TBRD: Task-Aligned LLM Distillation via Brier Relations over Candidate Tokens

Anonymous authors
Paper under double-blind review

Abstract

Knowledge distillation transfers knowledge from a teacher model to a student by matching their output distributions. However, distribution matching supervises candidate preferences implicitly. Its updates can also conflict with the learning signal from the ground-truth token. We propose TBRD to explicitly model candidate relations and adapt the strength of teacher supervision. TBRD includes the teacher's highest-probability token in the student candidate set, models pairwise preferences, and adjusts the weight at each position using the agreement between distillation and task gradients. TBRD improves the average score across nine evaluation tasks by 2.09% over TAD, the strongest baseline in this comparison. These results support jointly modeling candidate relations and adapting supervision to the student's task signal.

1 Introduction

Knowledge distillation (KD) trains a smaller student using a teacher's predictions, reducing inference and memory costs. Standard KD matches their output distributions at each token position, supervising candidate preferences implicitly. Teacher matching and next-token learning can favor conflicting updates. When the teacher assigns less probability to the ground-truth token than the student, a distillation update can lower that probability. Cross-entropy instead pushes it upward. This conflict motivates selecting which candidate relations to learn and deciding how strongly to supervise them at each position.

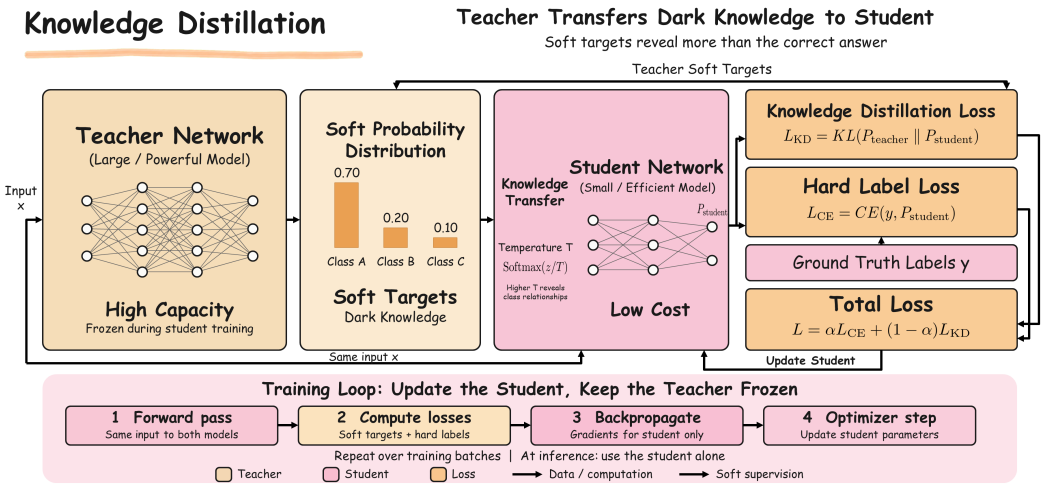


Figure 1: Knowledge distillation overview. A frozen teacher supplies output distributions for training a student. The figure shows standard KD with an observed-token loss. TBRD uses that loss as a gradient reference and trains the student with weighted distillation, as described in Section 4.

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.

Existing methods use teacher scores to select data, modify distribution matching, or redistribute supervision across probability components and token positions (Gu et al., 2024b;a; Zhao et al., 2022; Dasgupta et al., 2026; Zhong et al., 2024). Relational methods such as RKD and BiLD transfer comparisons between examples or candidate tokens (Park et al., 2019; Li et al., 2025). These approaches organize teacher knowledge, but their supervision rules do not directly compare distillation and task gradients. Strengthening supervision can therefore amplify updates that conflict with next-token learning.

We propose Task-Aligned Brier Relational Distillation (TBRD) to connect candidate-relation learning with task-guided supervision. At each prediction position, TBRD includes the teacher's highest-probability token in a candidate set drawn from the student. The teacher's preferred token can then participate in relation learning even when its student probability is low. For each candidate pair, TBRD maps the logit difference to a preference probability and matches teacher and student preferences through Brier squared error. The relation loss captures competition between candidates, while distribution matching supervises probability allocation across the vocabulary.

TBRD then computes the gradients of the full distillation loss and ground-truth cross-entropy with respect to student logits. Their directional agreement measures whether distillation supports the current task signal at that position. TBRD increases supervision at positions whose agreement exceeds the sequence average and reduces it at positions below that average. The mean weight remains one, preserving the overall weight assigned to the sequence. Cross-entropy supplies the task reference, and the weighted distillation loss updates the student with the position weights held fixed.

TBRD improves the average score on nine benchmarks by 0.72 percentage points (2.09%) over TAD. It achieves the highest average score among the compared methods at all three training budgets. Test-set evaluations also show gains in teacher distribution matching, next-token accuracy, and calibration.

Our contributions are three-fold:

1. We examine a position-level conflict in KD: updates that reduce teacher-student discrepancy can oppose the cross-entropy gradient from the observed token. This motivates using task-gradient agreement to assess teacher supervision.
2. We propose TBRD to learn candidate preferences and allocate supervision using the full distillation gradient. The allocation favors positions with greater task-gradient agreement while preserving the mean sequence weight.
3. TBRD achieves the highest average task score at three training budgets, exceeding TAD by 0.72 percentage points (2.09%) at 392M tokens. It also achieves the best teacher-student KL, cross-entropy, next-token accuracy, and calibration error in the test-set comparison.

2 Related Work

2.1 Language Model Distillation

Soft-target distillation matches teacher and student output distributions (Hinton et al., 2015). Sequence-KD extends distillation to generated sequences, while DistilBERT and TinyBERT apply it to pretraining and representation learning (Kim & Rush, 2016; Sanh et al., 2019; Jiao et al., 2020). MiniLLM adjusts the matching objective, GKD uses student-generated contexts, and MiniPLM selects training text using teacher scores (Gu et al., 2024a; Agarwal et al., 2024; Gu et al., 2024b). DPO uses preference data for supervision (Rafailov et al., 2023). TBRD uses a fixed corpus and adjusts teacher supervision with the observed next token as its task reference.

2.2 Supervision Organization and Allocation

DKD separates target-class and non-target knowledge, TAD adjusts head and tail supervision, and ATKD considers teaching differences across tokens (Zhao et al., 2022; Dasgupta

et al., 2026; Zhong et al., 2024). Distillation benefits also depend on the student and training setting (Mirzadeh et al., 2020; Cho & Hariharan, 2019). Gradient-based methods use task objectives to reweight examples, assess auxiliary supervision, and resolve task conflicts (Ren et al., 2018; Du et al., 2018; Yu et al., 2020). TBRD compares the full distillation gradient with the task gradient at each position to allocate supervision within a sequence.

2.3 Relational Knowledge and Probability Scoring

FitNets transfers intermediate representations, RKD matches relations between examples, and MiniLM transfers attention relations (Romero et al., 2014; Park et al., 2019; Wang et al., 2020). BiLD matches logit differences between candidate tokens (Li et al., 2025). TBRD matches token-pair preference probabilities through Brier squared error. For a fixed teacher target, this error equals the excess expected quadratic score (Glenn et al., 1950; Gneiting & Raftery, 2007). TBRD combines the relation and distribution losses, then uses their full gradient to determine supervision strength.

3 Preliminaries

Task and notation. A frozen teacher T and a student S_θ share vocabulary $\mathcal{V}$. For a sequence x from corpus $\mathcal{D}$, both predict $y_i = x_{i+1}$ from prefix $x_{\le i}$. Model $M \in \{\mathrm{T}, \mathrm{S}\}$ produces logits z_i^M and probabilities $p_i^M = \mathrm{softmax}(z_i^M)$, where S denotes the student. The set $\Omega(x)$ contains valid prediction positions, with $N_x = |\Omega(x)|$.

Candidate distributions. At each position, we divide the vocabulary into candidate set $\mathcal{I}_i$ and remainder $\mathcal{R}_i$. A group G has probability mass $a_{i,G}^M = \sum_{v \in G} p_i^M(v)$ and conditional distribution $\widetilde{p}_{i,G}^M(v) = p_i^M(v)/a_{i,G}^M$. Writing the two group masses as $b_i^M = (a_{i,\mathcal{I}_i}^M, a_{i,\mathcal{R}_i}^M)$ gives

$$D_{\mathrm{KL}}(p_i^{\mathrm{T}} \| p_i^{\mathrm{S}}) = D_{\mathrm{KL}}(b_i^{\mathrm{T}} \| b_i^{\mathrm{S}}) + \sum_{G \in \{\mathcal{I}_i, \mathcal{R}_i\}} a_{i,G}^{\mathrm{T}} D_{\mathrm{KL}}(\widetilde{p}_{i,G}^{\mathrm{T}} \| \widetilde{p}_{i,G}^{\mathrm{S}}). \tag{1}$$

This decomposition separates probability mass between groups from token preferences within each group.

Candidate relations. We form unordered token pairs $\mathcal{P}_i = \{(u, v) : u, v \in \mathcal{I}_i,\ u < v\}$, using vocabulary indices to fix their order. For each pair, the model's preference for u over v is

$$q_{i,uv}^M = \frac{p_i^M(u)}{p_i^M(u) + p_i^M(v)} = \sigma\big(z_i^M(u) - z_i^M(v)\big)\,, \tag{2}$$

where σ is the sigmoid function.

4 Method

TBRD processes each corpus prefix with a frozen teacher and a trainable student. It constructs distribution and relation losses, compares their full gradient with the task gradient, and weights the loss at each position. Figure 2 shows this training procedure.

4.1 Teacher-Anchored Candidate Relations

TBRD forms a pool from the student's high-probability tokens. If the teacher's most probable token falls outside the pool, it replaces the candidate with the lowest student probability. We normalize the teacher probabilities within this pool and use their entropy to estimate the effective support. This estimate determines the candidate count relative to other positions in the sequence. We then retain the tokens with the highest teacher probabilities as $\mathcal{I}_i$, including the teacher's preferred token. Appendix A gives the selection rule and budget bounds. Candidate indices remain fixed during differentiation, while selected token probabilities retain their gradients.

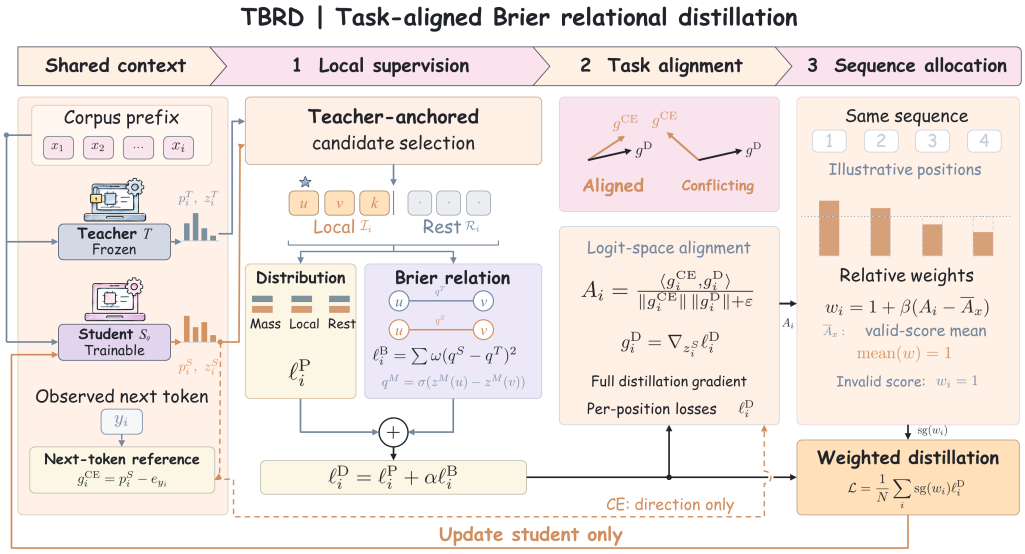


Figure 2: TBRD framework. Teacher-anchored selection supplies candidates for distribution and relation losses. TBRD compares the full distillation gradient with the observed-token gradient, adjusts position weights relative to the sequence average, and updates the student with the weighted loss.

The distribution loss matches the two group masses and the conditional distribution within each group:

$$\ell_i^{\mathrm{P}} = D_{\mathrm{KL}}(b_i^{\mathrm{T}} \| b_i^{\mathrm{S}}) + \sum_{G \in \{\mathcal{I}_i, \mathcal{R}_i\}} D_{\mathrm{KL}}(\widetilde{p}_{i,G}^{\mathrm{T}} \| \widetilde{p}_{i,G}^{\mathrm{S}}).$$

Each conditional term has coefficient one, giving both groups direct supervision. In contrast, the KL decomposition in Eq. (1) scales each term by its teacher probability mass.

TBRD matches each pair through $B_{i,uv} = (q_{i,uv}^{\mathrm{S}} - q_{i,uv}^{\mathrm{T}})^2$. For a fixed teacher preference, this error is minimized when the two probabilities agree (Glenn et al., 1950; Gneiting & Raftery, 2007). The pair score and normalized weight are

$$s_{i,uv} = e^{-\gamma |p_i^{\mathrm{S}}(u) - p_i^{\mathrm{S}}(v)|} \left[p_i^{\mathrm{S}}(u) + p_i^{\mathrm{S}}(v) \right], \quad \omega_{i,uv} = \frac{s_{i,uv}}{\sum_{(a,b) \in \mathcal{P}_i} s_{i,ab} + \epsilon_{\mathrm{pair}}}. \tag{3}$$

The probability sum favors pairs with greater student mass, while the exponential factor favors pairs with smaller probability gaps. Parameter $\gamma \geq 0$ controls this effect, and $\epsilon_{\mathrm{pair}} > 0$ stabilizes normalization.

TBRD forms the relation loss $\ell_i^{\mathrm{B}} = \sum_{(u,v) \in \mathcal{P}_i} \omega_{i,uv} B_{i,uv}$ and combines it with distribution supervision as $\ell_i^{\mathrm{D}} = \ell_i^{\mathrm{P}} + \alpha \ell_i^{\mathrm{B}}$. We calibrate $\alpha > 0$ from gradient norms at initialization and keep it fixed during training. Both the Brier errors and the pair weights retain their derivatives. For $\Delta = z_i^{\mathrm{S}}(u) - z_i^{\mathrm{S}}(v)$, $q = \sigma(\Delta)$, and $r = q_{i,uv}^{\mathrm{T}}$, the Brier derivative is $\partial B_{i,uv} / \partial \Delta = 2(q - r)q(1 - q)$. It moves the student's pairwise preference toward the teacher's, with weaker gradients near saturated probabilities. The relation and distribution losses supervise different quantities. Pairwise preferences depend on logit differences within the candidate set and remain unchanged when all candidate logits receive the same offset. They therefore describe competition among candidates while leaving their total probability mass unspecified. The distribution loss supervises the mass assigned to the candidate set and its remainder, together with the conditional distribution within each group. The combined objective supervises both probability allocation and pairwise preferences.

4.2 Comparing Distillation and Task Directions

TBRD compares the full distillation gradient with the task gradient in logit space. The observed-token loss is $\ell_i^{\mathrm{CE}} = -\log p_i^{\mathrm{S}}(y_i)$, with gradient $g_i^{\mathrm{CE}} = p_i^{\mathrm{S}} - e_{y_i}$, where e_{y_i} is the one-hot target. The full distillation gradient is $g_i^{\mathrm{D}} = \nabla_{z_i^{\mathrm{S}}} \ell_i^{\mathrm{D}}$. Their agreement is measured by $A_i = \frac{\langle g_i^{\mathrm{CE}}, g_i^{\mathrm{D}} \rangle}{\|g_i^{\mathrm{CE}}\|_2 \|g_i^{\mathrm{D}}\|_2 + \epsilon_{\mathrm{align}}}$.

Here, $\epsilon_{\mathrm{align}} > 0$ stabilizes the denominator. To interpret the score, consider a small logit update $z_i^{\mathrm{S}} \leftarrow z_i^{\mathrm{S}} - \eta g_i^{\mathrm{D}}$. A first-order expansion gives

$$\ell_i^{\mathrm{CE}}(z_i^{\mathrm{S}} - \eta g_i^{\mathrm{D}}) - \ell_i^{\mathrm{CE}}(z_i^{\mathrm{S}}) = -\eta \langle g_i^{\mathrm{CE}}, g_i^{\mathrm{D}} \rangle + O(\eta^2). \tag{4}$$

A positive inner product predicts a local decrease in cross-entropy for a sufficiently small step, while a negative inner product predicts an increase. Normalization makes A_i measure directional agreement; the size of the loss change also depends on the gradient norms.

The direction score uses the full distillation gradient:

$$g_i^{\mathrm{D}} = \nabla_{z_i^{\mathrm{S}}} \ell_i^{\mathrm{P}} + \alpha \sum_{(u,v) \in \mathcal{P}_i} \left[\omega_{i,uv} \nabla_{z_i^{\mathrm{S}}} B_{i,uv} + B_{i,uv} \nabla_{z_i^{\mathrm{S}}} \omega_{i,uv} \right]. \tag{5}$$

The first term inside the sum changes pairwise preferences. The second accounts for changes in the student-dependent pair weights. Retaining both terms makes the score reflect the gradient of the combined objective used for distillation. Cross-entropy supplies the task reference, while the full distillation loss supplies the training objective. The expansion describes a local update in logit space. Parameter updates also combine contributions across positions through the shared student model.

4.3 Allocating Supervision Within a Sequence

TBRD uses direction scores to adjust supervision across positions in the same sequence. Let $\mathcal{C}_x \subseteq \Omega(x)$ contain positions with valid scores. Each position receives weight

$$w_i = \begin{cases} 1 + \beta \left(A_i - \dfrac{1}{|\mathcal{C}_x|} \displaystyle\sum_{j \in \mathcal{C}_x} A_j \right), & i \in \mathcal{C}_x, \\ 1, & i \in \Omega(x) \setminus \mathcal{C}_x. \end{cases} \tag{6}$$

Positions above the mean score receive more supervision, and positions below it receive less. The average weight stays at one. Parameter β controls the adjustment; scores in $[-1, 1]$ and $0 < \beta \leq 1/2$ give nonnegative weights. Invalid scores receive unit weight. If $\mathcal{C}_x$ is empty, every position receives unit weight.

Before backpropagation, TBRD detaches the position weights as $\widehat{w}_i = \mathrm{sg}(w_i)$, where sg denotes stop-gradient. The training objective is

$$\mathcal{L}_{\mathrm{TBRD}} = \mathbb{E}_{x \sim \mathcal{D}} \left[\frac{1}{N_x} \sum_{i \in \Omega(x)} \widehat{w}_i \ell_i^{\mathrm{D}} \right]. \tag{7}$$

Algorithm 1 summarizes one TBRD update. The teacher remains frozen. Pair weights retain their gradients, while position weights are detached before backpropagation.

4.4 Properties of Supervision Allocation

TBRD favors positions with greater task agreement while preserving the mean supervision weight. For fixed scores and $m = |\mathcal{C}_x| > 0$, its weights solve

$$\max_{\substack{w_i \geq 0 \\ \sum_{i \in \mathcal{C}_x} w_i = m}} \sum_{i \in \mathcal{C}_x} \left[w_i A_i - \frac{(w_i - 1)^2}{2\beta} \right]. \tag{8}$$

Algorithm 1 TBRD

Initialize student S_θ, freeze teacher T, calibrate α, and set β.
for each training batch $\mathcal{B}$ do
  Obtain teacher probabilities p_i^{T} and student outputs $(z_i^{\mathrm{S}}, p_i^{\mathrm{S}})$ at each valid position.
  Form teacher-anchored candidate sets $\mathcal{I}_i$ and token pairs $\mathcal{P}_i$.
  Compute distribution loss ℓ_i^{P} and Brier relation loss ℓ_i^{B}.
  Form $\ell_i^{\mathrm{D}} = \ell_i^{\mathrm{P}} + \alpha \ell_i^{\mathrm{B}}$.
  Compute $g_i^{\mathrm{D}} = \nabla_{z_i^{\mathrm{S}}} \ell_i^{\mathrm{D}}$, retaining pair-weight derivatives.
  Set $g_i^{\mathrm{CE}} = p_i^{\mathrm{S}} - e_{y_i}$ and compute agreement A_i as defined in Section 4.2.
  Assign w_i within each sequence using Eq. (6).
  Set $\widehat{w}_i = \mathrm{sg}(w_i)$.
  Form $\mathcal{L}_{\mathcal{B}} = \dfrac{1}{|\mathcal{B}|} \displaystyle\sum_{x \in \mathcal{B}} \frac{1}{N_x} \sum_{i \in \Omega(x)} \widehat{w}_i \ell_i^{\mathrm{D}}$.
  Update θ using $\nabla_\theta \mathcal{L}_{\mathcal{B}}$.
end for

The quadratic term controls deviations from uniform weighting. For $A_i \in [-1, 1]$ and $0 < \beta \leq 1/2$, the unique solution is $w_i = 1 + \beta(A_i - \overline{A}_x)$, where $\overline{A}_x = m^{-1} \sum_{i \in \mathcal{C}_x} A_i$. These weights are nonnegative and have mean one.

The resulting weighted agreement satisfies

$$\frac{1}{m} \sum_{i \in \mathcal{C}_x} w_i A_i = \overline{A}_x + \frac{\beta}{m} \sum_{i \in \mathcal{C}_x} (A_i - \overline{A}_x)^2. \tag{9}$$

Thus, allocation improves the weighted agreement whenever scores differ and recovers uniform weighting when they are equal. Positions with invalid scores retain unit weight, preserving the mean across the full sequence.

The allocation favors agreement relative to the sequence average. Equal scores give unit weights; when all scores are negative, positions with weaker conflict can still receive greater weight. The mean-weight constraint preserves the average coefficient, while the weighted loss and gradient norm may change. With detached weights, the sequence gradient is $\nabla_\theta \mathcal{L}_x = N_x^{-1} \sum_{i \in \Omega(x)} \widehat{w}_i J_i^\top g_i^{\mathrm{D}}$, where $J_i = \partial z_i^{\mathrm{S}} / \partial \theta$. Task improvement therefore also depends on shared parameters and the optimizer. Appendix B provides the mathematical details.

5 Experiments

5.1 Experimental Setup

We train a randomly initialized 200M student with a frozen Qwen1.5-1.8B teacher (Bai et al., 2023). We evaluate the student after approximately 78M, 196M, and 392M training tokens, corresponding to 10,000, 25,000, and 50,000 updates, respectively. The zero-shot tasks are ARC-Easy, ARC-Challenge, HellaSwag, LAMBADA, OpenBookQA, PIQA, Social IQA, and WinoGrande (Clark et al., 2018; Zellers et al., 2019; Paperno et al., 2016; Mihaylov et al., 2018; Bisk et al., 2020; Sap et al., 2019; Sakaguchi et al., 2021). Evaluations at the later two budgets also include StoryCloze 2016. All methods at a given budget use the same task set.

We compare TBRD with Pretrain, Vanilla KD, PD, ATKD, RLD, LDRLD, TAD, and BiLD at all three training budgets. At 392M tokens, we also evaluate the nine methods on a shared test set containing 2,048 texts and 1,000,397 valid tokens. We report teacher-student KL divergence, cross-entropy (CE), perplexity (PPL), next-token accuracy, and expected calibration error (ECE). Appendix C gives the evaluation protocol.

5.2 Performance on Evaluation Tasks

For the student with 200M parameters, we evaluate TBRD after approximately 78M, 196M, and 392M training tokens, corresponding to 10,000, 25,000, and 50,000 updates, respectively. TBRD achieves the highest average score among the compared methods at each budget: 30.60%, 33.66%, and 35.09%. The gains over the strongest baseline are 0.49, 0.31, and 0.72 percentage points, respectively; the corresponding gains over Vanilla KD are 0.58, 0.44, and 1.35 percentage points. At the largest training budget, the gain over TAD is 2.09% in relative terms. Figure 3(b) summarizes these results, and Table 1 reports the scores after 392M training tokens.

The 200M-parameter student leads on HellaSwag, LAMBADA, and PIQA after 78M training tokens, and on ARC-Easy after 196M tokens. After 392M tokens, it ranks first on HellaSwag, LAMBADA, PIQA, and StoryCloze 2016. Each budget uses the same task set across methods: eight tasks at 78M tokens and nine at the later budgets.

Table 1: Task scores (%) for the student with 200M parameters. The average covers nine tasks. Bold marks the best result in each column, and underlining marks the second-best.

Method	ARC-C	ARC-E	HS	LAMB	OBQA	PIQA	SIQA	SC16	WG	Avg. (9)
Pretrain	18.94	32.11	26.61	5.59	23.80	54.57	33.78	51.31	50.83	33.06
Vanilla KD	18.60	31.57	27.02	10.46	23.80	56.69	33.67	51.31	50.51	33.74
PD	18.00	33.46	26.41	<u>11.74</u>	22.80	57.67	**34.60**	51.79	49.25	33.97
ATKD	<u>19.62</u>	35.82	27.00	6.23	25.40	57.34	<u>34.49</u>	51.90	47.59	33.93
RLD	18.00	34.93	26.95	4.44	25.40	57.13	33.37	51.79	**53.04**	33.89
LDRLD	17.92	**37.08**	<u>27.20</u>	6.66	25.20	57.62	34.34	<u>52.32</u>	49.80	34.24
TAD	18.26	35.98	27.03	6.09	<u>27.00</u>	<u>58.05</u>	34.24	51.58	<u>51.07</u>	<u>34.37</u>
BiLD	**21.93**	25.59	26.78	3.57	**30.00**	54.79	32.34	47.57	50.20	32.53
TBRD	17.66	<u>36.78</u>	**27.57**	**13.12**	25.20	**58.71**	34.24	**53.02**	49.49	**35.09**

Results with 500M parameters. We also compare TBRD with Vanilla KD using a student with 500M parameters. As shown in Table 2, TBRD achieves higher scores on all nine tasks. Its average score increases from 33.80% to 35.56%, a gain of 1.76 percentage points over Vanilla KD. This comparison shows that the gains over Vanilla KD also extend to the larger student.

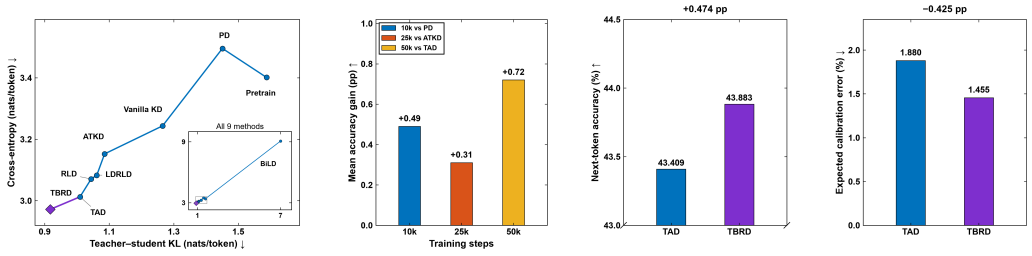
Table 2: Task scores (%) for the student with 500M parameters. Bold marks the higher score for each task and the average. The last row reports differences in percentage points, computed from the displayed scores.

Method	ARC-C	ARC-E	HS	LAMB	OBQA	PIQA	SIQA	SC16	WG	Avg. (9)
Vanilla KD	18.26	33.33	26.99	10.32	25.20	57.45	33.83	51.20	47.59	33.80
TBRD	**18.69**	**37.42**	**27.38**	**13.14**	**26.40**	**59.09**	**33.93**	**52.32**	**51.70**	**35.56**
TBRD − Vanilla KD	+0.43	+4.09	+0.39	+2.82	+1.20	+1.64	+0.10	+1.12	+4.11	+1.76

5.3 Prediction Quality on the Test Set

TBRD achieves the lowest teacher-student KL divergence, cross-entropy, and perplexity among the nine models (Table 3). Compared with TAD, the strongest baseline on these metrics, TBRD reduces KL from 1.0096 to 0.9172, a relative reduction of approximately 9.15%. Cross-entropy falls from 3.0126 to 2.9717 nats per token, and perplexity falls from 20.3393 to 19.5247. TBRD thus matches the teacher distribution more closely and achieves a lower language-modeling loss on this test set.

TBRD achieves 43.883% next-token accuracy, compared with 43.409% for TAD, and reduces calibration error from 1.880% to 1.455%. Its predictions are more accurate, and its confidence better reflects accuracy under this measure. Perplexity is the exponential of cross-entropy, so both reflect the same improvement. Accuracy and calibration assess additional aspects of prediction quality. Figure 3 summarizes teacher matching, task-score gains, and prediction quality.



(a) Teacher Fit (b) Downstream Gain (c) Token Accuracy (d) Calibration Error

Figure 3: Distillation quality and task performance of the 200M student. (a) Teacher-student KL divergence and cross-entropy at 392M training tokens. The inset includes all nine methods; connecting lines guide the eye across methods. (b) Average task-score gains over the strongest baseline at each training budget. The 10k, 25k, and 50k checkpoints correspond to approximately 78M, 196M, and 392M training tokens, with averages over eight, nine, and nine tasks, respectively. (c) Next-token accuracy and ECE for TBRD and TAD on the shared test set at 392M tokens.

Table 3: Prediction quality on the shared test set after 392M training tokens. ECE, accuracy (Acc.), and mean confidence (Conf.) are percentages. Lower KL, CE, PPL, and ECE are better; higher accuracy is better. The best result for each performance metric is bold, and the second-best is underlined. Mean confidence is descriptive.

Method	KL	CE	PPL	ECE (%)	Acc. (%)	Conf. (%)
Pretrain	1.5878	3.4011	29.9967	3.129	41.505	44.367
Vanilla KD	1.2655	3.2435	25.6236	6.581	42.992	36.499
PD	1.4510	3.4955	32.9660	15.619	43.059	58.608
ATKD	1.0858	3.1523	23.3900	4.524	39.456	35.002
RLD	1.0441	3.0706	21.5541	5.482	41.522	36.041
LDRLD	1.0616	3.0829	21.8225	2.832	40.767	43.599
TAD	<u>1.0096</u>	<u>3.0126</u>	<u>20.3393</u>	<u>1.880</u>	<u>43.409</u>	42.639
BiLD	7.0075	9.0760	8743.3437	35.985	40.221	4.236
TBRD	**0.9172**	**2.9717**	**19.5247**	**1.455**	**43.883**	44.378

These results show gains in teacher matching and language modeling for the evaluated runs. The runs use different random seeds and training-data prefixes. Matched component comparisons are therefore needed to attribute the gains to relation learning or position weighting.

5.4 Ablation Study

The ablation varies the Brier loss and direction-based position weighting while keeping candidate selection and distribution supervision fixed. It compares four variants: distribution supervision alone, distribution supervision with Brier relations, distribution supervision with position weighting, and the full TBRD method. Removing the Brier term also changes the gradient used for direction scoring. Uniform weighting assigns one to every position. Results at the main training budgets remain pending.

The short runs use matched initialization and data order. TBRD improves cross-entropy and token accuracy over uniform relational KL in both recorded pairs. With Brier relations held fixed, however, uniform position weights achieve better results on both measures. These comparisons support the full configuration over uniform relational KL, while the Brier comparison favors uniform position weights. Appendix C reports the paired results.

5.5 Training Dynamics

Figure 4 shows the relative training loss of TBRD and seven baselines over the first 50,000 updates, corresponding to approximately 392M training tokens. TBRD's loss drops rapidly during early training and continues to decrease at a slower rate as training progresses. This trend complements the task evaluations by showing how its training objective evolves. Each method uses its own loss function, so comparisons of prediction quality rely on the shared evaluation metrics reported above.

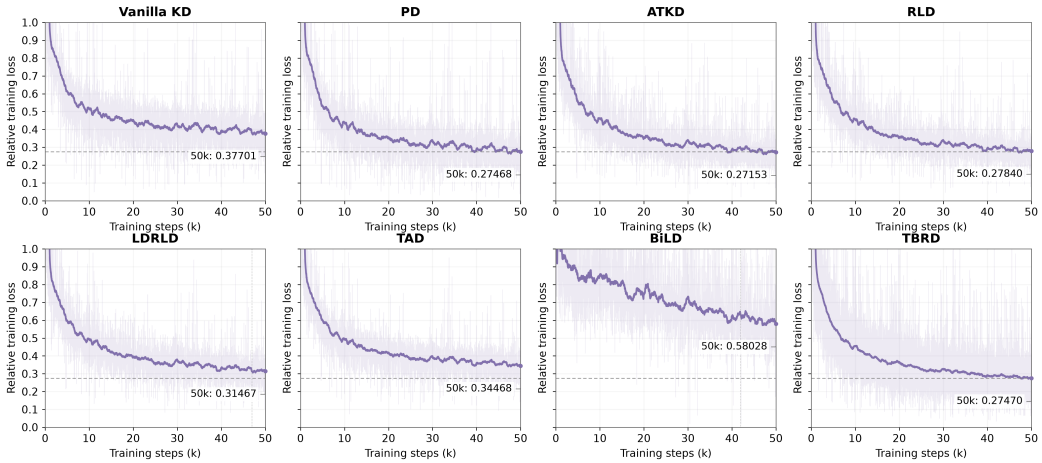


Figure 4: Training loss over approximately 392M tokens. Relative training loss for TBRD and seven baselines during the first 50,000 updates. Each panel tracks the corresponding method's training objective. Relative loss values describe progress within each method and have different meanings across objectives.

6 Conclusion and Limitations

TBRD combines candidate-relation learning with supervision guided by the student's task signal. It includes the teacher's preferred token, matches pairwise preferences, and uses agreement between the full distillation gradient and the observed-token gradient to adjust position weights. TBRD achieves the highest average task score across three training budgets among the compared methods. At 392M tokens, it also achieves the best teacher-student KL, cross-entropy, calibration error, and next-token accuracy on the shared test set.

TBRD assumes a shared vocabulary and uses the observed next token as its task reference. Its direction analysis concerns logits; parameter updates also depend on shared representations and the optimizer. Saturated preference probabilities weaken Brier gradients and can limit correction of confidently incorrect preferences. The available Brier ablation favors uniform position weights, so the conditions under which direction-based weighting helps remain open. Further seeds, component comparisons, and student sizes are needed to establish those conditions.

AI Use Statement

Large language models were used only for language polishing, including improving clarity, wording, and grammar. They were not involved in research ideas, model design, dataset construction, experiments, or result analysis. All content, results, figures, and references were reviewed and verified by the authors, who take full responsibility for the work.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In International Conference on Learning Representations, volume 2024, pp. 21246–21263, 2024. URL https://arxiv.org/abs/2306.13649.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023. URL https://arxiv.org/abs/2309.16609.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pp. 7432–7439, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4793–4801. IEEE, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Cho_On_the_Efficacy_of_Knowledge_Distillation_ICCV_2019_paper.html.

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 preprint arXiv:1803.05457, 2018. URL https://arxiv.org/abs/1803.05457.

Sayantan Dasgupta, Trevor Cohn, and Timothy Baldwin. Don't ignore the tail: Decoupling top-k probabilities for efficient language model distillation. arXiv preprint arXiv:2602.20816, 2026. URL https://arxiv.org/abs/2602.20816.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. arXiv preprint arXiv:1812.02224, 2018. URL https://arxiv.org/abs/1812.02224.

W Brier Glenn et al. Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1):1–3, 1950. URL https://doi.org/10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association, 102(477):359–378, 2007. URL https://doi.org/10.1198/016214506000001437.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In International Conference on Learning Representations, volume 2024, pp. 32694–32717, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/8ac015d409635f196f9e3e9dcfb9a94e-Abstract-Conference.html.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Miniplm: Knowledge distillation for pre-training language models. arXiv preprint arXiv:2410.17215, 2024b. URL https://arxiv.org/abs/2410.17215.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015. URL https://arxiv.org/abs/1503.02531.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In Findings of the association for computational linguistics: EMNLP 2020, pp. 4163–4174, 2020. URL https://aclanthology.org/2020.findings-emnlp.372/.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 1317–1327, 2016. URL https://aclanthology.org/D16-1139/.

Minchong Li, Feng Zhou, and Xiaohui Song. Bild: Bi-directional logits difference loss for large language model distillation. In Proceedings of the 31st International Conference on Computational Linguistics, pp. 1168–1182, 2025. URL https://aclanthology.org/2025.coling-main.78/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Proceedings of the 2018 conference on empirical methods in natural language processing, pp. 2381–2391, 2018. URL https://aclanthology.org/D18-1260/.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pp. 5191–5198, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/5963.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers), pp. 1525–1534, 2016. URL https://aclanthology.org/P16-1144/.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3962–3971. IEEE, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Park_Relational_Knowledge_Distillation_CVPR_2019_paper.html.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36: 53728–53741, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In International conference on machine learning, pp. 4334–4343. PMLR, 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. CoRR, abs/1412.6550, 2014. URL https://api.semanticscholar.org/CorpusID:2723173.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. Communications of the ACM, 64(9): 99–106, 2021. URL https://arxiv.org/abs/1907.10641.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019. URL https://arxiv.org/abs/1910.01108.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp. 4463–4473, 2019. URL https://aclanthology.org/D19-1454/.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in neural information processing systems, 33:5776–5788, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. Advances in neural information processing systems, 33:5824–5836, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Proceedings of the 57th annual meeting of the association for computational linguistics, pp. 4791–4800, 2019. URL https://aclanthology.org/P19-1472/.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11943–11952. IEEE, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Zhao_Decoupled_Knowledge_Distillation_CVPR_2022_paper.html.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. Revisiting knowledge distillation for autoregressive language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 10900–10913, 2024. URL https://aclanthology.org/2024.acl-long.587/.

A Candidate Selection and Scale Calibration

A.1 Teacher-Anchored Candidate Budget

Pool $\mathcal{K}_i$ contains the student's top 25 tokens. If needed, the teacher's highest-probability token replaces the lowest-probability student candidate. We normalize the teacher probabilities within the pool as $\rho_i(v) = p_i^{\mathrm{T}}(v)/\sum_{u\in\mathcal{K}_i} p_i^{\mathrm{T}}(u)$. The effective support is $E_i = \exp[-\sum_{v\in\mathcal{K}_i}\rho_i(v)\log\rho_i(v)]$, with sequence mean $\overline{E}_x = N_x^{-1}\sum_{i\in\Omega(x)} E_i$. These quantities determine the candidate budget:

$$d_i = \mathrm{clip}\left(\mathrm{round}\left[d_{\min} + (d_{\max} - d_{\min})\frac{E_i}{\overline{E}_x + \epsilon_{\mathrm{sel}}}\right], d_{\min}, d_{\max}\right), \tag{10}$$

where $d_{\min} = 3$ and $d_{\max} = 25$. We retain the d_i pool tokens with the highest teacher probabilities, breaking ties by vocabulary index. The teacher anchor remains in this set, and the remaining vocabulary receives conditional-distribution supervision through $\mathcal{R}_i$.

A.2 Brier Relations and Fixed Scale

For a teacher preference r, a student preference q, and $Y \sim \mathrm{Bern}(r)$, the quadratic score satisfies $\mathbb{E}[(q-Y)^2] - \mathbb{E}[(r-Y)^2] = (q-r)^2$. Thus, squared error between the two preference probabilities equals the excess expected Brier score for a fixed teacher target. When $q = \sigma(\Delta)$, its derivative is $\partial(q-r)^2/\partial\Delta = 2(q-r)q(1-q)$. The factor $q(1-q)$ reduces this derivative near saturated probabilities. Student-dependent pair weights contribute additional derivatives, so the full weighted objective requires its own analysis of scoring-rule properties.

We calibrate the Brier coefficient against a relational KL objective. Let $r^M_{i,uv} = (q^M_{i,uv}, 1 - q^M_{i,uv})$ and $\ell_i^{\mathrm{RKL}} = \sum_{(u,v)\in\mathcal{P}_i}\omega_{i,uv}D_{\mathrm{KL}}(r^{\mathrm{T}}_{i,uv}\|r^{\mathrm{S}}_{i,uv})$. At initialization θ_0, set $\mathcal{C}_0$ contains up to the first 32 valid positions from each of the first two training sequences. Selection statistics use the full sequences, and the scale is

$$\alpha = \frac{\mathrm{mean}_{i\in\mathcal{C}_0}\|\nabla_{z_i^{\mathrm{S}}}\ell_i^{\mathrm{RKL}}|_{\theta_0}\|_2}{\mathrm{mean}_{i\in\mathcal{C}_0}\|\nabla_{z_i^{\mathrm{S}}}\ell_i^{\mathrm{B}}|_{\theta_0}\|_2}. \tag{11}$$

Both mean norms must be positive and finite. Both gradients include pair-weight derivatives. Each initialization has its own fixed α, which is also used by its paired Brier run with uniform position weights.

B Task Direction and Sequence Allocation

B.1 Full Gradient and Valid Scores

The full distillation gradient contains derivatives of both the Brier errors and their weights:

$$g_i^{\mathrm{D}} = \nabla_{z_i^{\mathrm{S}}}\ell_i^{\mathrm{P}} + \alpha\sum_{(u,v)\in\mathcal{P}_i}\left[\omega_{i,uv}\nabla_{z_i^{\mathrm{S}}}B_{i,uv} + B_{i,uv}\nabla_{z_i^{\mathrm{S}}}\omega_{i,uv}\right]. \tag{12}$$

A small logit update along $-g_i^{\mathrm{D}}$ changes the observed-token loss according to

$$\ell_i^{\mathrm{CE}}(z_i^{\mathrm{S}} - \eta g_i^{\mathrm{D}}) - \ell_i^{\mathrm{CE}}(z_i^{\mathrm{S}}) = -\eta\langle g_i^{\mathrm{CE}}, g_i^{\mathrm{D}}\rangle + O(\eta^2). \tag{13}$$

The score in Section 4.2 normalizes the inner product into a direction score. The implementation clips scores to $[-1, 1]$ and includes a position in $\mathcal{C}_x$ when both gradient norms exceed $\max(10^{-7}, 16\epsilon_{\mathrm{dtype}})$ and the score is finite. Other valid prediction positions receive unit weights.

B.2 Mean-Constrained Allocation

For fixed scores and $m_x = |\mathcal{C}_x| > 0$, the position weights solve

$$\max_{w_i \geq 0,\ \sum_{i \in \mathcal{C}_x} w_i = m_x} \sum_{i \in \mathcal{C}_x} \left[w_i A_i - \frac{(w_i - 1)^2}{2\beta} \right]. \tag{14}$$

To derive the weights, introduce a multiplier λ for the sum constraint. Stationarity gives $A_i - (w_i - 1)/\beta - \lambda = 0$, so $w_i = 1 + \beta(A_i - \lambda)$. Summing over $\mathcal{C}_x$ yields $\lambda = \overline{A}_x$, where $\overline{A}_x = m_x^{-1} \sum_{i \in \mathcal{C}_x} A_i$. The objective is strictly concave for $\beta > 0$. Since $A_i, \overline{A}_x \in [-1, 1]$, the solution satisfies $w_i \geq 1 - 2\beta \geq 0$ for $0 < \beta \leq 1/2$ and is therefore the unique constrained maximizer.

The centered scores sum to zero. Substituting the weights and assigning unit weight to positions outside $\mathcal{C}_x$ gives

$$\frac{1}{N_x} \sum_{i \in \Omega(x)} w_i = 1, \qquad \frac{1}{m_x} \sum_{i \in \mathcal{C}_x} w_i A_i = \overline{A}_x + \beta \operatorname{Var}_{\mathcal{C}_x}(A). \tag{15}$$

Here, $\operatorname{Var}_{\mathcal{C}_x}(A) = m_x^{-1} \sum_{i \in \mathcal{C}_x} (A_i - \overline{A}_x)^2$. The second identity follows from $\sum_i (A_i - \overline{A}_x) A_i = \sum_i (A_i - \overline{A}_x)^2$. If $\mathcal{C}_x$ is empty, all weights equal one. These identities describe allocation for fixed logit-space scores; the effect on parameter updates also depends on model Jacobians and the optimizer.

C Experimental Protocol and Paired Results

C.1 Evaluation Records

Completed evaluations cover the 200M student at 78M, 196M, and 392M training tokens. The first budget covers eight tasks; the later budgets add StoryCloze 2016 using its public 1,871-example test set. Each mean is the arithmetic average of the displayed task scores. HellaSwag and OpenBookQA use acc_norm; ARC-C, ARC-E, LAMBADA, PIQA, Social IQA, StoryCloze 2016, and WinoGrande use acc. Rankings and ties use unrounded scores, while tables show two decimal places.

Training tokens are counted from the selected sequences and their actual lengths. Each optimizer update processes 16 sequences. A stored sequence of length L contributes $L - 1$ prediction targets. The resulting counts are 78,374,579, 195,992,525, and 391,970,932 tokens at 10,000, 25,000, and 50,000 updates, respectively. These totals count training passes only.

HS, LAMB, OBQA, SIQA, SC16, and WG denote HellaSwag, LAMBADA, OpenBookQA, Social IQA, StoryCloze 2016, and WinoGrande. the 50,000-update runs also differ in training-data prefixes. The comparisons report one run per method at each budget, so they provide descriptive comparisons rather than estimates of variation across seeds. The student has 203,437,824 parameters and the teacher is Qwen1.5-1.8B.

The student uses the Qwen2 architecture with 12 layers, hidden size 768, feed-forward size 2,112, and 12 attention heads. It has a vocabulary of 151,936 tokens and tied input and output embeddings.

The baseline data manifest specifies a corpus reconstructed from monology/pile-uncopyrighted, revision 3be90335b66f24456a5d6659d9c8d208c0357119. It uses the text field of the first 2,000,000 documents in streaming order. The training driver reads the resulting token sequences in a fixed order, without resampling or cycling. Each stored sequence contains at most 513 tokens, giving at most 512 input tokens and next-token targets after shifting.

The baseline implementation uses AdamW with betas $(0.9, 0.98)$, $\epsilon = 10^{-6}$, weight decay 0.01, and gradient clipping at 0.5. A microbatch contains one sequence, and gradients

accumulate over 16 sequences. The learning rate increases linearly to 6×10^{-4} over 512 updates, then follows a cosine schedule toward 6×10^{-5} at update 126,400. Shorter runs use the corresponding prefix of this schedule. Student parameters and losses use FP32, while GPU forward passes use BF16. Each sequence loss is averaged over its valid tokens before averaging across the 16 sequences.

The 392M-token prediction evaluation uses the same 2,048 texts and 1,000,397 valid tokens for all nine models. Cross-entropy is the mean negative log-probability of the observed token, measured in nats, and perplexity is its exponential. Teacher-student KL compares full output distributions. Next-token accuracy measures whether the highest-probability prediction matches the observed token, and confidence is the probability assigned to that prediction.

Token-level ECE uses 15 confidence bins and is computed as $\mathrm{ECE} = \sum_{b=1}^{15} (|\mathcal{B}_b|/N)|\mathrm{acc}(\mathcal{B}_b) - \mathrm{conf}(\mathcal{B}_b)|$, where N is the number of valid tokens and $\mathcal{B}_b$ is a confidence bin. Accuracy and confidence are averaged within each nonempty bin; empty bins contribute zero. Tables report ECE, accuracy, and mean confidence as percentages. Repeated evaluation reproduced the recorded cross-entropy values.

C.2 Available Short-Run Comparisons

The short-run comparisons use 512 updates with matched initialization, data order, optimizer, schedule, and evaluation data. Their evaluation set contains 192 sequences and 95,102 valid positions and was also used during development. Table 4 reports the first method minus the second; lower cross-entropy and higher accuracy favor the first method.

Table 4: Short-run paired results. Accuracy differences are in percentage points.

Comparison	Pair	CE difference	Accuracy difference
TBRD − uniform relational KL	I	−0.0776	+0.2282
TBRD − uniform relational KL	II	−0.0419	+1.3859
TBRD − direction-weighted relational KL	I	−0.0253	−0.4995
TBRD − direction-weighted relational KL	II	−0.0526	+0.6677
Uniform Brier − TBRD	Existing pair	−0.0178	+0.6267

TBRD improves both measures over uniform relational KL in the two pairs. Against direction-weighted relational KL, it lowers cross-entropy in both pairs, while the accuracy difference changes sign. These short-run results support the full method over uniform relational KL, while the Brier comparison favors uniform position weights. Their scope is the recorded initialization pairs and training duration.