ROKA: ROBUST KNOWLEDGE ALIGNMENT FOR CROSS-WIDTH HIDDEN-STATE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Distilling hidden states from a wide teacher to a narrower student requires comparing representations with unequal widths. Learned projections address this mismatch but add parameters and geometric bias, whereas relational objectives such as centered kernel alignment (CKA) avoid coordinate-wise matching altogether. We introduce RoKA, a framework for cross-width hidden-state distillation that separates two coupled design choices: the geometry of dimensional alignment and the influence of the resulting residuals on optimization. RoKA combines fixed, learned, and spectral alignment operators with a Student-t-inspired radial penalty that attenuates the gradient contribution of large operator-induced residuals. We further show that even fixed coordinates introduced by dimensional expansion can alter gradients on the retained student coordinates. Across BERT, BART, and GPT-2 teacher–student pairs, we evaluate RoKA against normal knowledge distillation (Normal KD) and CKA over five seeds. In the encoder-only complete-method comparison, student-anchored SVD achieves the highest unweighted nine-metric average, 80.13 against CKA's 78.59. On XSUM, one padding outperforms CKA on all ROUGE metrics at 1.05× versus CKA's 2.30× Normal-KD training time in a single-GPU probe. At equal hidden coefficients, robust residual optimization outperforms the corresponding quadratic objective on XSUM; on MNLI, the residual objectives perform similarly, although learned ν exceeds fixed $\nu = 1$ by 0.91 points despite approaching that bound. CKA remains strongest on CNN/DailyMail, and decoder-only differences are small, indicating that no single alignment operator dominates across settings. Overall, RoKA positions explicit alignment geometry and robust residual influence as practical, interacting design axes for cross-width hidden-state distillation.

1 INTRODUCTION

Knowledge distillation (KD) transfers behavior from a high-capacity teacher to a smaller student by augmenting supervised learning with teacher-derived targets (Hinton et al., 2015). Logit matching is broadly applicable because teacher and student predictions inhabit the same label or vocabulary space, but it exposes only the teacher's final behavior. Intermediate representations encode additional information about lexical composition, task structure, and transformations across depth. FitNets, DistilBERT, Patient KD, TinyBERT, and MiniLM demonstrate that transferring such information can improve compact students (Romero et al., 2015; Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020; Wang et al., 2020). Progressive module replacement provides a distinct compression route that does not require an additional matching loss (Xu et al., 2020).

The standard hidden-matching formulation conceals a restrictive assumption. For teacher and student states $H_T \in \mathbb{R}^{n\times d_T}$ and $H_S \in \mathbb{R}^{n\times d_S}$, a coordinate-wise distance is defined only when $d_T = d_S$. Compression commonly violates this equality. Retaining a wide student weakens compression, whereas adding a learned projector introduces capacity and geometric bias that are rarely separated from the residual loss.

Relational alignment offers a way to compare teacher and student representations with unequal hidden widths. For example, hidden-state distillation based on centered kernel alignment (CKA) compares representational relationships without requiring coordinate-wise width agreement (Dasgupta & Cohn, 2025; Kornblith et al., 2019). For linear CKA, orthogonal transformations and isotropic

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.

rescaling leave the similarity score unchanged, reducing its sensitivity to coordinate frames and overall magnitude (Kornblith et al., 2019). This is a useful inductive bias, not a failure to provide training gradients. Explicit *placement* makes a different choice: construct a student representation in the teacher-width space and penalize its coordinate residual. Figure 1 contrasts these objectives. Neither is universally preferable.

Explicit placement raises two design questions: how should comparable representations be constructed, and how should their residuals influence learning? Recent KD work separately studies optimization trade-offs, gradient disparity, limited fidelity, and imperfect teacher information (Stanton et al., 2021; Hayder et al., 2026; On et al., 2026). These studies address related aspects of projection, optimization, and residual learning, but do not systematically isolate how alignment geometry interacts with robust residual influence under width mismatch.

Our goal is therefore to systematically study cross-width hidden-state distillation by disentangling two coupled choices: the geometry used to reconcile representation widths and the influence assigned to the resulting residual. To this end, we propose **RoKA** (Robust Knowledge Alignment), which jointly analyzes these two axes within a unified framework. An operator determines what is inserted, learned, or spectrally reconstructed; a radial penalty determines how the resulting mismatch influences optimization. We assemble zero padding, one padding, learnable padding, identity-block-initialized linear expansion, and student- or teacher-anchored SVD under this formulation. The contribution is their joint analysis and evaluation, not the invention of padding, projection, SVD, or the Student-t distribution.

We evaluate RoKA across encoder-only, encoder-decoder, and decoder-only Transformer settings using BERT, BART, and GPT-2 families, with teacher and student drawn from the same backbone family in each setting. These experiments assess RoKA across diverse architectural settings rather than isolate the causal effect of architecture.

Our contributions are:

1. We formulate cross-width hidden KD through operator geometry and residual influence, and show analytically how fixed expansion coordinates can affect gradients on retained student coordinates.
2. We distinguish student- and teacher-anchored spectral reconstructions by their factors and differentiable paths, then evaluate them alongside fixed and learned expansion.
3. We provide five-seed evidence across three Transformer families, separating complete-method comparisons from within-operator residual ablations and reporting protocol-scoped training-time measurements.

2 Related Work and Motivation

Prediction and intermediate KD. Logit KD minimizes a divergence between softened teacher and student predictions (Hinton et al., 2015). Intermediate KD transfers hidden states, attention, or relation structure (Sun et al., 2019; Jiao et al., 2020; Wang et al., 2020). Cross-layer work in vision shows that layer correspondence and cross-stage aggregation are consequential design choices (Chen et al., 2021a;b); a Transformer comparison likewise finds hidden-state transfer sensitive to layer mapping (Udagawa et al., 2023). These methods establish that internal supervision can be useful, but coordinate losses require compatible widths and learned projections confound the representation objective with the alignment map.

Width-agnostic relational alignment. Relational and contrastive KD, developed largely in vision, transfer pairwise geometry or cross-sample information instead of enforcing only pointwise feature equality (Park et al., 2019; Tian et al., 2020). HSM applies CKA as width-agnostic relational supervision for language-model distillation (Dasgupta & Cohn, 2025); Kornblith et al. provide the underlying representation-similarity analysis (Kornblith et al., 2019). Explicit placement is also established: TinyBERT learns a student-to-teacher width projector (Jiao et al., 2020), and MobileBERT combines bottlenecked width reduction with layer-wise knowledge transfer (Sun et al., 2020). Residual-learning KD instead projects teacher states into a student-compatible space and learns their residual (On et al., 2026). RoKA differs by comparing fixed, learned, and batch-wise SVD operators

in teacher width and robustly weighting the operator-induced reconstruction residual. Its experimental scope is the listed baselines, not an exhaustive comparison against projection-based distillation.

Robust residual optimization. Padding introduces deterministic coordinates, learned expansion changes their dependence on the input, and low-rank reconstruction removes subspaces. These operations can create structured mismatch, which need not be random corruption or empirically heavy-tailed noise. Student-t modeling motivates robust treatment of large residuals (Lange et al., 1989), while adaptive robust objectives learn a continuous loss-shape parameter in neural optimization (Barron, 2019). RoKA does not originate either idea: it defines a Student-t-inspired radial specialization for operator-induced hidden residuals and derives its gradient in Section 4.1.

Architectural role. Encoder-only states support bidirectional discrimination, encoder-decoder states separate source encoding from autoregressive decoding, and decoder-only states directly parameterize next-token distributions. Recent autoregressive KD studies include imitation-learning trajectories, alternative divergence directions, adaptive teaching, student/on-policy targets, and score-based objectives (Lin et al., 2020; Gu et al., 2024; Zhong et al., 2024; Agarwal et al., 2024; Kim et al., 2026). These are output- or trajectory-level precedents; RoKA instead studies hidden-width alignment across architectural roles.

3 Problem Formulation

For mapped teacher and student hidden-state entries, let

$$H_T^{(l)} \in \mathbb{R}^{n\times d_T}, \qquad H_S^{(m)} \in \mathbb{R}^{n\times d_S}, \qquad d_S < d_T. \tag{1}$$

We seek g_ϕ with output width d_T. Teacher directions are used only by teacher-anchored SVD; student mode uses teacher shapes to match the retained rank. For zero-indexed hidden-state lists of lengths $L_S, L_T > 1$, including the embedding entry, uniform mapping is

$$\pi(m) = \left\lfloor \frac{m(L_T-1)}{L_S-1} \right\rfloor. \tag{2}$$

Let $r_i^{(m)}$ be a compared residual vector after the operator and the method's pooling or masking. With N_m compared vectors and layer weight a_m, the coordinate-residual objective is

$$\mathcal{L} = \lambda_{\text{task}}\mathcal{L}_{\text{task}} + \lambda_{\text{KD}}\mathcal{L}_{\text{KD}} + \lambda_H \sum_{m\in\mathcal{M}} a_m \frac{1}{N_m}\sum_{i=1}^{N_m} \mathcal{R}(r_i^{(m)}). \tag{3}$$

RoKA sets $\mathcal{R} = \mathcal{R}_\nu$; the quadratic control sets $\mathcal{R}(r) = \|r\|^2/2$. CKA instead replaces the entire hidden term with a layer-wise relational loss, and Normal KD sets $\lambda_H = 0$. Encoder-only padding compares sequence means over all positions, including padding, with $a_m = 1/|\mathcal{M}|$; SVD compares all token vectors with $a_m = 1$. Generative variants score valid tokens and sum layer losses, including both BART stacks; SVD factorization precedes masking. RoKA requires white-box teacher states during training but adds no inference-time module.

We ask whether explicit placement competes with relational CKA, how residual objectives compare at the tested coefficients, and which operators and spectral anchors are effective in each setting.

Per-axis instantiation. Encoder-only hidden-matching runs use $\lambda_{\text{task}} = 0$, $\lambda_{\text{KD}} = 1$, and $\lambda_H = 1$. Classification KD is KL at temperature 4 scaled by T^2; STS-B uses squared prediction error instead. Normal KD additionally includes the task loss. BART uses label CE plus token KD, with hidden weight 1.0 for padding and 0.005 for SVD; the reported encoder/decoder scales are both 1.0. Decoder-only uses next-token CE plus KD with $\lambda_H = 5\times 10^{-4}$.

Table 1: RoKA dimension-reconciliation operators.

Operator	Extra parameters	Teacher basis	Encoder-only reduction
Zero padding	none	no	sequence / layer mean
One padding	none	no	sequence / layer mean
Learnable padding	shared vector p	no	sequence / layer mean
Identity-block init.	unconstrained matrix W	no	sequence / layer mean
SVD student	none	no	token / layer sum
SVD teacher	none	yes	token / layer sum

4 RoKA

4.1 Student-t-Inspired Radial Penalty

For residual vector $r \in \mathbb{R}^{d_T}$, let $q = \|r\|_2^2$. We use

$$\mathcal{R}_\nu(r) = -\log\Gamma\left(\frac{\nu+1}{2}\right) + \log\Gamma\left(\frac{\nu}{2}\right) + \frac{1}{2}\log(\nu\pi) + \frac{\nu+1}{2}\log\left(1+\frac{q}{\nu}\right). \tag{4}$$

Equation 4 applies a univariate Student-t form to a vector's squared norm. It is therefore a Student-t-inspired radial penalty, not the normalized likelihood of a d_T-dimensional Student-t distribution. Its residual gradient is

$$\frac{\partial \mathcal{R}_\nu}{\partial r} = \frac{\nu+1}{\nu+q} r. \tag{5}$$

The multiplier decreases for large q, unlike MSE's constant radial multiplier. We learn one global degree of freedom,

$$\nu = \min\{\mathrm{softplus}(\theta) + \nu_{\min},\, \nu_{\max}\}, \tag{6}$$

with $\nu_{\min} = 1$, $\nu_{\max} = 50$, and initialization $\nu = 50$. A global scalar avoids adding layer-specific capacity. Our claim is adaptivity, not universal superiority over every fixed ν.

For fixed padding, split $r = [r_{\text{shared}}; r_{\text{added}}]$. The factor $(\nu+1)/(\nu + \|r_{\text{shared}}\|^2 + \|r_{\text{added}}\|^2)$ scales gradients on the shared coordinates even though the added constants cannot change. Operator and penalty therefore interact; they are not independent preprocessing and loss choices. Attenuation can also suppress useful supervision, so this derivative property does not establish a generalization advantage. As $\nu \to \infty$, Equation 4 approaches $q/2$ up to a constant, motivating the quadratic control $\frac{1}{2}\mathbb{E}[\sum_d r_d^2]$. Fixed $\nu = 50$ remains a robust penalty, not a Gaussian control. Its comparison with learned ν tests the stated initialization and bound, not all possible fixed values.

4.2 Fixed and Learned Width Reconciliation

For $h \in \mathbb{R}^{d_S}$,

$$g_0(h) = [h; \mathbf{0}], \qquad g_1(h) = [h; \mathbf{1}]. \tag{7}$$

Zero padding embeds the student in a coordinate subspace. One padding appends the same all-ones suffix to each compared vector; it is an affine offset, not an identity-matrix insertion. Learnable padding uses $g_p(h) = [h; p]$ with a shared $p \in \mathbb{R}^{d_T - d_S}$ initialized to zero. It learns a global offset rather than input-dependent coordinates, which limits its additional capacity. Table 1 summarizes the operators and their encoder-only reductions.

The experimental method called block diagonal is precisely an identity-block-initialized linear expansion:

$$g_B(h) = Wh, \qquad W_0 = [I_{d_S}; 0]. \tag{8}$$

All entries of W are trainable after initialization; no block constraint is enforced during optimization.

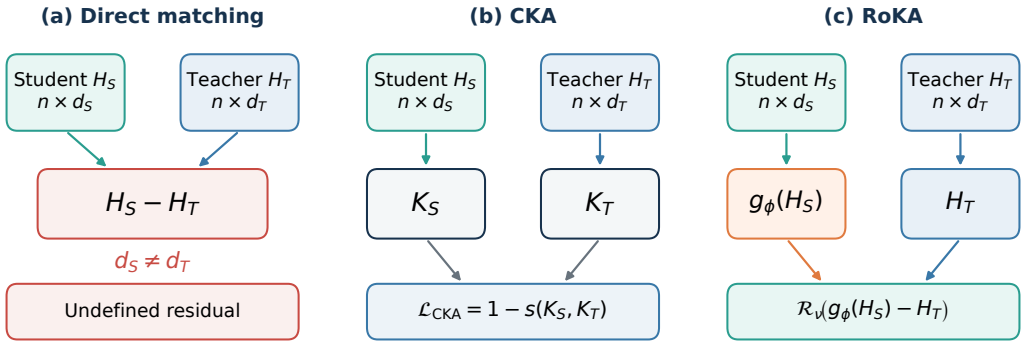


Figure 1: Comparison versus placement under hidden-width mismatch. The RoKA branch is used only during training.

4.3 SVD Student and Teacher Anchors

For $M \in \{S, T\}$, let $Z_M = \text{flat}(\text{LN}(H_M) + \epsilon\xi_M)$, where normalization acts on hidden width and ξ_M is standard Gaussian noise. We use truncated low-rank approximations and a randomized SVD routine (Eckart & Young, 1936; Halko et al., 2011); our implementation calls `torch.svd_lowrank`:

$$Z_S \simeq U_{S,k}\Sigma_{S,k}V_{S,k}^\top, \qquad Z_T \simeq U_{T,k}\Sigma_{T,k}V_{T,k}^\top. \tag{9}$$

Here $U_{M,k} \in \mathbb{R}^{n\times k}$, $\Sigma_{M,k} \in \mathbb{R}^{k\times k}$ is diagonal, and $V_{M,k} \in \mathbb{R}^{d_M\times k}$. The effective k is bounded by the requested rank and matrix dimensions. When numerical filtering leaves different ranks, student factors are padded/truncated to the teacher rank and singular magnitudes are floored at ϵ. The equations use these rank-matched factors. Student mode is

$$g_S(H_S) = U_{S,k}\Sigma_{S,k}\begin{bmatrix} V_{S,k} \\ \mathbf{0}_{(d_T-d_S)\times k}\end{bmatrix}^\top. \tag{10}$$

The right factor gains rows from d_S to d_T, equivalently columns after transposition. The reconstructed suffix is zero, so teacher energy in that suffix cannot be fitted by this operator but still affects the radial multiplier. Gradients propagate through the retained student U, Σ, and V; these factors are not detached. Teacher mode retains teacher directions while injecting student singular magnitudes:

$$g_T(H_S; H_T) = U_{T,k}\Sigma_{S,k}V_{T,k}^\top. \tag{11}$$

Figure 2 shows the reused factors. Teacher U and V are constants, so this reconstruction depends on the student through at most k singular magnitudes per layer and batch. This restricts the hidden reconstruction channel, not the entire student update: the output objective remains active. We use rank 32 for encoder-only and encoder-decoder and rank 64 for the reported decoder-only comparison. Both reconstructions are reshaped and scored against raw H_T, not Z_T; the resulting scale mismatch is part of the evaluated implementation.

Interpretation. Intuitively, U describes patterns across sample/token rows, V gives hidden-feature directions, and Σ specifies component magnitudes. Student mode retains the student's selected patterns and directions, extending their reconstruction with zero coordinates. Teacher mode in Figure 2 instead combines selected teacher patterns and directions with student magnitudes. Neither mode retains the full spectrum: k limits the number of components, not the output width d_T, and teacher mode uses only the retained teacher factors. The reconstructed teacher-width state is then compared with raw H_T by robust-t. This links the spectral operator to residual optimization without assuming that every reconstruction mismatch is harmful noise.

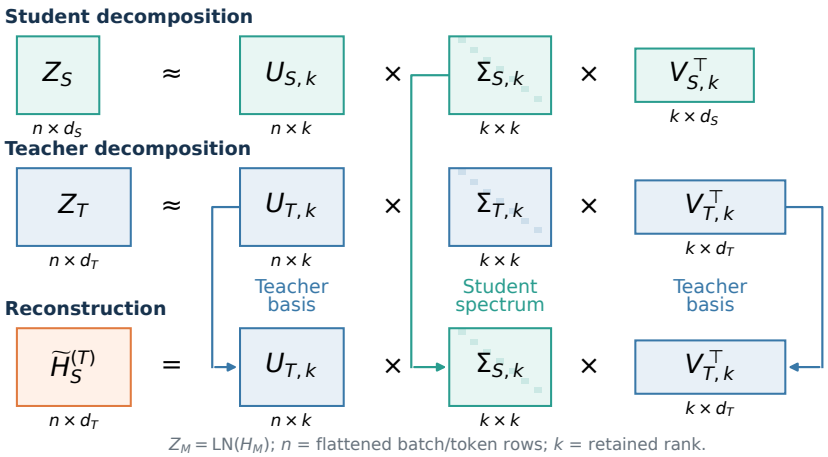


Figure 2: Teacher-mode basis replacement. Teacher $U_{T,k}, V_{T,k}^\top$ and student $\Sigma_{S,k}$ reconstruct a state of width d_T. Arrows show factor reuse; the diagram omits the small stabilization noise. Rank padding/truncation is applied when needed.

5 Experimental Setup

The cited publications establish model and dataset provenance; fine-tuning status, custom student widths, checkpoints, fixed subsets, and optimization settings below are choices of our implementation.

Encoder-only. We distill a fine-tuned BERT-base teacher into a 12-layer width-512 BERT student on CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE (Devlin et al., 2019; Wang et al., 2019). We report Matthews correlation for CoLA, Spearman correlation for STS-B, F1 for MRPC and QQP, and accuracy otherwise.

Encoder-decoder. BART-large teachers and a 12-layer encoder/12-layer decoder width-768 BART student are evaluated on XSUM and CNN/DailyMail (Lewis et al., 2020; Narayan et al., 2018; Hermann et al., 2015). We report ROUGE-1/2/Lsum.

Decoder-only. A dataset-adapted GPT-2 XL is the teacher and the pretrained GPT-2 Small (`gpt2`) checkpoint is used directly as student on fixed WikiText-103 and OpenWebText subsets (Radford et al., 2019; Merity et al., 2017; Gokaslan & Cohen, 2019). We report perplexity and next-token accuracy; this is a controlled extension, not a frontier-scale LLM claim.

Compression profiles. Encoder-only uses BERT-base (768, 12 layers) to BERT (512, 12 layers); encoder-decoder uses BART-large (1024, 12/12 layers) to BART (768, 12/12 layers); decoder-only uses GPT-2 XL (1600, 48 layers) to the pretrained GPT-2 Small checkpoint (768, 12 layers). Their student/teacher width ratios are 0.67, 0.75, and 0.48, and depth ratios 1.00, 1.00, and 0.25.

Baselines and selection. Normal KD has no hidden loss; CKA is the principal width-agnostic baseline. For each configuration we train seeds 1–5 and select one validation checkpoint per run; MNLI maximizes mean matched/mismatched accuracy, and summarization maximizes ROUGE-Lsum. We report all metrics from that checkpoint and average the five selected checkpoints. Selection and reporting on the same development sets can introduce optimism, so these are validation comparisons rather than independent test estimates. Appendix A specifies objective and optimization details.

Comparison scope. Tables 2–4 compare complete method configurations. Encoder-only padding and SVD differ in pooling and layer reduction (Table 1), while CKA excludes padded tokens and those methods do not. Normal KD also retains a task loss omitted by encoder-only hidden-matching

Table 2: Encoder-only complete-method comparison (%). Teacher scores use frozen checkpoints; student scores average five seeds. Pooling, masking, and objective differences are specified in Experimental Setup. Avg. is the unweighted mean of nine metrics.

Method	MRPC	CoLA	RTE	SST-2	QNLI	STS-B	QQP	MNLI-m	MNLI-mm	Avg.
BERT-base teacher	90.61	60.33	65.83	92.09	91.69	88.36	88.38	84.43	84.47	82.91
Normal KD	85.14	39.60	59.86	90.25	85.97	86.75	86.86	80.11	80.08	77.18
CKA	88.07	37.12	60.07	**90.94**	89.85	**87.52**	87.95	82.65	83.12	78.59
Zero padding	86.07	40.98	58.99	**90.94**	89.27	86.39	87.97	82.00	82.51	78.35
One padding	85.15	38.32	58.06	90.69	87.87	86.42	87.43	80.98	81.48	77.38
Learnable padding	86.19	40.47	59.50	90.89	89.20	86.39	88.00	82.16	82.52	78.37
Identity-block init.	86.37	40.07	59.42	90.78	89.44	86.67	**88.19**	82.43	83.13	78.50
SVD student	**90.04**	**46.50**	**61.15**	**90.94**	**90.38**	87.07	88.06	**83.26**	**83.79**	**80.13**
SVD teacher	81.94	21.93	53.74	88.99	83.90	50.54	86.48	80.02	80.16	69.75

Table 3: Summarization ROUGE (%). SVD uses tuned $\lambda_H = 0.005$; padding uses 1.0. Training time is reported separately in Table 6.

	XSUM			CNN/DM		
Method	R-1	R-2	R-L	R-1	R-2	R-L
Normal	36.37	15.27	29.12	43.88	20.88	40.24
CKA	43.09	20.41	35.33	**44.20**	**21.14**	**40.36**
One pad.	**43.50**	**20.71**	**35.68**	44.10	21.08	40.14
Zero pad.	43.14	20.10	35.08	44.06	21.06	39.76
Learn. pad.	42.94	19.90	34.78	44.04	21.03	39.75
Block init.	42.28	19.14	33.79	43.93	20.94	39.48
SVD-S ($\lambda_H = .005$)	42.02	19.37	34.14	43.66	20.66	39.97
SVD-T ($\lambda_H = .005$)	41.68	18.98	33.56	43.36	20.40	39.60

methods. BART SVD uses $\lambda_H = 0.005$ rather than padding's 1.0. Thus these tables do not isolate operator geometry or the effect of adding hidden supervision. In contrast, Table 5 holds the operator and reduction fixed within each dataset; its scope is the tested residual objectives at equal hidden coefficient, not equal gradient strength.

Timing measurement protocol. We report wall-clock time relative to Normal KD, measured by single-GPU probes for MNLI and XSUM and by full-run elapsed time for WikiText-103 (Appendix A). Table 6 reports these within-setting ratios as point estimates without repeated-timing dispersion; absolute times are not comparable across settings.

6 Results

6.1 Encoder-Only

Table 2 favors student-anchored SVD on the average and over teacher anchoring on every metric in all five paired seeds. The CoLA and STS-B gaps show that retaining student singular magnitudes does not guarantee an effective teacher-anchored reconstruction; they do not directly measure destruction of task geometry. CKA leads on STS-B and the identity-block projection on QQP, so the average does not imply a universal task-level winner.

6.2 Encoder-Decoder

Table 3 favors one padding over CKA on all XSUM metrics, whereas CNN/DailyMail favors CKA. SVD remains in the comparison at its selected coefficient; the table does not establish that it would rank similarly at padding's coefficient. Together with the measured XSUM ratios in Table 6, this is the clearest example of competitive placement with less observed training-time overhead than CKA.

6.3 Decoder-Only

In Table 4, teacher-anchored SVD has the lowest mean WikiText-103 perplexity, while CKA is worse than Normal KD on both datasets. Differences among non-CKA methods are small and

Table 4: Decoder-only results. Lower PPL and higher accuracy are better. Boldface uses unrounded means; equal displayed values need not be exact ties.

Method	WikiText-103 PPL	WikiText-103 Acc.(%)	OpenWebText PPL	OpenWebText Acc.(%)
Normal	27.109	39.934	23.077	40.634
CKA	27.787	39.700	23.463	40.450
One pad.	27.108	**39.953**	**23.068**	40.639
Zero pad.	27.102	39.934	23.069	**40.646**
Learn. pad.	27.097	39.941	23.070	40.645
Block init.	27.104	39.938	23.071	40.642
SVD student	27.088	39.946	23.074	40.634
SVD teacher	**27.080**	39.933	23.068	40.634

Table 5: Within-operator residual ablations at hidden weight 1.0 (five-seed mean ± sample standard deviation). Metrics are ROUGE-1/2/Lsum for XSUM and matched/mismatched accuracy for MNLI. Hidden is the unweighted hidden loss.

Axis	Residual	Metric 1	Metric 2	Metric 3	Hidden
XSUM/one pad.	MSE	42.142±.074	18.970±.069	33.906±.055	3942
	fixed $\nu = 50$	43.432±.092	20.603±.125	35.568±.129	1317
	learned ν	**43.460**±.068	**20.694**±.031	**35.663**±.040	1203
MNLI/SVD-S	MSE	83.502±.101	83.474±.090	–	979
	fixed $\nu = 50$	**83.533**±.175	83.476±.108	–	436
	learned ν	83.262±.116	**83.786**±.146	–	74

perplexity winners need not maximize token accuracy. The mean ordering supports comparison of the evaluated modes, not a statistically established decoder-only advantage.

7 Analysis and Ablations

Residual objective and learned ν. Table 5 holds the operator, reduction, and clipping policy fixed within each dataset. On XSUM, both robust variants exceed MSE. On MNLI, fixed $\nu = 50$ reaches a two-metric average of 83.505 ± 0.115, only 0.019 points below learned ν at 83.524 ± 0.067; this is not an equivalence test or evidence of universal superiority. The learned degree of freedom decreases to $\nu \approx 41.86$ on XSUM and approaches the lower bound on MNLI, yet fixing $\nu = 1$ from initialization lowers MNLI accuracy by 0.91 points (Table 7, p = 0.00025). CoLA fixed-versus-learned differences are not significant at the examined checkpoints ($p = 0.35$ at epoch 9; $p = 0.91$ at epoch 10). The XSUM one-padding entries in Tables 3 and 5 are separate main and ablation run sets; they are not pooled into one estimate. Appendix A records ablation optimization settings.

Scale caveat. The hidden-loss magnitudes in Table 5 differ substantially despite equal coefficients. Scalar loss values do not measure gradient norms, and matching their averages would not equalize optimization. The results therefore establish performance at the reported coefficients, not a scale-independent advantage of robust loss or superiority over a fully tuned MSE baseline.

Operator and spectral anchor. Tables 2–4 do not show a monotonic benefit from operator complexity. Learned expansion does not consistently beat fixed padding, and the mean SVD-mode ordering differs between encoder-only and decoder-only experiments. Because tasks and compression profiles also differ, architecture is a possible explanation rather than an isolated cause.

Training-time analysis. Table 6 shows that padding remains close to Normal KD in every measured setting. CKA is cheaper than rank-32 SVD in the heavily padded MNLI probe but substantially more expensive on XSUM and the packed WikiText-103 run. This reversal is consistent with the implementation: CKA excludes invalid tokens before forming $X^\top Y$, $X^\top X$, and $Y^\top Y$, whereas SVD decomposes full padded hidden matrices with a fixed retained rank. Roughly, their method-specific terms scale as $O(N_{\text{valid}}(d_S d_T + d_S^2 + d_T^2))$ and rank-dependent $O(Nk(d_S + d_T))$, respectively. Both SVD modes compute the same decompositions, so their small probe-time difference is not

Table 6: Measured training time relative to Normal KD within each setting. MNLI and XSUM use rank-32 SVD probes; the rank-64 SVD values apply only to the five-epoch WikiText-103 runs. Dashes denote unavailable measurements.

Method	MNLI	XSUM	WikiText-103
Normal KD	1.00	1.00	1.00
CKA	1.23	2.30	1.83
Zero padding	1.03	1.05	1.03
One padding	1.03	1.05	1.04
Learnable padding	1.03	1.06	1.03
Identity-block init.	1.03	1.07	1.04
SVD student ($k = 32$)	1.56	1.48	–
SVD teacher ($k = 32$)	1.54	1.43	–
SVD student ($k = 64$)	–	–	1.12
SVD teacher ($k = 64$)	–	–	1.11

interpreted. Token density and implementation details vary together, making this an explanatory consistency check rather than a causal test.

8 Discussion and Limitations

Among the evaluated configurations, student-anchored SVD raises the encoder-only average from CKA's 78.59 to 80.13, while one padding improves XSUM ROUGE-Lsum from 35.33 to 35.68 and also improves ROUGE-1/2. These are concrete examples of effective explicit placement, alongside CNN/DailyMail's preference for CKA. They retain the complete-method qualifications in Experimental Setup. The derivative analysis explains why placement is not neutral: inserted coordinates can influence gradients on retained coordinates, but these performance results do not isolate that mechanism's causal contribution.

Anchor-bandwidth hypothesis. Teacher anchoring fixes the reconstruction directions, whereas student anchoring differentiates through student directions and magnitudes. This local distinction suggests different constraints on hidden supervision. It does not imply that the student's own singular vectors or parameters are frozen in teacher mode. The lower encoder-only teacher-mode scores are consistent with a restrictive reconstruction channel, but causal attribution would require controlled gradient or representation measurements; the small decoder-only differences do not establish the opposite mechanism.

The study requires white-box teacher states and evaluates one student shape per role within the same backbone family. Decoder-only uses fixed subsets. Pooling, masking, objective, and weight differences limit attribution in the main tables, while validation-based checkpoint selection limits generalization claims. SVD scores normalized reconstructions against raw teacher targets; its numerical fallbacks are uninstrumented. Timing probes lack repetitions and DDP communication, and decoder-only timing records lack per-run dispersion. Together with small decoder-only performance margins, these limitations preclude claims of a universal operator or uniformly superior learned ν.

9 Conclusion

RoKA combines explicit width reconciliation with robust residual optimization. Student-anchored SVD achieves the strongest encoder-only average, and one padding improves all XSUM ROUGE metrics over CKA while measuring $1.05\times$ versus CKA's $2.30\times$ Normal-KD time in the XSUM probe. CNN/DailyMail and decoder-only results show why this is not a universal operator ranking. On MNLI, learned ν performs comparably to fixed $\nu = 50$ but outperforms fixed $\nu = 1$, the value it approaches, indicating that in these runs the trajectory of ν, rather than its endpoint alone, matters. The contribution is a practical design framework with setting-specific empirical support, not a claim that every robust operator or adaptive parameter is superior.

AI USE STATEMENT

Generative AI tools were used to assist with English-language editing, improving the clarity and presentation of the manuscript, LaTeX formatting, retrieval and discovery of relevant literature, and drafting portions of the manuscript. The research ideas, methodological design, experimental implementation, data analysis, interpretation of results, and conclusions were developed and verified by the authors. The authors reviewed all AI-assisted material and take full responsibility for all content in the manuscript.

REPRODUCIBILITY STATEMENT

The appendix specifies objective, optimization, and implementation details together with the result-selection rule and paired analyses. Encoder-only and encoder-decoder use per-seed records; decoder-only statistics use the supplied five-seed summaries, as stated in the result-provenance paragraph.

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*, 2024.

Jonathan T. Barron. A general and adaptive robust loss function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7028–7036, 2021a.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021b.

Sayantan Dasgupta and Trevor Cohn. Improving language model distillation through hidden state matching. In *International Conference on Learning Representations*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `https://skylion007.github.io/OpenWebTextCorpus/`, 2019.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations*, 2024.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

Zeeshan Hayder, Ali Cheraghian, Lars Petersson, Mehrtash Harandi, and Richard Hartley. Dto-kd: Dynamic trade-off optimization for effective knowledge distillation. In *International Conference on Learning Representations*, 2026.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1693–1701, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

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.

Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-chul Moon. Distillation of large language models via concrete score matching. In *International Conference on Learning Representations*, 2026.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 3519–3529, 2019.

Kenneth L. Lange, Roderick J. A. Little, and Jeremy M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020.

Alexander Lin, Jeremy Wohlwend, Howard Chen, and Tao Lei. Autoregressive knowledge distillation through imitation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6121–6133, 2020.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, 2018.

Thinh On, Hengzhi Pei, Leonard Lausen, and George Karypis. Knowledge distillation for large language models through residual learning. In *International Conference on Learning Representations*, 2026.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015.

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.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In *Advances in Neural Information Processing Systems*, volume 34, pp. 6906–6919, 2021.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4323–4332, 2019.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. MobileBERT: A compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2158–2170, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

Takuma Udagawa, Aashka Trivedi, Michele Merler, and Bishwaranjan Bhattacharjee. A comparative analysis of task-agnostic distillation methods for compressing transformer language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 20–31, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788, 2020.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. BERT-of-theseus: Compressing BERT by progressive module replacing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 7859–7869, 2020.

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.

A Reproducibility Details

Optimization and synchronization. Teacher parameters are frozen throughout distillation, while the student, learned ν, and learned expansion parameters are optimized jointly. All optimizer gradients are reset at every step. In distributed runs, ν is synchronized across data-parallel ranks, and gradients for learnable padding and identity-block expansion are averaged across ranks before each update, so that nominally identical seeds do not follow rank-dependent objectives. We refer to runs with these controls as the corrected implementation; results produced before these controls, including runs with inconsistent teacher-mode weights, are excluded.

Common settings. Comparisons use seeds 1–5 and ten epochs. Encoder-only uses batch size 16 and gradient clipping 1.0 in the corrected robust-loss runs. Encoder-decoder uses batch size 16 and eight gradient-accumulation steps. The three XSUM and three main-table MNLI residual-ablation conditions use hidden weight 1.0; clipping is disabled for XSUM and set to 1.0 for MNLI. The additional fixed-$\nu = 1$ MNLI analysis uses the same weight and clipping as the other MNLI conditions. Learned ν starts at 50 and is constrained to $[1, 50]$. The per-axis objectives and main-table differences are specified in Sections 3 and 5.

Encoder-only optimization. We use AdamW with an initial student learning rate of 5×10^{-5} and weight decay 0.01, with linear warmup over 10% of the training steps followed by linear decay. The initial learning rates for ν, learnable padding, and identity-block expansion are 10^{-3}, 5×10^{-6}, and 10^{-5}, respectively. Inputs are padded or truncated to 512 tokens.

Encoder-decoder optimization and generation. We use AdamW with a constant student learning rate of 10^{-4} and weight decay 5×10^{-4}. The learning rates for ν and learned expansion parameters are 10^{-3} and 10^{-5}, respectively. Source and target tokenization lengths are capped at 1024 and 128, and the KD temperature is 1.0. Generation uses six beams with minimum/maximum lengths 11/62 for XSUM and four beams with lengths 56/142 for CNN/DailyMail. Both settings prohibit repeated trigrams.

Decoder-only optimization and subsets. WikiText-103 and OpenWebText use the same optimization and subset-selection settings, with a teacher checkpoint trained on the respective dataset. The teacher is GPT-2 XL and the student is pretrained GPT-2 Small. AdamW uses a constant base learning rate of 10^{-4} with no scheduler and weight decay 5×10^{-4}. The microbatch size is 8 with two gradient-accumulation steps, giving an effective batch size of 16, and the input block length is 512 tokens. For each dataset, we remove empty-text entries from its training and evaluation pools, shuffle each with seed 1, and retain the first 20,000 training entries and 2,000 evaluation entries. These counts refer to dataset entries selected before token blocking. Every training and evaluation batch within the subsets is used, without a batch-count cap.

Measurement environment and protocol. The MNLI and XSUM probes each use one NVIDIA GeForce RTX 5090, with FP32 model parameters and no mixed-precision autocasting; CKA Gram products use FP64. MNLI/BERT uses five warmup and 20 timed training steps; XSUM/BART uses one warmup and two timed optimizer updates, each containing eight microbatches. Batches are resident on the GPU, and the probes include teacher and student forward passes, the hidden objective, backward, clipping, and the optimizer update, but exclude data loading, transfer, evaluation, and scheduler time. The WikiText-103 value is the recorded elapsed time of a full five-epoch run on NVIDIA GB10 with FP32 model parameters, no BF16/FP16 autocasting, and no gradient checkpointing.

Relational baseline implementation. For centered, valid-token matrices X, Y, the implemented CKA loss is

$$\mathcal{L}_{\mathrm{CKA}} = 1 - \frac{\|X^\top Y\|_F}{\sqrt{(\|X^\top X\|_F + \epsilon)(\|Y^\top Y\|_F + \epsilon)}}. \tag{12}$$

Ignoring stabilization, its similarity is the square root of conventional linear CKA, not its squared form. It shares the corresponding invariances but has a different gradient scaling; the reported baseline refers to this implementation. Gram products are evaluated in float64. No claim that CKA must materialize an $n \times n$ matrix follows from its relational interpretation.

Numerical stability. If low-rank SVD fails or returns non-finite factors, the implementation uses identity-based fallback factors; if the reconstruction is non-finite, it substitutes teacher features, yielding zero alignment residual for that comparison. These fallback events were not counted, so their frequency and impact are unknown.

B Additional Analyses

The encoder-only statistic averages nine task metrics per seed and inherits the complete-method qualification in Section 5. Paired t-tests are descriptive at $n = 5$, with no multiple-comparison adjustment, and are not sole evidence for a claim. The learned-ν versus fixed-$\nu = 1$ MNLI comparison uses the same per-seed joint checkpoint rule; fixed 1 selects epoch 10 in all seeds. Decoder-only results report means without seed-level dispersion, so their lowest mean perplexity is not treated as a statistically established improvement.

Table 7: Paired five-seed analyses for selected comparisons.

Axis	Comparison	Method A	Method B	Paired p
Encoder-only Avg.	SVD-S vs CKA	80.132 ± 0.139	78.590 ± 0.263	0.00053
XSUM R-1	One pad. vs CKA	43.505 ± 0.082	43.094 ± 0.169	0.0157
XSUM R-2	One pad. vs CKA	20.705 ± 0.032	20.410 ± 0.101	0.0063
XSUM R-L	One pad. vs CKA	35.678 ± 0.039	35.327 ± 0.153	0.0100
MNLI Avg.	Learned ν vs fixed 1	83.524 ± 0.067	82.614 ± 0.134	0.00025

Fixed-endpoint control. The fixed-$\nu = 1$ control tests whether learning from 50 toward the lower bound can be replaced by fixing that bound from initialization. Although the selected learned values approach $\nu = 1$, mean MNLI accuracy is 83.524% for learned ν and 82.614% for fixed 1 (Table 7), a difference of 0.910 percentage points. In these runs, a similar final degree of freedom therefore does not make the two training procedures interchangeable. Fixed $\nu = 50$ achieves 83.505%, so this control does not establish superiority over all fixed values.

Result provenance. Encoder-only and encoder-decoder statistics are computed from per-seed records using the declared checkpoint rule; decoder-only uses supplied five-seed summaries. Hidden-loss magnitudes and learned-ν trajectories use the corresponding ablation records. Cross-metric averages are descriptive summaries, not a common task-independent measurement scale.