely on the student[3]. The second is that the usage of a JL transform provides guarantees about the preservation of structure in the projection of teacher representations. Recent mechanistic interpretability research provides evidence that LLMs represent semantic concepts and their relationships through structured geometric organization in their representation spaces Elhage et al. (2022); Bricken et al. (2023); Park et al. (2024); Gurnee & Tegmark (2024); Park et al. (2025). By preserving teacher representational geometry, RAPID targets the distillation of semantic relationships– a key component of advanced knowledge and intelligence. The final distinct feature is that RAPID allows the student to dynamically adapt its learning to target its weaknesses while appropriately weighting the noisy signals from distilling internal representations.

4.2 Randomized Projector

We use the dense binary coin JL transform described in Algorithm 1 to create the randomized projectors in RAPID. The dense formulation was selected on the basis of modern GPUs being highly optimized for dense matrix operations. Since the JL lemma applies only to data compression, we set d to the dimension of the teacher representation space and k to the dimension of the student representation space to construct a *downward* projection.

[3]Further details regarding the separation of projection and representation learning are in Section 7.1

We construct a linearly-spaced layer mapping ($\mathcal{H}$) between the layers of the teacher and student such that every layer of the student is mapped to a layer in the teacher. For each activation being distilled, we independently initialize a separate projection matrix. For example, in our Gemma 3 experiments in Section 5 where we distill only the residual activations, we create (18 layers) $\times$ (1 activation per layer) = 18 projectors. Our Section 7.3 experiments with the expanded activation set construct (18 layers) $\times$ (3 activations per layer) + (1 projector for embeddings) = 55 projectors.

4.3 Multi-Objective Loss

We draw from recent work in knowledge distillation Li et al. (2021); Wang et al. (2023a) and utilize a dynamic objective function which combines learned uncertainty weighting Kendall et al. (2018) with exponential moving average (EMA) progress balancing. Internal distillation is a multi-objective optimization problem, and using a dynamic objective function allows the model to prioritize different components of the objective at different points throughout training.

At training step t, given an input sequence x, raw losses are computed between the frozen teacher $\mathcal{M}_T$ and the student $\mathcal{M}_S$ across three levels: language modeling task loss on next-token labels, output distribution divergence, and internal activation matching error.

Task Loss The student is directly trained on target sequence labels y using cross-entropy loss– the standard language modeling loss– over student output logits $\mathbf{z}_S$:

$$\mathcal{L}_{\text{CE}} = \text{CE}(\mathbf{z}_S, y). \tag{4}$$

Output Distribution Distillation Output alignment between teacher logits $\mathbf{z}_T$ and student logits $\mathbf{z}_S$ is enforced using forward Kullback-Leibler (KL) divergence computed at temperature τ:

$$\mathcal{L}_{\text{div}} = \tau^2 \ \text{KL}\left(\exp\left[\log\text{softmax}\left(\frac{\mathbf{z}_T}{\tau}\right)\right] \middle\| \exp\left[\log\text{softmax}\left(\frac{\mathbf{z}_S}{\tau}\right)\right]\right). \tag{5}$$

Intermediate Representation Distillation To align internal representations, we intercept internal activations for the teacher $\mathbf{a}_T^{(l)}$ and student $\mathbf{a}_S^{(l)}$ across matched hook layers $l \in \mathcal{H}$. Representation spaces are aligned using a linear projection head P_ϕ. Depending on the projector, representation losses for internal activations $\mathcal{L}_{act}$ are computed as the Mean Squared Error (MSE) summed over matching layer pairs $\mathcal{H}$:

$$\mathcal{L}_{act} = \sum_{l \in \mathcal{H}_a} \begin{cases} \text{MSE}\left(\mathbf{a}_T^{(l)}, P_\phi\left(\mathbf{a}_S^{(l)}\right)\right), & \text{if Projector} = \text{Learned} \\ \text{MSE}\left(P_\phi\left(\mathbf{a}_T^{(l)}\right), \mathbf{a}_S^{(l)}\right), & \text{if Projector} = \text{Randomized} \end{cases}. \tag{6}$$

Section 6 shows results when distilling only the residual activations, though our implementation can handle an expanded set of activations (see Section 7.3). Inactive internal activation objectives for a given experiment are assigned zero ($\mathcal{L}_{act} = 0$) and masked out.

4.4 Hybrid Dynamic Loss Weighting

To dynamically balance optimization among each term in the objective, we employ a dynamic weighting mechanism which unifies learned homoscedastic uncertainty weighting with an exponential moving average (EMA) progress ratio.

Let $\mathcal{K}$ represent the set of objective components. At step t, let $\mathcal{L}_{i,t}$ denote the unweighted scalar loss calculated for component $i \in \mathcal{K}$.

Tracking Objective Baselines To maintain historical loss scale baselines without retaining full step histories, an exponential moving average (EMA) buffer $\bar{\mathcal{L}}_{i,t}$ is tracked for each objective:

$$\bar{\mathcal{L}}_{i,t} = \eta \bar{\mathcal{L}}_{i,t-1} + (1-\eta)\text{sg}[\mathcal{L}_{i,t}], \tag{7}$$

where $\eta \in (0,1)$ is the EMA decay rate and $\text{sg}[\cdot]$ denotes the stop-gradient operator. We set $\eta = 0.99$.

Dual Signal Generation The unnormalized weighting signal for objective i combines two distinct mechanisms:

1. **Learned Uncertainty Weighting:** Parameterized by learnable log-variance terms $s_i \in \mathbb{R}$, capturing intrinsic task noise characteristics:
$$w_i^{\text{unc}} = \exp(-s_i). \tag{8}$$
2. **EMA Progress Weighting:** Measures transient loss spikes relative to historical baseline performance, scaled by hyperparameter $\alpha \geq 0$:
$$w_{i,t}^{\text{prog}} = \left(\frac{\text{sg}[\mathcal{L}_{i,t}]}{\bar{\mathcal{L}}_{i,t} + \epsilon} \right)^{\alpha}, \tag{9}$$
where $\epsilon > 0$ provides numerical stability. We set $\alpha = 0.5$ and $\epsilon = 1 \times 10^{-8}$.

Combination, Masking, and Bounding The combined raw weight for objective i is calculated as the product of both signals, masked by indicator $m_i = \mathbb{I}(i \in \mathcal{A})$:
$$\tilde{w}_{i,t} = m_i \cdot \left(w_i^{\text{unc}} \cdot w_{i,t}^{\text{prog}} \right). \tag{10}$$
These raw weights are normalized across active objectives to project onto a probability simplex:
$$\hat{w}_{i,t} = \frac{\tilde{w}_{i,t}}{\sum_{j \in \mathcal{A}} \tilde{w}_{j,t} + \epsilon}. \tag{11}$$
To prevent gradient starvation or single-objective dominance, normalized weights are clamped within lower and upper bounds $[\omega_{\min}, \omega_{\max}]$ for active tasks and re-normalized:
$$w'_{i,t} = \begin{cases} \text{clip}\left(\hat{w}_{i,t}, \omega_{\min}, \omega_{\max}\right), & \text{if } i \in \mathcal{A} \\ 0, & \text{otherwise} \end{cases}, \qquad w_{i,t} = \frac{w'_{i,t}}{\sum_{j \in \mathcal{A}} w'_{j,t} + \epsilon}. \tag{12}$$
We use $\omega_{min} = 0.05$, $\omega_{max} = 0.80$.

Total Objective Function The final scalar loss minimized at step t combines dynamically scaled raw objectives with a linear log-variance regularization term over active tasks:
$$\mathcal{L}_{\text{total}} = \sum_{i \in \mathcal{A}} w_{i,t} \mathcal{L}_{i,t} + \sum_{i \in \mathcal{A}} s_i. \tag{13}$$

Optimization Mechanics All operations computing intermediate weights $w_{i,t}$ from $\mathcal{L}_{i,t}$ and s_i are evaluated without gradient tracking. As a result, $\partial \mathcal{L}_{\text{total}} / \partial s_i = 1$ for all active heads, preventing representation backpropagation from corrupting s_i and avoiding scale-invariance degeneracies. Further details regarding this are provided in Appendix D.

5 Experiments

5.1 Comparisons

We compare RAPID with internal distillation using learned linear projectors, the most common approach in prior work Xu et al. (2024); Zhu et al. (2024). Following existing methods Jiao et al. (2019); Sanh et al. (2020); Liang et al. (2023); Sun et al. (2019), each projector maps a student activation upward into the teacher's representation space. We use separate projectors for each distilled activation and equally weight the ground-truth, output-distribution, and internal-activation losses.

5.2 Models

We demonstrate RAPID using distillation pairs from the Gemma 3 Team et al. (2025) and Qwen 2.5 Qwen et al. (2025) model families. We exclusively perform in-family distillation, distilling Gemma 3 4B IT into Gemma 3 270M PT[4], and Qwen 2.5 3B Instruct into Qwen 2.5 0.5B. These

[4]Gemma 3 270M actually has 436M parameters, but the embedding and unembedding weights are tied

Dataset	Distillation Pair	Variant	ARC-C	BBH	GPQA	MMLU	Average
OpenThoughts	Gemma 3	Learned	**0.2770**	0.2804	0.2438	**0.2475**	**0.2622**
		RAPID	0.2747	**0.2848**	**0.2506**	0.2381	0.2621
	Qwen 2.5	Learned	**0.3376**	**0.3018**	0.2508	**0.2560**	**0.2866**
		RAPID	0.3348	0.3005	**0.2539**	0.2520	0.2853
SmolTalk	Gemma 3	Learned	0.2742	**0.2898**	**0.2550**	0.2355	0.2636
		RAPID	**0.2796**	0.2895	0.2517	**0.2382**	**0.2647**
	Qwen 2.5	Learned	0.3490	0.3129	**0.2581**	**0.3241**	0.3110
		RAPID	**0.3549**	**0.3234**	0.2503	0.3156	**0.3111**

Table 1: Performance comparison between learned projectors and RAPID across 2 datasets and 2 distillation pairs in the residual distillation scheme. Reported scores are averages across 3 seeds.

approximate parameter compression ratios of 9.2:1 for Gemma 3 and 6:1 for Qwen 2.5 follow the existing distillation literature range of 1.6:1 to 10:1 Jiao et al. (2019); Sanh et al. (2020); Meta (2024); Gu et al. (2026) with absolute parameter counts scaled to modern LLM sizes.

We selected the student initialization point to be the base pretrained models for the cleanest starting point for our post-training. Each model is trained with three separate seeds $\{3, 33, 333\}$ and the scores reported are averaged across the seeds.

5.3 Training

We train on 40M tokens of reasoning data with a sequence length of 1024[5], corresponding to about 40k training samples per experiment. To show data generalization, we trained with 2 separate datasets: a version of SmolTalk Allal et al. (2025) curated for reasoning and maximum 1024 sequence length per sample, and unmodified OpenThoughts-114k Guha et al. (2025). Given the chat format of these data, we use the tokenizer for the instruction-tuned variants of the student models.

5.4 Evaluation Tasks

We evaluate reasoning task performance on four benchmarks: MMLU Hendrycks et al. (2021), ARC-Challenge Clark et al. (2018), and the Open Leaderboard V2 Fourrier et al. (2024) versions of BBH Suzgun et al. (2022) and GPQA Rein et al. (2023). These are common benchmarks used in the literature for LLMs of similar size to our student models Qwen et al. (2025); Team (2025); Team et al. (2024; 2025); Biderman et al. (2023); Meta (2024) and are designed to evaluate knowledge and reasoning capabilities. Following the conventions in the literature, we evaluate MMLU as 5-shot, ARC-Challenge as 25-shot, BBH as 3-shot, and GPQA as 0-shot.

6 Results

6.1 Task Performance

As shown in Table 1, two methods achieve nearly identical performance despite RAPID being over 6% more efficient in the Gemma 3 experiments, and over 8% more efficient in the Qwen 2.5 experiments. No difference exceeds 0.15 percentage points in the overall average, with largest individual delta being +1.05 percentage points for RAPID over Learned for SmolTalk on BBH.

6.2 Efficiency

RAPID improves training efficiency by eliminating learned projectors. Table 2 reports their additional parameter costs across internal distillation configurations, where **All** includes residual, MLP-output, attention-output, and embedding activations. We calculate RAPID's efficiency gain as

$$\frac{P_{\text{added}}}{P_{\text{model}} + P_{\text{added}}} \times 100\%. \tag{14}$$

[5] 1024 was the largest sequence length our hardware could accommodate

Teacher	Student	Added Params		Added Params (%)		RAPID Efficiency	
		Residual	All	Residual	All	Residual	All
Gemma 3 4B	Gemma 3 270M	29.5M	90.1M	6.8%	20.7%	+6.3%	+17.1%
Qwen 2.5 3B	Qwen 2.5 0.5B	44M	134M	8.8%	26.8%	+8.1%	+21.1%
Llama 3.1 8B	Llama 3.2 1B	134.2M	411M	13.4%	41.1%	+11.8%	+29.1%
Olmo 3 32B	Olmo 3 7B	671.1M	2.03B	9.6%	29.0%	+8.8%	+22.5%

Table 2: Projector parameter overhead for each teacher–student model pairing and the parameter efficiency gain from eliminating these additional parameters with RAPID. We use 436M as the parameter count for Gemma 3 270M. We provide additional model pairings which represent reasonable candidates for distillation.

Learned projectors increase trainable parameters by approximately 10% for residual-only distillation and 30% for **All** activations—up to 2B additional parameters for a 7B student. RAPID reduces trainable parameters by 8.1% in our Qwen experiments (Section 6.1) and 21.1% when distilling **All** activations (Section 7.3).

7 Analysis

7.1 Projector–Student Separation of Responsibilities

Internal-activation distillation ideally assigns distinct roles to the student and projector: the student learns the transferred teacher representation, while the projector performs a transformation to reconcile dimensionality. RAPID enforces this separation by applying a fixed downward projection $\mathbf{H}_{\text{fix}} \in \mathbb{R}^{d_s \times d_t}$ to the teacher:

$$\mathcal{L}_{\text{down}} = \frac{1}{d_s} \left\| \mathbf{a}_s - \mathbf{H}_{\text{fix}} \mathbf{a}_t \right\|_2^2 . \tag{15}$$

Because the projected teacher activation is stationary, the projector only defines the target; all adaptation required to reduce the loss must occur in the student.

Existing methods commonly utilize a learned upward projection $\mathbf{H}_{\text{learn}} \in \mathbb{R}^{d_t \times d_s}$ Liang et al. (2023); Jiao et al. (2019); Sun et al. (2019):

$$\mathcal{L}_{\text{up}} = \frac{1}{d_t} \left\| \mathbf{H}_{\text{learn}} \mathbf{a}_s - \mathbf{a}_t \right\|_2^2 . \tag{16}$$

Learned projectors are popular for their flexibility: beyond matching dimensions, they can learn particular transforms which facilitate student learning Miles & Mikolajczyk (2024); Chen et al. (2022). However, jointly training the projector and student allows both to reduce the loss. In upward projections, the projector determines both the activation residual and how it reaches the student, potentially distributing alignment across the student and a projector discarded after training. We call this effect *dilution*.

RAPID trades this flexibility for a strict separation of responsibilities by removing the competing adaptation path and placing representation learning entirely on the student, eliminating the possibility of dilution. Appendix E provides the corresponding gradient analysis.

7.2 Representational Alignment

We evaluate representational alignment between students and teachers using centered kernel alignment (CKA) and representational similarity analysis (RSA), the latter with an inner metric of cosine similarity and an outer metric of Spearman correlation. CKA measures the overall alignment between representations and RSA measures the similarity of pairwise relationships. We analyzed the students trained on SmolTalk in the residual-only distillation scheme from Section 6. Final token activations were collected from students and teachers across 3 seeds $\{123, 45, 67\}$ for 500 samples of the train set of GSM8K Cobbe et al. (2021) with a 1024 sequence length. Tables and figures for these results can be found in Appendix F. **Global** refers to an analysis of the full set of teacher layers

Distillation Pair	Variant	ARC-C	BBH	GPQA	MMLU	Average
Gemma 3	Learned	0.2688	**0.2881**	**0.2550**	0.2363	0.2621
	RAPID	**0.2779**	**0.2881**	**0.2550**	**0.2386**	**0.2649**
Qwen 2.5	Learned	**0.3461**	**0.3115**	0.2562	**0.3092**	**0.3058**
	RAPID	0.3441	0.3027	**0.2578**	0.2521	0.2892

Table 3: Performance comparison between learned projection and RAPID on SmolTalk, distilling **All** activations.

vs the full set of student layers, and **Mapped** only uses the set of teacher layers in our distillation mapping $\mathcal{H}$. P_{25} is the 25th percentile value.

RAPID produces substantially stronger RSA values for Gemma 3 across all global and mapped metrics, while the learned projector performs slightly better on most CKA metrics. For Qwen 2.5, the learned projector consistently—but only marginally—outperforms RAPID across both CKA and RSA. Overall, the small differences suggest that RAPID achieves representational alignment comparable to learned projection without requiring training additional projection parameters. Interestingly, the Qwen 2.5 values are significantly higher than the Gemma 3 values, even though the entire Gemma 3 family was pretrained with distillation Team et al. (2025). We leave an explanation of this to future work.

7.3 Distilling Additional Activations

While some existing internal distillation methods distill only the residual stream activations Sun et al. (2019); Liang et al. (2023), other methods such as TinyBERT Jiao et al. (2019) distill an expanded set of activations. To demonstrate that RAPID generalizes to an expanded activation set, we include an analysis which distills attention outputs, MLP outputs, and embeddings in addition to the residual activations on our Smoltalk dataset. We refer to this expanded set of activations as **All** activations.

Table 3 contains the results for distilling **All** activations for the SmolTalk dataset. The Gemma 3 distillation pair achieves very similar results as in the residual-only distillation scheme in Table 1, however there is a small performance degradation for Qwen 2.5, with RAPID performing 2.2 percentage points worse when distilling **All** activations. These Qwen 2.5 results also exhibit the largest performance gap of all of our experiments, at +1.67 percentage points for the learned projection. However, this small performance lift for learned projectors over RAPID comes at the cost of over 20% worse efficiency.

8 Limitations

Limited compute[6] constrained model size, sequence length, and teacher finetuning, preventing full distillation and evaluation on R1-style long-context reasoning. Moreover, we evaluate RAPID only during reasoning-focused supervised finetuning and do not assess its effectiveness in pretraining, mid-training, general instruction tuning, tool-use training, or as an initialization for subsequent reinforcement learning and other post-training stages. Future work should evaluate RAPID across these training phases and at larger scales, as well as optimize the dynamic-weight bounds $\omega_{\min}, \omega_{\max}$ in Equation 12.

9 Conclusion

We introduced RAPID, a novel methodology for internal distillation which achieves comparable performance to existing methods with learned projectors while being 6.3%–21.2% more efficient in our experiments. RAPID generalizes across models families, training datasets, and activation sets. RAPID places the responsibility of representation learning on the student, resulting in comparable or better representational similarity with the teacher.

[6]We ran all experiments on a single server with two RTX A6000 48GB GPUs.

AI USE STATEMENT

In this work, we used generative AI tools for: formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assistance in writing proofs in Appendices D and E; designing and providing feedback on the objective formulation; and implementing methods in our experimental harness. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, supporting qualitative and thematic data analysis, or interpreting results.and generating synthetic datasets, assisting with translation, and cleaning and reformatting data are not applicable to this work.

Additionally, we used generative AI tools for suggesting experimental parameters, creating and editing code, drafting Sections 4.3 and 4.4, summarizing existing literature, identifying research gaps, brainstorming, sourcing/searching for information, editing Sections 1, 2, and 3 for brevity, identifying relevant literature, and formatting references.

We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey. LLM-generated code was verified and tested for correctness by the first author. LLM-generated proofs were reviewed and modified for correctness and clarity. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We have taken a number of steps to ensure high levels of reproducibility in our work. All experiments were run with random seeds (whose values we provided) and the PyTorch reproducible backend for NVIDIA GPUs. Algorithm 1 contains all details needed to reproduce the randomized projectors, and Sections 4.3 and 4.4 show detailed formulations of our multiple objectives, dynamic weight updates, and parameters used.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, June 2003. ISSN 0022-0000. doi: 10.1016/S0022-0000(03)00025-4. URL https://www.sciencedirect.com/science/article/pii/S0022000003000254.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model, 2025. URL https://arxiv.org/abs/2502.02737.

Authors Emmanuel Ameisen*, Jack Lindsey*, Adam Pearce*, Wes Gurnee*, Nicholas L. Turner*, Brian Chen*, Craig Citro*, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson*‡ Affiliations Anthropic Published March 27. Circuit Tracing: Revealing Computational Graphs in Language Models. URL https://transformer-circuits.pub/2025/attribution-graphs/methods.html.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and others. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pp. 535–541, New York, NY, USA, August 2006. Association for Computing Machinery. ISBN 978-1-59593-339-3. doi: 10.1145/1150402.1150464. URL https://dl.acm.org/doi/10.1145/1150402.1150464.

Yudong Chen, Sen Wang, Jiajun Liu, Xuwei Xu, Frank de Hoog, and Zi Huang. Improved Feature Distillation via Projector Ensemble. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 12084–12095. Curran Associates, Inc., 2022. doi: 10.52202/068431-0878. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/4ec0b6648bdf487a2f1c815924339022-Paper-Conference.pdf.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence, 2026.

Eleni Drinea, Petros Drineas, and Patrick S. Huggins. A Randomized Singular Value Decomposition Algorithm for Image Processing Applications. 2004. URL https://api.semanticscholar.org/CorpusID:5727723.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and others. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open LLM Leaderboard v2, 2024. URL https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. GLM-5: from Vibe Coding to Agentic Engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-Policy Distillation of Large Language Models, 2026. URL https://arxiv.org/abs/2306.08543.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The False Promise of Imitating Proprietary LLMs, 2023. URL https://arxiv.org/abs/2305.15717.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. OpenThoughts: Data Recipes for Reasoning Models, 2025. URL https://arxiv.org/abs/2506.04178.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi

Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Wes Gurnee and Max Tegmark. Language Models Represent Space and Time, 2024. URL https://arxiv.org/abs/2310.02207.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, March 2015. URL http://arxiv.org/abs/1503.02531. arXiv:1503.02531 [stat].

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding, 2020. URL https://arxiv.org/abs/1909.10351. _eprint: 1909.10351.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. 26:189–206, 1984. doi: 10.1090/conm/026/737400. URL http://www.ams.org/conm/026/. Book Title: Contemporary Mathematics ISBN: 9780821850305 9780821876114 Place: Providence, Rhode Island.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics, 2018. URL https://arxiv.org/abs/1705.07115.

Anish Lakkapragada, Essam Sleiman, Saimourya Surabhi, and Dennis P. Wall. Mitigating Negative Transfer in Multi-Task Learning with Exponential Moving Average Loss Weighting Strategies, 2022. URL https://arxiv.org/abs/2211.12999.

Lei Li, Yankai Lin, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. Dynamic Knowledge Distillation for Pre-trained Language Models, 2021. URL https://arxiv.org/abs/2109.11295.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: task-aware layer-wise distillation for language model compression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, pp. 20852–20867, Honolulu, Hawaii, USA, July 2023. JMLR.org.

Michael W. Mahoney. Randomized algorithms for matrices and data, 2011. URL https://arxiv.org/abs/1104.5557.

Per-Gunnar Martinsson and Joel Tropp. Randomized Numerical Linear Algebra: Foundations & Algorithms, 2021. URL https://arxiv.org/abs/2002.01387.

Meta. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models, September 2024. URL https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.

Roy Miles and Krystian Mikolajczyk. Understanding the Role of the Projector in Knowledge Distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):4233–4241, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i5.28219. URL https://ojs.aaai.org/index.php/AAAI/article/view/28219.

Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models, July 2024. URL http://arxiv.org/abs/2311.03658. arXiv:2311.03658 [cs.CL].

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. The Geometry of Categorical and Hierarchical Concepts in Large Language Models, 2025. URL https://arxiv.org/abs/2406.01506.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Qwen Team. Qwen3.8-Max: A New Bar for Coding and Cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A Graduate-Level Google-Proof Q&A Benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, March 2020. URL http://arxiv.org/abs/1910.01108. arXiv:1910.01108 [cs].

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient Knowledge Distillation for BERT Model Compression, 2019. URL https://arxiv.org/abs/1908.09355. _eprint: 1908.09355.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them, 2022. URL https://arxiv.org/abs/2210.09261.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen

Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving Open Language Models at a Practical Size, 2024. URL https://arxiv.org/abs/2408.00118.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, C. J. Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju-yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier

Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 Technical Report, March 2025. URL http://arxiv.org/abs/2503.19786. arXiv:2503.19786 [cs].

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang, Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu,

Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi K3: Open Frontier Intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Qwen Team. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388. _eprint: 2505.09388.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Chenglong Wang, Yi Lu, Yongyu Mu, Yimin Hu, Tong Xiao, and Jingbo Zhu. Improved Knowledge Distillation for Pre-trained Language Models via Knowledge Selection, 2023a. URL https://arxiv.org/abs/2302.00444.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions, May 2023b. URL http://arxiv.org/abs/2212.10560. arXiv:2212.10560 [cs].

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A Survey on Knowledge Distillation of Large Language Models, October 2024. URL http://arxiv.org/abs/2402.13116. arXiv:2402.13116 [cs].

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A Survey on Model Compression for Large Language Models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, November 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00704. URL https://doi.org/10.1162/tacl_a_00704. _eprint: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00704/2482209/tacl_a_00704.pdf.

A JOHNSON-LINDENSTRAUSS LEMMA

We provide a restatement of the JL lemma here for convenience.

Lemma 1 (Johnson–Lindenstrauss Lemma)**.** *Let $0 < \varepsilon < 1$ and let $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$. For any target dimension*

$$k \geq C\frac{\log n}{\varepsilon^2},$$

where $C > 0$ is a universal constant, there exists a linear map $f : \mathbb{R}^d \to \mathbb{R}^k$ such that, for every $x_i, x_j \in X$,

$$(1-\varepsilon)\|x_i - x_j\|_2^2 \leq \|f(x_i) - f(x_j)\|_2^2 \leq (1+\varepsilon)\|x_i - x_j\|_2^2.$$

B ACHLIOPTAS PROJECTOR INITIALIZATION ALGORITHMS

Algorithm 1 Dense Binary Coin Random Projection

Require: $\mathbf{X} \in \mathbb{R}^{n \times d}$, target dimension k
Ensure: $\mathbf{Z} \in \mathbb{R}^{n \times k}$
Initialize $\mathbf{R} \in \mathbb{R}^{d \times k}$
for $i = 1, \dots, d$ **do**
  for $j = 1, \dots, k$ **do**
    Sample $u \sim \mathrm{Uniform}(0, 1)$
    if $u < 1/2$ **then**
      $R_{ij} \leftarrow +1$
    else
      $R_{ij} \leftarrow -1$
    end if
  end for
end for
$\mathbf{Z} \leftarrow \frac{1}{\sqrt{k}}\mathbf{XR}$
return $\mathbf{Z}$

Algorithm 2 Sparse Ternary Random Projection

Require: $\mathbf{X} \in \mathbb{R}^{n \times d}$, target dimension k
Ensure: $\mathbf{Z} \in \mathbb{R}^{n \times k}$
Initialize $\mathbf{Q} \in \{-1, 0, +1\}^{d \times k}$
for $i = 1, \dots, d$ **do**
  for $j = 1, \dots, k$ **do**
    Sample $u \sim \mathrm{Uniform}(0, 1)$
    if $u < 1/6$ **then**
      $Q_{ij} \leftarrow +1$
    else if $u < 5/6$ **then**
      $Q_{ij} \leftarrow 0$
    else
      $Q_{ij} \leftarrow -1$
    end if
  end for
end for
$\mathbf{Z} \leftarrow \sqrt{\frac{3}{k}}\mathbf{XQ}$
return $\mathbf{Z}$

C CONTEXTUALIZED JL LEMMA GUARANTEES

We provide a concrete example of the theoretical guarantees of the JL lemma using Gemma 3 270M as a student model. The representation space is $k = 640$ dimensions, and we train for

$$n = \frac{40\,\text{M tokens}}{1024\,\text{tokens per sample}} \approx 40{,}000\ \text{samples}. \tag{17}$$

For $n = 40{,}000$, $k = 640$, and a desired confidence of 99%, set the failure probability to $\delta = 0.01$. Since Achlioptas Achlioptas (2003) gives confidence $1 - n^{-\beta}$, solve

$$n^{-\beta} = \delta$$

to obtain

$$\beta = \frac{\ln(1/\delta)}{\ln n} = \frac{\ln(100)}{\ln(40{,}000)} \approx 0.435.$$

Substituting this value into the Achlioptas bound gives

$$640 \geq \frac{(4 + 2(0.435)) \ln(40{,}000)}{\varepsilon^2/2 - \varepsilon^3/3}.$$

Equivalently,

$$\frac{\varepsilon^2}{2} - \frac{\varepsilon^3}{3} \geq \frac{(4 + 2(0.435)) \ln(40{,}000)}{640} \approx 0.08062.$$

Solving numerically gives

$$\varepsilon \approx 0.489.$$

Therefore, with probability at least 99%, all pairwise squared distances satisfy

$$0.511\|x_i - x_j\|_2^2 \leq \|f(x_i) - f(x_j)\|_2^2 \leq 1.489\|x_i - x_j\|_2^2.$$

In terms of ordinary Euclidean distances,

$$0.715\|x_i - x_j\|_2 \leq \|f(x_i) - f(x_j)\|_2 \leq 1.220\|x_i - x_j\|_2.$$

Thus, at 99% confidence, projecting 40,000 points into 640 dimensions guarantees that every distance is between approximately 71.5% and 122.0% of its original value for every one of the $\binom{40{,}000}{2} = 799{,}980{,}000$ sequence pairs.

D Stop-Gradient Optimization Mechanics in Dynamic Objective

In Section 4.4, we note that all operations computing the dynamic objective weights $w_{i,t}$ from raw losses $\mathcal{L}_{i,t}$ and log-variance parameters s_i are evaluated using a stop-gradient operator ($\text{sg}[\cdot]$). This section provides a formal analysis of why gradient detachment is mathematically necessary and proves the optimization failure modes that arise when $w_{i,t}$ remains fully differentiable with respect to $\mathbf{s} = [s_1, \ldots, s_K]^\top$.

D.1 Fully Differentiable Formulation without Stop-Gradient

Consider a simplified variant of the normalized uncertainty weighting objective where intermediate weights $w_i(\mathbf{s})$ maintain active autograd graph connections:

$$\mathcal{J}(\mathbf{s}) = \sum_{j=1}^{K} w_j(\mathbf{s})\mathcal{L}_j + \sum_{j=1}^{K} s_j, \qquad \text{where } w_i(\mathbf{s}) = \frac{\exp(-s_i)}{\sum_{k=1}^{K} \exp(-s_k)}. \tag{18}$$

Evaluating the partial derivative of $\mathcal{J}(\mathbf{s})$ with respect to an individual log-variance s_i yields:

$$\frac{\partial \mathcal{J}}{\partial s_i} = \sum_{j=1}^{K} \frac{\partial w_j}{\partial s_i}\mathcal{L}_j + 1. \tag{19}$$

Using the standard Jacobian properties of softmax-style normalizations:

$$\frac{\partial w_i}{\partial s_i} = -w_i(1 - w_i), \qquad \frac{\partial w_j}{\partial s_i} = w_i w_j \quad (j \neq i). \tag{20}$$

Substituting these derivatives back into the total gradient gives:

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial s_i} &= -w_i(1 - w_i)\mathcal{L}_i + \sum_{j \neq i} w_i w_j \mathcal{L}_j + 1 \\ &= -w_i\mathcal{L}_i + w_i \sum_{j=1}^{K} w_j \mathcal{L}_j + 1 \\ &= 1 + w_i \left(\overline{\mathcal{L}}_w - \mathcal{L}_i\right), \end{aligned} \tag{21}$$

where $\overline{\mathcal{L}}_w = \sum_{j=1}^{K} w_j \mathcal{L}_j$ represents the current weighted-average loss across all active objectives.

D.2 Failure Modes of the Un-Detached System

Equation 21 exposes two severe optimization vulnerabilities when gradient paths through $w_i(\mathbf{s})$ are preserved:

1. Loss-Evasion Shortcut. If objective i experiences a temporary loss spike such that $\mathcal{L}_i > \overline{\mathcal{L}}_w$, the term $1 + w_i(\overline{\mathcal{L}}_w - \mathcal{L}_i)$ can become negative. Under standard gradient descent ($\mathbf{s} \leftarrow \mathbf{s} - \gamma \nabla_{\mathbf{s}} \mathcal{J}$), a negative gradient forces s_i to grow ($s_i \to +\infty$). The optimizer discovers that it can minimize the global objective $\mathcal{J}$ faster by inflating s_i (suppressing w_i) rather than updating the network backbone parameters to reduce the underlying task error $\mathcal{L}_i$.

2. Scale-Invariance Collapse to Negative Infinity. More critically, normalized uncertainty weights are invariant under uniform translations of the parameter vector $\mathbf{s}$. For any scalar constant $c \in \mathbb{R}$:

$$w_i(\mathbf{s} + c\mathbf{1}) = \frac{\exp[-(s_i + c)]}{\sum_{j=1}^{K} \exp[-(s_j + c)]} = \frac{\exp(-c)\exp(-s_i)}{\exp(-c)\sum_{j=1}^{K} \exp(-s_j)} = w_i(\mathbf{s}). \tag{22}$$

However, the linear regularization penalty $\sum s_j$ is strictly shift-variant:

$$\mathcal{J}(\mathbf{s} + c\mathbf{1}) = \sum_{j=1}^{K} w_j(\mathbf{s})\mathcal{L}_j + \sum_{j=1}^{K}(s_j + c) = \mathcal{J}(\mathbf{s}) + K \cdot c. \tag{23}$$

Taking the limit as $c \to -\infty$:

$$\lim_{c \to -\infty} \mathcal{J}(\mathbf{s} + c\mathbf{1}) = -\infty. \tag{24}$$

Without a stop-gradient operator, the optimizer can exploit this scale symmetry by driving all $s_i \to -\infty$ at a uniform rate. The relative task weights w_i remain fixed, but the total loss vanishes to negative infinity, completely collapsing training stability.

D.3 Guaranteed Stability via Stop-Gradient Execution

In our implementation, we enforce stop-gradient execution by calculating $w_{i,t}$ inside a non-differentiable step routine. In the computational graph for $\mathcal{L}_{\text{total}}$, the weights $w_{i,t}$ are treated as constant coefficients:

$$\mathcal{L}_{\text{total}} = \sum_{i \in \mathcal{A}} \text{sg}[w_{i,t}]\mathcal{L}_{i,t} + \sum_{i \in \mathcal{A}} s_i. \tag{25}$$

Because $\frac{\partial \text{sg}[w_{i,t}]}{\partial s_j} = 0$, the partial derivative with respect to any active log-variance simplifies strictly to:

$$\frac{\partial \mathcal{L}_{\text{total}}}{\partial s_i} = 1, \qquad \forall i \in \mathcal{A}. \tag{26}$$

This structural disconnection guarantees that:

1. The cross-derivatives $\frac{\partial w_j}{\partial s_i}$ are zeroed out, eliminating the scale-invariance drift $c \to -\infty$.
2. Task loss gradients $\nabla_\theta \mathcal{L}_{i,t}$ update the model parameters θ, while s_i is updated exclusively by its regularizer, isolating it as a stable baseline estimator.

E Gradient Analysis of Projector–Student Responsibilities

Here we derive the optimization behavior underlying the projector–student separation discussed in Section 7.1. Let

$$\mathbf{a}_s \in \mathbb{R}^{d_s}, \qquad \mathbf{a}_t \in \mathbb{R}^{d_t}, \qquad d_s < d_t, \tag{27}$$

denote student and teacher activations for a single sample. For student parameters $\boldsymbol{\theta} \in \mathbb{R}^p$, define

$$\mathbf{J}_{\boldsymbol{\theta}} = \frac{\partial \mathbf{a}_s}{\partial \boldsymbol{\theta}^\top} \in \mathbb{R}^{d_s \times p}. \tag{28}$$

E.1 Fixed Downward Projection

For the fixed downward-projection loss in Section 7.1, differentiation with respect to the student activation gives

$$\nabla_{\mathbf{a}_s}\mathcal{L}_{\text{down}} = \frac{2}{d_s}\left(\mathbf{a}_s - \mathbf{H}_{\text{fix}}\mathbf{a}_t\right). \tag{29}$$

Applying the chain rule yields

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{down}} = \frac{2}{d_s}\mathbf{J}_{\boldsymbol{\theta}}^{\top}\left(\mathbf{a}_s - \mathbf{H}_{\text{fix}}\mathbf{a}_t\right). \tag{30}$$

Since

$$\Delta\mathbf{H}_{\text{fix}} = \mathbf{0}, \tag{31}$$

the projected teacher activation is stationary for a fixed input. The projector influences learning only through the target $\mathbf{H}_{\text{fix}}\mathbf{a}_t$; it neither adapts to reduce the loss nor transforms the residual during its pullback through the student.

E.2 Learned Upward Projection

For the learned upward-projection loss, define the residual

$$\mathbf{r}_{\text{up}} = \mathbf{H}_{\text{learn}}\mathbf{a}_s - \mathbf{a}_t. \tag{32}$$

The projector gradient is

$$\nabla_{\mathbf{H}_{\text{learn}}}\mathcal{L}_{\text{up}} = \frac{2}{d_t}\mathbf{r}_{\text{up}}\mathbf{a}_s^{\top}. \tag{33}$$

Thus, the projector can directly reduce the matching residual. The student-activation gradient is

$$\nabla_{\mathbf{a}_s}\mathcal{L}_{\text{up}} = \frac{2}{d_t}\mathbf{H}_{\text{learn}}^{\top}\mathbf{r}_{\text{up}}, \tag{34}$$

which gives

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{up}} = \mathbf{J}_{\boldsymbol{\theta}}^{\top}\nabla_{\mathbf{a}_s}\mathcal{L}_{\text{up}} \tag{35}$$

$$= \frac{2}{d_t}\mathbf{J}_{\boldsymbol{\theta}}^{\top}\mathbf{H}_{\text{learn}}^{\top}\left(\mathbf{H}_{\text{learn}}\mathbf{a}_s - \mathbf{a}_t\right). \tag{36}$$

The learned projector therefore participates in optimization in two ways: it determines the residual and transforms that residual through $\mathbf{H}_{\text{learn}}^{\top}$ before it reaches the student Jacobian. Both effects change as the projector is updated. Consequently, low matching loss does not imply that the student alone has reproduced the teacher representation; part of the alignment may reside in the learned projector. Fixing the downward projection removes this ambiguity and ensures that only the student adapts during distillation.

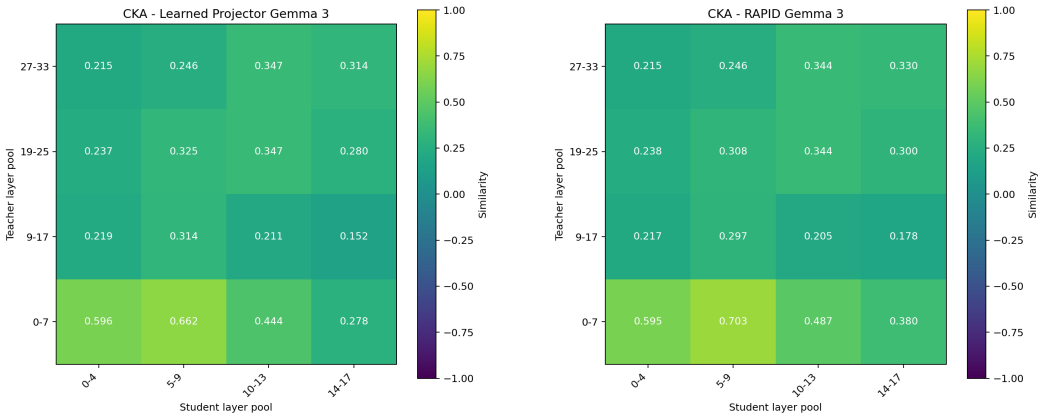
F Representational Alignment Results

Model Family	Method	Global			Mapped		
		Mean ± Std.	Median	P_{25}	Mean ± Std.	Median	P_{25}
Gemma 3	Learned Projector	0.3318 ± 0.1819	0.2985	0.2017	$\mathbf{0.3791} \pm 0.1298$	0.3630	**0.2914**
	RAPID	$\mathbf{0.3442} \pm 0.1875$	**0.3231**	**0.2069**	0.3739 ± 0.1287	**0.3648**	0.2868
Qwen 2.5	Learned Projector	$\mathbf{0.7860} \pm 0.1651$	**0.8410**	**0.7061**	$\mathbf{0.8963} \pm 0.0883$	**0.9295**	**0.8917**
	RAPID	0.7810 ± 0.1693	0.8395	0.6944	0.8935 ± 0.0921	0.9275	0.8874

Table 4: CKA representational similarity results for the learned projector and RAPID distillation methods.

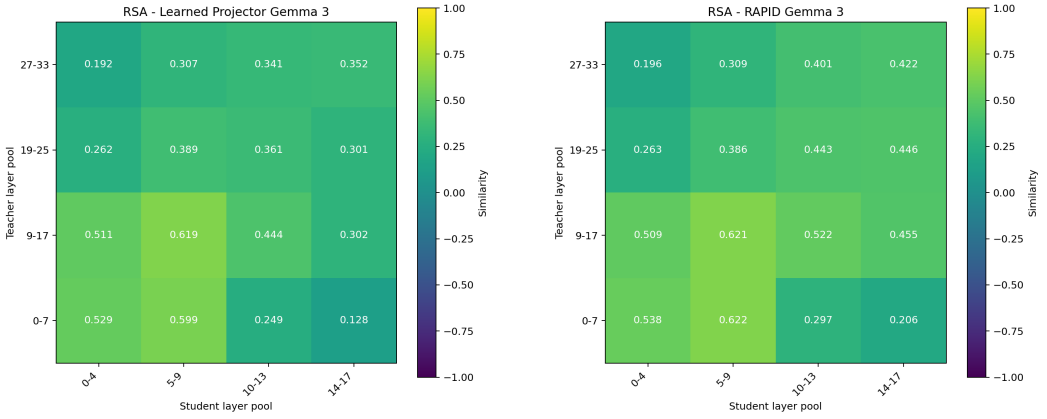
Model Family	Method	Global			Mapped		
		Mean ± Std.	Median	P_{25}	Mean ± Std.	Median	P_{25}
Gemma 3	Learned Projector	0.3815 ± 0.1856	0.3440	0.2551	0.4745 ± 0.1647	0.4212	0.3590
	RAPID	$\mathbf{0.4239} \pm 0.1729$	**0.4215**	**0.3171**	$\mathbf{0.5063} \pm 0.1393$	**0.4603**	**0.4321**
Qwen 2.5	Learned Projector	$\mathbf{0.6926} \pm 0.1903$	**0.7400**	**0.5839**	$\mathbf{0.8515} \pm 0.0880$	**0.8757**	**0.8439**
	RAPID	0.6866 ± 0.1939	0.7363	0.5771	0.8454 ± 0.0957	0.8715	0.8381

Table 5: RSA representational similarity results for the learned projector and RAPID distillation methods.



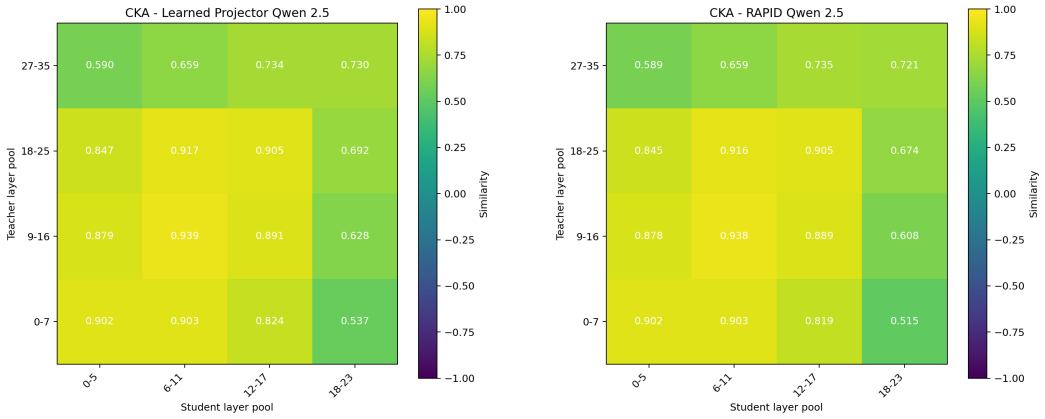
(a) Learned projector (b) Rapid

Figure 2: Pooled-layer CKA scores for the Gemma 3 distillation pair comparing RAPID against the traditional learned projector method.



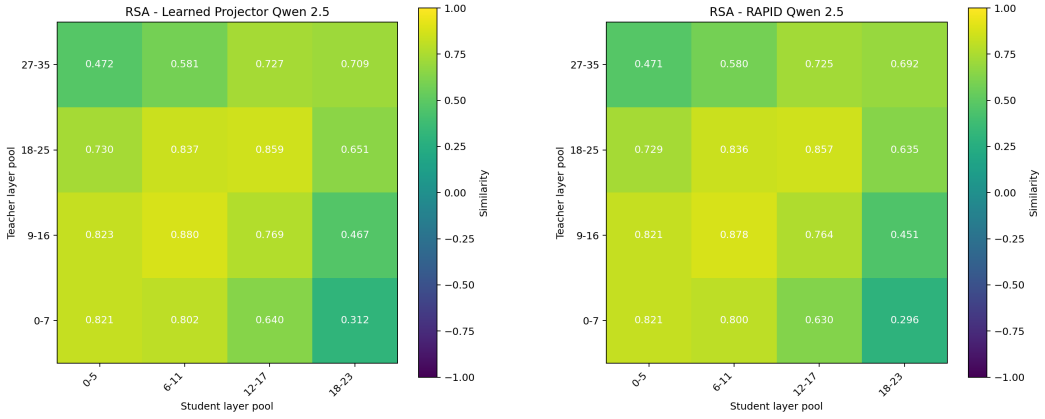
(a) Learned projector (b) Rapid

Figure 3: Pooled-layer RSA scores for the Gemma 3 distillation pair comparing RAPID against the traditional learned projector method.



(a) Learned projector (b) Rapid

Figure 4: Pooled-layer CKA scores for the Qwen 2.5 distillation pair comparing RAPID against the traditional learned projector method.



(a) Learned projector (b) Rapid

Figure 5: Pooled-layer RSA scores for the Qwen 2.5 distillation pair comparing RAPID against the traditional learned projector method.

G TRAINING HYPERPARAMETERS

Hyperparameter	Value
Learning rate	3×10^{-5}
Weight decay	1×10^{-4}
Maximum gradient norm	2.0
Distillation temperature	1.1

Table 6: Training hyperparameters used across all experiments