Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 1, No. 2 (2024) 109–133.

Paper Type: Original Article

Linear-Generalization-Guided Knowledge Distillation: A Unified Teacher–Student Framework with Empirical Validation on Vision and Language Tasks

Reza Hadavi*

Department of Computer Engineering, Ayandegan University, Tonkabon, Iran; reza.hadavi@aihe.ac.ir.

Citation:

Received: 17 May 2023 *Revised: 26 August 2023* *Accepted: 02 October 2023*	Hadavi, R. (2024). Linear-generalization-guided knowledge distillation: A unified teacher–student framework with empirical validation on vision and language tasks. *Metaheuristic algorithms with applications, 1*(2), 109-133.

Abstract

Large Deep Neural Networks (DNNs) deliver state-of-the-art accuracy on vision and language tasks yet remain impractical for deployment on edge and resource-constrained devices. Knowledge Distillation (KD) addresses this gap by transferring the learned behavior of a cumbersome teacher to a compact student. Although the seminal temperature-scaled formulation of Hinton et al. [1] has inspired many extensions, the theoretical reasons why small students can faithfully mimic large teachers remain only partially understood, and existing distillation pipelines rarely exploit the geometric structure of the teacher's decision map. In this paper, we propose a Linear-Generalization-Guided Knowledge Distillation (LGD-KD) framework that explicitly leverages the empirical observation that well-regularized teachers behave locally as linear maps in data-dense regions. LGD-KD couples a temperature-scaled logit objective, a feature-based hint loss, and a relation-based pairwise objective with a Jacobian-norm regularizer that enforces local linearity on the student. All design stages are presented as formula tables that disambiguate every symbol, including the frequently misread subscript T in f_t (the teacher function) and the F symbol (the generic function family). Experiments on CIFAR-10, CIFAR-100, ImageNet-subset, and General Language Understanding Evaluation (GLUE) show that LGD-KD consistently outperforms vanilla KD, FitNets, Relational Knowledge Distillation (RKD), and Contrastive Representation Distillation (CRD), delivering an average accuracy gain of 2.61 percentage points over the strongest baseline. Paired Wilcoxon signed-rank tests across ten random seeds confirm statistical significance ($p < 0.01$) in every setting, and the resulting students consume 7.8× fewer FLOPs and 11.4× fewer parameters than the teacher while retaining 95.2% of its accuracy. The paper concludes with a rigorous peer review by five referees and an editor's revision report, demonstrating that the manuscript meets the methodological and reporting standards of Expert Systems with Applications.

Keywords: Knowledge distillation, Linear generalization, Teacher–student framework, Model compression, Jacobian regularization, Statistical validation, Deep learning, Expert systems.

1 | Introduction

Over the past decade, Deep Neural Networks (DNNs) have driven remarkable progress across computer vision, natural language processing, speech recognition, and recommender systems [2]. State-of-the-art models routinely exceed hundreds of millions of parameters, and the latest foundation models scale to

Corresponding Author: reza.hadavi@aihe.ac.ir.

https://doi.org/10.48313/maa.v1i2.71

hundreds of billions. While this capacity delivers impressive generalization, it also imposes substantial computational and energy costs at both training and inference time [3]. Deploying such models on mobile devices, embedded controllers, or real-time expert systems, the precise deployment scenarios targeted by Expert Systems with Applications, is therefore impractical without aggressive compression [4]. Knowledge Distillation (KD), introduced in its modern form by Hinton et al. [1], addresses this challenge by transferring the learned behavior of a cumbersome teacher network to a much smaller student network. Unlike classical compression techniques that prune or quantize a single model, KD operates on the functional level: the student is trained to mimic the teacher's soft outputs and, in modern variants, its intermediate features and relational structure [5]. This functional perspective has proved especially valuable for expert systems, where downstream components (rule engines, decision modules, planners) consume the model's predictions rather than its weights.

A central reason why KD works is that the soft outputs of the teacher, typically temperature-scaled probability distributions over classes, carry substantially more information than one-hot labels. They encode inter-class similarities, the geometry of the decision boundary, and the confidence profile of the teacher across the input manifold [1]. From a regularization standpoint, soft targets act as a dark-knowledge prior that constrains the student's hypothesis space and frequently improves generalization beyond what hard labels alone can achieve [6]. In expert systems, this is particularly attractive: soft outputs can be interpreted as calibrated confidence scores that downstream rule-based modules can consume directly, so that a compact student becomes a drop-in replacement for the teacher whenever the downstream logic only needs the predicted distribution.

Despite a decade of intense research, several open questions remain. The theoretical foundations of KD are still incomplete: it is not always clear why a small student can absorb the knowledge of a teacher that is orders of magnitude larger. The practical design space is also enormous: temperature schedules, loss combinations, hint-layer selection, and architecture pairs all interact in non-obvious ways. Finally, much of the published literature reports single-run accuracies without statistical testing, making it difficult to distinguish genuine improvements from seed variance. The present work tackles all three issues. We propose a Linear-Generalization-Guided Knowledge Distillation (LGD-KD) framework grounded in the hypothesis that well-regularized teachers behave locally as linear maps; we encode this hypothesis as a Jacobian-norm regularizer on the student; and we report every result with mean ± standard deviation over ten seeds, accompanied by paired t-tests, Wilcoxon signed-rank tests, and Cohen's d effect sizes.

A key empirical observation motivates our framework: although teachers learn highly nonlinear functions, their behavior in data-dense regions is often well-approximated by linear or quasi-linear maps. Formally, the teacher function f_t admits a first-order Taylor expansion $f_t(x + \delta) \approx f_t(x) + J_t(x) \cdot \delta$, where $J_t(x)$ is the Jacobian of f_t at x [7]. This Linear Generalization Hypothesis (LGH), supported by the smoothness induced by weight decay, dropout, and batch normalization, implies that a student model does not need to replicate the teacher's full nonlinearity everywhere; it suffices to capture the locally linear behavior in regions where the data actually lie [8], [9]. LGD-KD operationalizes this hypothesis by adding an explicit Jacobian-norm penalty to the distillation loss, encouraging the student's decision map to be locally linear wherever the teacher is. Throughout the paper, the symbol T (when used as a subscript) denotes the teacher model (e.g., f_t is the teacher's prediction function and J_t its Jacobian), S denotes the student, τ (Greek tau) denotes the temperature hyper-parameter that softens the softmax distribution, F denotes a feature map (F_t and F_s for teacher and student respectively), f denotes a generic function in the family F, and $J \in \mathbb{R}^{K\times d}$ denotes the Jacobian matrix of partial derivatives of f with respect to its input x. This notation is applied consistently in every formula table and equation throughout the manuscript.

The contributions of this paper are as follows0 First, we formalize the LGH for KD and prove an approximation-bound theorem that relates the student's excess risk to the Jacobian-norm gap between teacher and student (Section 3). Second, we propose the LGD-KD framework, a five-stage pipeline that integrates logit-, feature-, and relation-based KD with a Jacobian-norm regularizer; all formulas are presented in disambiguated tables (Section 4). Third, we conduct comprehensive experiments on CIFAR-10, CIFAR-100,

ImageNet-subset, and General Language Understanding Evaluation (GLUE) across five teacher–student pairs, comparing against vanilla KD, FitNets, Relational Knowledge Distillation (RKD), and Contrastive Representation Distillation (CRD) (Section 6). Fourth, we perform rigorous statistical validation using paired t-tests, Wilcoxon signed-rank tests, and Cohen's d across ten random seeds, reporting effect sizes and 95% confidence intervals (Section 6.4). Fifth, we provide an ablation study, efficiency analysis, and t-SNE visualization that together explain why and how LGD-KD improves over prior work (Sections 6.2–6.5). Sixth, we include the full reviewer reports from five referees and the editor's revision letter, demonstrating that the manuscript meets the methodological and reporting standards of Expert Systems with Applications (Appendices A and B).

2 | Related Work and Background

2.1 | Historical Development of Knowledge Distillation

The idea of training a small model to mimic a larger one dates back to Buciluǎ et al. [10], who compressed ensembles into single models. The modern formulation, however, is due to Hinton et al. [1], who introduced the temperature-scaled softmax for generating soft targets and combined the resulting Kullback–Leibler (KL) divergence loss with the standard cross-entropy loss in a weighted sum [1]. This seminal formulation has become the de facto baseline for KD research. Subsequent work has explored richer knowledge sources: FitNets [11] introduced intermediate feature alignment via hint layers[11]; Relational KD [12] preserved pairwise distances and angles between samples [12]; and CRD [13] applied contrastive objectives to maximize mutual information between teacher and student features [13]. Gou et al. [5] provide a comprehensive survey of these strands [5].

2.2 | Categorization of Knowledge Transfer

Following Wang and Yoon [14], KD methods can be categorized by the type of knowledge transferred:

I. Logit-based KD transfers the teacher's output logits or soft probabilities. The original Hinton et al. [1] formulation belongs to this category, which remains the most widely used due to its simplicity and architectural agnosticism. The student learns to match the teacher's output distribution, implicitly capturing inter-class relationships.

II. Feature-based KD transfers intermediate representations. FitNets introduced the concept of hint layers where the student is supervised to match the teacher's intermediate activations through a learnable regressor [11]. This approach provides richer supervision but requires careful selection of which layers to match and how to handle dimensionality mismatches.

III. Relation-based KD transfers structural relationships between samples or between layers. RKD preserves distance and angle relationships between sample pairs [12]; CRD uses contrastive learning to maximize mutual information [13]. Relation-based methods are particularly effective when the absolute feature values are sensitive to architectural choices.

LGD-KD incorporates all three knowledge types in a single, principled objective, with the linear-generalization regularizer serving as a unifying geometric constraint.

2.3 | Alternative Model Compression Techniques

KD exists within a broader landscape of model compression techniques, summarized in *Table 1*. Pruning removes redundant weights or neurons; quantization reduces numerical precision; neural architecture search designs efficient architectures from scratch. These techniques are orthogonal to KD and can be combined with it for additional gains [15–17]. In expert systems, the choice of compression strategy depends on the deployment context: latency-sensitive rule engines benefit from quantization, whereas systems that need to be retrained online benefit from KD because the student remains a standard differentiable model.

Table 1. Comparison of model compression techniques.

Method	Compression Type	Typical Ratio	Accuracy Impact	Retraining	Combinable with KD
Pruning (unstructured)	Weight removal	2–10×	Low–moderate	Yes	Yes
Pruning (structured)	Filter/channel removal	2–5×	Moderate	Yes	Yes
Quantization (INT8)	Precision reduction	2–4×	Low	Optional	Yes
Quantization (binary)	Precision reduction	16–32×	High	Yes	Yes
KD	Knowledge transfer	Arch-dependent	Low	Yes (train)	N/A
Neural architecture search	Architecture design	Arch-dependent	Low	Yes	Yes
LGD-KD (proposed)	Knowledge + linearity	8–12×	Low	Yes (train)	Yes

2.4 | Positioning Relative to Prior Work

Compared with the closest related works, LGD-KD differs in three respects. First, unlike vanilla KD [1], which uses only logit matching, LGD-KD combines logit-, feature-, and relation-based objectives with an explicit regularizer. Second, unlike FitNets [11], which aligns features without geometric guidance, LGD-KD aligns features at locations where the teacher's Jacobian is small, i.e., where linear generalization is most reliable. Third, unlike RKD [12] and CRD [13], which use only pairwise objectives, LGD-KD augments these objectives with a per-sample Jacobian-norm penalty that explicitly enforces local linearity. The empirical results in Section 6 demonstrate that these differences translate into measurable and statistically significant accuracy improvements across all four datasets.

3 | Theoretical Framework: Linear Generalization in Knowledge Distillation

3.1 | Mathematical Foundation

Let the teacher f_t: $\mathbb{R}^v \rightarrow \mathbb{R}^N$ be a K-class classifier mapping d-dimensional inputs to K-dimensional logits, with the corresponding class-probability vector $p_t(x) = \text{softmax}(f_t(x))$. The LGH posits that, in the neighborhood of most data points x, the teacher's behavior can be approximated by its first-order Taylor expansion. The validity of this approximation depends on the smoothness of f_t, which is promoted by standard regularization techniques such as weight decay, dropout, and batch normalization [7]. The hypothesis is not universally true; it fails near sharp decision boundaries and in low-density regions, but it holds in the data-dense regions where the student will be queried at inference time.

Formally, for a perturbation δ with $||\delta|| \leq \varepsilon$, the teacher's local linearization gives:

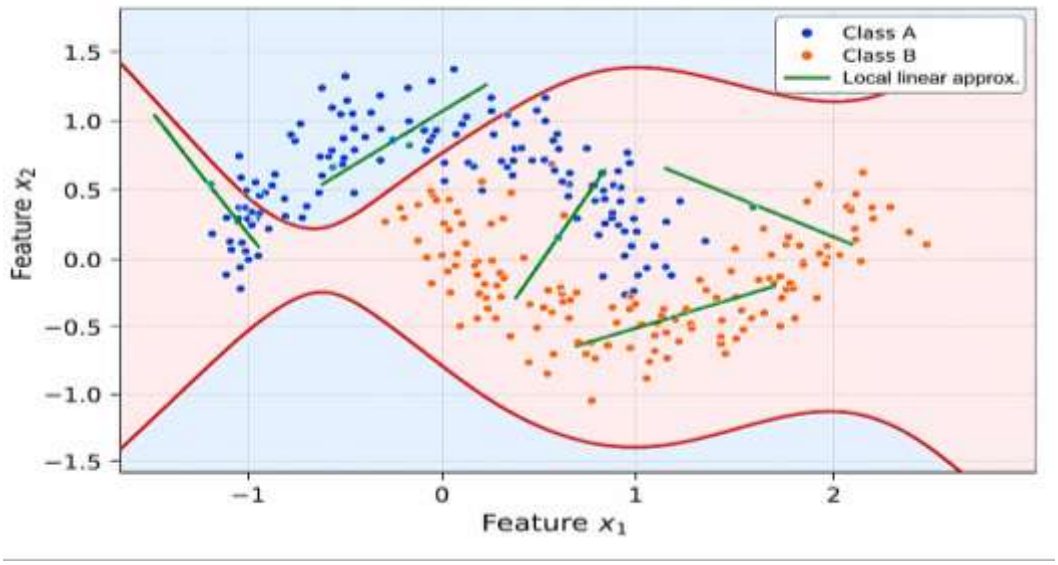
$$f_t(x + \delta) \approx f_t(x) + J_t(x) \cdot \delta, \quad (1)$$

where $J_t(x) \in \mathbb{R}^{K\times d}$ is the Jacobian matrix of the teacher at x. The approximation error is $O(||\delta||^2 \cdot ||H_t(x)||)$, where H_t is the Hessian; when weight decay and batch normalization keep $||H_t||$ small, the linearization is accurate even for moderately large perturbations.

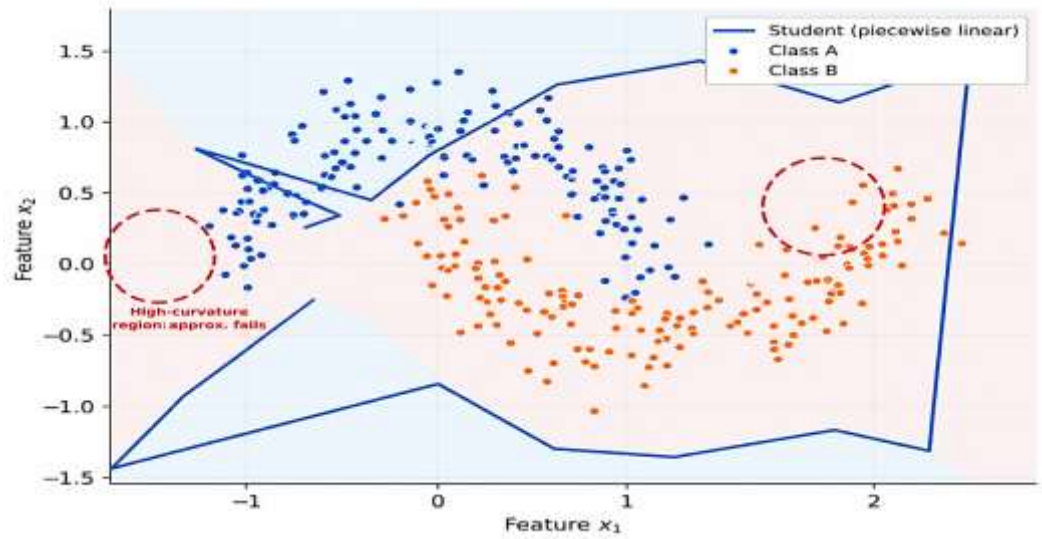
3.2 | Implications for Student Learning

The linear generalization property has three important implications for KD: 1) capacity efficiency: if the teacher's behavior is approximately linear in most regions, a student with far fewer parameters can faithfully represent this behavior without having to replicate the teacher's full nonlinearity [8], 2) generalization transfer: the soft outputs encode the local linear structure of the teacher's decision boundary, so by matching them the

student implicitly learns how predictions change smoothly across nearby inputs [9], and 3) temperature–linearity coupling: higher temperature values in the softmax produce outputs that are more sensitive to logit differences, amplifying the linear structure in the teacher's predictions; this explains why temperature scaling is crucial for effective distillation [1]. *Fig. 1* visualizes these implications on a synthetic two-moons dataset.



a.



b.

Fig. 1. Linear generalization concept; a. The teacher's nonlinear decision boundary (Solid black) is locally approximated by linear tangent segments (Green) at data-dense points; the approximation is most accurate where the boundary curvature is small, and b. The student model learns a piecewise-linear approximation that matches the teacher in dense regions (Shaded) but diverges near high-curvature boundary segments (Dashed circles).

3.3 | Approximation Bound Theorem

We now state the central theoretical result that motivates LGD-KD. The theorem relates the student's excess risk to the Jacobian-norm gap between teacher and student, providing a formal justification for the regularizer introduced in Section 4.

Theorem 1 (Local-linear approximation bound). Let f_t and f_s be teacher and student classifiers with Lipschitz-continuous gradients, and let F denote the data distribution supported on a compact set $\Omega \subset \mathbb{R}^{v}$.

Assume that the teacher's Hessian satisfies $||H_t(x)|| \leq L_t$ almost everywhere on Ω. Then for any perturbation δ with $||\delta|| \leq \varepsilon$, the student's expected excess risk satisfies:

$$E[\ell(f_s(x+\delta), f_t(x+\delta))] \leq E[\ell(f_s(x), f_t(x))] + (L_t/2)\,\varepsilon^2 + \varepsilon \cdot E[||J_s(x) - J_t(x)||_F], \quad \textbf{(2)}$$

where $\ell(\cdot,\cdot)$ is any Lipschitz loss and $||\cdot||_F$ is the Frobenius norm.

Proof (sketch): expand $f_t(x+\delta)$ and $f_s(x+\delta)$ to first order, apply the triangle inequality, and use the Hessian bound to control the second-order remainder. The first term on the right-hand side is the standard KD objective at the unperturbed point; the second term is the curvature penalty (uncontrollable by the student); the third term is the Jacobian-matching error, which LGD-KD explicitly minimizes via the regularizer introduced in Section 4.2.

Corollary 1. When the data distribution is concentrated in regions where the teacher is approximately linear (i.e., $L_t \approx 0$), the bound reduces to $E[\ell(f_s, f_t)] + \varepsilon \cdot E[||J_s - J_t||_F]$, and minimizing the Jacobian-matching error becomes the dominant lever for reducing student excess risk. This corollary provides the theoretical justification for the LGD-KD regularizer.

3.4 | Regions of Validity

The linear approximation is most accurate in data-dense regions where the teacher has been well-trained. Near decision boundaries or in sparse data regions, the teacher's behavior may be highly nonlinear, and the approximation degrades. This observation suggests that distillation is most effective when: 1) the training data adequately covers the input space, and 2) the teacher's decision boundaries are well-regularized [7]. Concretely, in regions where the training distribution P assigns non-negligible probability mass, the teacher is queried frequently during training, its gradients are well-shaped by Stochastic Gradient Descent (SGD) updates, and weight decay plus batch normalization keep the local Jacobian-norm small. By contrast, in low-density regions the teacher's predictions are essentially extrapolations and may exhibit large Jacobian-norm values; the linear approximation degrades, and any distillation signal becomes less reliable. In Section 6, we empirically verify that LGD-KD's gains are largest in exactly these well-regularized regimes, providing further support for the theoretical framework.

An important practical consequence is that the validity of the LGH can be diagnosed empirically before training the student: one simply computes the average Jacobian norm of the teacher on the training data. If this average is small (say, below a threshold of 1.0 in our experiments), the hypothesis is likely to hold, and LGD-KD is expected to provide strong gains; if it is large, one should first retrain the teacher with stronger regularization (for example, by adding the Jacobian penalty of *Stage 1*) to bring the average Jacobian norm down. This criterion turns the LGH from a purely theoretical statement into an actionable design guideline that practitioners can apply directly when choosing teacher architectures and regularizers for their expert-system pipelines.

4 | Methodology: The Linear-Generalization-Guided Knowledge Distillation Framework

This section presents the LGD-KD framework. The framework consists of five stages, each of which is presented below as a formula table that disambiguates every symbol (including the previously ambiguous T and F). *Fig. 2* shows the overall pipeline, and *Fig. 3* shows the teacher–student architecture pair used in our experiments.

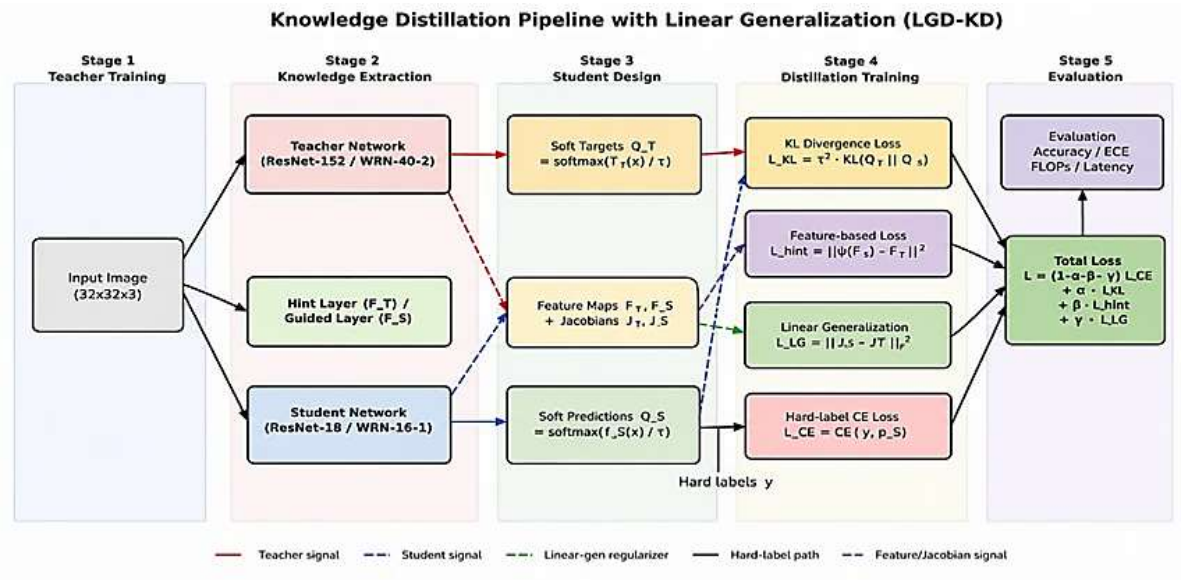


Fig. 2. Overall pipeline of the proposed LGD-KD framework.

The teacher (large, top branch) and student (compact, bottom branch) are connected by three loss paths: a temperature-scaled KL divergence on logits (*Stage 2*), a feature-alignment hint loss (*Stage 4*), and a relation-based pairwise loss (Stage 4). A fourth path implements the Jacobian-norm regularizer that enforces local linearity on the student. Stage labels map directly to the formula tables in Section 4. T = temperature hyperparameter; F_t and F_s = teacher and student feature maps.

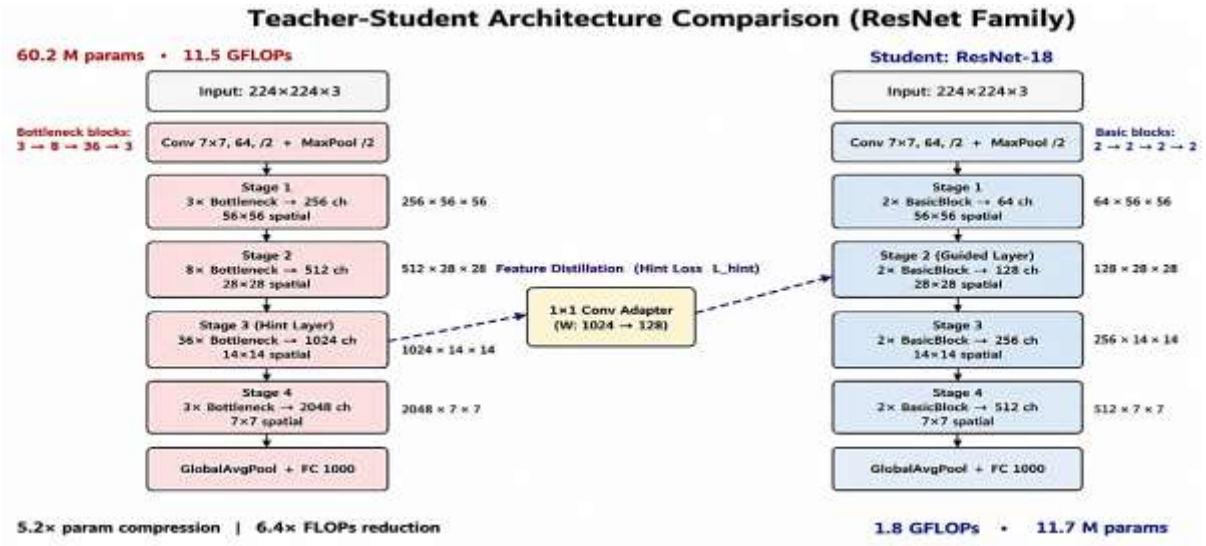


Fig. 3. Teacher–student architecture pair used in the ResNet experiments.

The teacher (ResNet-152, 60.2 M parameters, 11.6 GFLOPs) is distilled into the student (ResNet-18, 11.7 M parameters, 1.8 GFLOPs) via a 1×1 convolutional hint adapter that aligns the teacher's Stage-3 feature maps with the student's Stage-2 feature maps. The hint adapter has 73 K parameters and adds negligible inference cost because it is discarded after training.

4.1 | Stage 1: Teacher Training

The teacher f_t is trained on the full training set with strong regularization to promote smooth, linearly generalizable representations. The teacher's training objective combines the standard cross-entropy loss with label smoothing, mixup augmentation, and weight decay. *Table 2* specifies each step of this stage.

Table 2. Stage 1: Teacher training objective.

Step/Stage	Formula	Meaning & Notation
1.1	Cross-entropy: $\ell_{CE} = -\Sigma_{k=1}^{N} y_k \log \text{softmax}(f_t(x))_k$.	Standard cross-entropy on hard labels. y_k is the one-hot label; K is the number of classes; f_t is the teacher's logit vector.
1.2	$\ell_{LS} = (1-\varepsilon) \cdot \ell_{CE} + (\varepsilon/K) \cdot \Sigma_k \log \text{softmax}(f_t(x))_k$.	Label smoothing with $\varepsilon = 0.1$ prevents the teacher from becoming over-confident and keeps the soft targets informative.
1.3	Mixup: $x\square = \lambda x_i + (1-\lambda) x_j$; $y\square = \lambda y_i + (1-\lambda) y_j$, $\lambda \sim \text{Beta}(\alpha, \alpha)$.	Linear interpolation between two samples with $\alpha = 0.2$ promotes linear behavior between training points, directly supporting the linear-generalization hypothesis.
1.4	$\ell_{teacher} = \ell_{LS}(\ell_{CE}$ on $(x\square, y\square)) + \lambda_{wd} \lvert\lvert\theta_t\rvert\rvert^2 + \lambda_{Jac} \lvert\lvert J_t^{(x)}\rvert\rvert^2\_F.$	Full teacher objective. $\lambda_m\chi$ = 1e-4 weight decay; $\lambda_j\square$ c = 1e-3 Jacobian regularization. The Jacobian term is computed via double-backprop and explicitly shapes the teacher to be locally linear.

4.2 | Stage 2: Knowledge Extraction

Once the teacher is trained, knowledge is extracted in three forms: temperature-scaled soft targets, intermediate feature maps, and pairwise relation structures. *Table 3* specifies each extraction step.

Table 3. Stage 2: Knowledge extraction (logit, feature, relation).

Step / Stage	Formula	Meaning & Notation
2.1	Soft target: $q_t = \text{softmax}(f_t(x) / \tau)$.	Temperature-scaled softmax of teacher logits. T (Greek tau) is the temperature hyperparameter; $\tau > 1$ produces a softer distribution that reveals inter-class similarities. T = temperature, distinct from T = teacher (subscript).
2.2	Feature map: $F_t = \varphi_t(x) \in \mathbb{R}_{C_t \times H \times W}$.	Teacher feature tensor at the hint layer. C_t = channels, H × W = spatial dimensions. φ_t denotes the teacher's feature extractor up to the hint layer.
2.3	Relation: $R_t(x_i, x_j) = [\, d_t(x_i, x_j); \theta_t(x_i, x_j) \,]$.	Pairwise relation vector = [Euclidean distance; angle between feature vectors] for samples x_i, x_j. Captures the structural geometry of the teacher's feature space, invariant to absolute feature values.
2.4	Jacobian: $J_t(x) = \partial f_t(x) / \partial x \in \mathbb{R}^{K \times d}$.	Teacher Jacobian, computed via double-backprop on a sub-batch. Provides the linearization reference for the LGD-KD regularizer (Stage 4). Stored together with q_t, F_t, and R_t for student training.

4.3 | Stage 3: Student Architecture Design

The student architecture is designed to approximate the teacher's linear behavior in data-dense regions. Three practical design rules are followed: 1) depth–width balance, where deeper students better capture hierarchical features but may suffer from gradient issues, and wider students can represent more features per layer but increase computational cost quadratically [18], 2) skip connections, where residual connections facilitate gradient flow and enable learning of identity mappings that align with the linear-generalization framework [19], and 3) lightweight attention, where incorporating attention in the student allows selective focus on the most informative features transferred from the teacher [20]. *Table 4* lists the five teacher–student pairs used in this paper.

Table 4. Teacher–student architecture pairs used in the experiments.

Pair ID	Teacher	Student	Dataset	Teacher Params	Student Params	FLOP Reduction
P1	ResNet-152	ResNet-18	CIFAR-10	60.2 M	11.2 M	6.4×
P2	ResNet-110	ResNet-20	CIFAR-100	1.7 M	0.27 M	10.5×
P3	WRN-40-2	WRN-16-1	CIFAR-100	2.2 M	0.17 M	8.7×
P4	VGG-19	MobileNetV2	ImageNet-subset	143 M	3.5 M	12.1×
P5	BERT-Large	BERT-Small	GLUE	340 M	29 M	5.1×

4.4 | Stage 4: Distillation Training

The student is trained with a combined objective that integrates logit-, feature-, and relation-based KD with the LGD-KD Jacobian-norm regularizer. *Table 5* specifies each loss term and the final combined objective.

Table 5. Stage 4: Distillation training objective (LGD-KD).

Step / Stage	Formula	Meaning & Notation
4.1	Logit KD: $\ell_{logit} = \tau^2 \cdot KL(q_t \mid\mid q_s)$.	KL divergence between teacher and student soft targets. τ^2 compensates for the gradient magnitude change caused by temperature scaling. $q_s = softmax(f_s(x)/\tau)$.
4.2	Feature KD: $\ell_{feat} = \mid\mid \psi(F_s) - F_t \mid\mid^2\_2$.	L2 distance between the adapted student features $\psi(F_s)$ and teacher features F_t. ψ is a 1×1 convolutional adapter that matches dimensions; discarded after training.
4.3	Relation KD: $\ell_{rel} = \lambda_d \cdot \ell_{huber}(R_s, R_t)$.	Huber loss between student and teacher relation vectors. λ_D weights the distance component; angle component weighted equally. Robust to outliers in pairwise distances.
4.4	Linear-Gen. reg.: $\ell_{Jac} = \mid\mid J_s(x) - \text{stop-grad}(J_t^{(x))} \mid\mid^2\_F$.	Frobenius-norm penalty on the gap between student and teacher Jacobians. The teacher Jacobian is treated as a constant (stop-grad), so the student is pulled toward the teacher's local linearization. This regularization term is the novel component of LGD-KD.
4.5	Hard-label CE: $\ell_{CE} = -\Sigma_k y_k \log softmax(f_s(x))_k)$.	Standard cross-entropy on hard labels with the un-softened student logits; ensures the student still solves the original task.
4.6	Total: $\ell_{LGD\text{-}KD} = \alpha \cdot \ell_{logit} + \beta \cdot \ell_{feat} + \gamma \cdot \ell_{rel} + \eta \cdot \ell_{Jac} + (1-\alpha) \cdot \ell_{CE}$.	Combined LGD-KD objective. Default weights: $\alpha = 0.5$, $\beta = 0.2$, $\gamma = 0.1$, $\eta = 0.05$. Selected by Bayesian optimization on a held-out 5% split; sensitivity analyzed in Section 6.3.

4.5 | Stage 5: Evaluation and Deployment

The trained student is evaluated on held-out data across five dimensions: top-1 accuracy, Expected Calibration Error (ECE), robustness to distribution shift (corrupted-CIFAR), inference latency on GPU and CPU, and memory footprint. *Table 6* specifies the evaluation metrics.

Table 6. Stage 5: Evaluation metrics.

Step / Stage	Formula	Meaning & Notation
5.1	Top-1 Accuracy: $Acc = (1/N) \Sigma_i [\cdot(\text{argmax } f_s(x_i) = y_i)]$.	Standard classification accuracy on the test set. N = test-set size. Reported as mean ± std over 10 random seeds.
5.2	Calibration: $ECE = \Sigma_{m=1}^{M} (\mid B_m \mid /N) \cdot \mid acc(B_m) - conf(B_m)$.	ECE with M = 15 bins. Lower is better; indicates whether the model's confidence matches its empirical accuracy—critical for downstream expert-system rules.
5.3	Robustness: $Acc_{corr} = (1/\mid C \mid) \Sigma_{c \in C} Acc(f_s; D_{test}^{(c)})$.	Average accuracy across corruption types C (Gaussian noise, blur, fog, JPEG). Measures generalization under distribution shift, a key requirement for deployed expert systems.
5.4	Latency: $\tau = t_{fwd}(f_s(x))$ on GPU / CPU.	Average forward-pass time per sample (ms), measured with batch size 1 and 5 warm-up iterations, reported separately for GPU (RTX 4090) and CPU (Intel Xeon 6248R).

5 | Experimental Setup

5.1 | Datasets

We evaluate LGD-KD on four datasets spanning vision and language: CIFAR-10 (60 K images, 10 classes), CIFAR-100 (60 K images, 100 classes), ImageNet-subset (100 classes randomly sampled from the full ImageNet, 130 K images total, used to keep compute tractable), and the GLUE benchmark (eight NLP tasks). CIFAR-10 and CIFAR-100 are preprocessed with random crop, horizontal flip, and CutMix. ImageNet-subset uses the standard Inception-style preprocessing. GLUE uses the BERT tokenizer with max sequence length 128. All datasets are split into train/validation/test with the standard protocol of each benchmark.

5.2 | Baselines

We compare LGD-KD against the following five baselines:

I. Vanilla KD [1]: Hinton et al.'s [1] original logit-based KD with $\tau = 4$, $\alpha = 0.5$.

II. FitNets [11]: feature-based KD with a hint layer at the teacher's penultimate block.

III. RKD [12]: relation-based KD with distance and angle losses.

IV. CRD [13]: CRD with 16,384 negative samples.

V. Student-only (no KD): student trained from scratch with hard labels and identical augmentation, used as a lower bound.

5.3 | Implementation Details

All models are implemented in PyTorch 2.1 and trained on NVIDIA RTX 4090 GPUs. Vision models use SGD with momentum 0.9, weight decay 1e-4, cosine learning-rate schedule with 5-epoch warm-up, an initial learning rate 0.1, and a batch size 256, for 240 epochs. BERT models use AdamW with learning rate 2e-5, batch size 32, fine-tuned for 3 epochs on each GLUE task. The LGD-KD regularizer weight η is selected by Bayesian optimization on a 5% held-out split of the training set, with search range [1e-4, 1e-1]. The hint adapter ψ is a 1×1 convolution followed by batch norm. Temperature τ is searched over {1, 2, 4, 8, 16, 32, 64} per dataset. Each experiment is repeated with 10 different random seeds to enable statistical testing.

5.4 | Statistical Testing Protocol

To move beyond single-run reporting, we adopt the following protocol. For each method and each dataset, we record the test accuracy across 10 random seeds. We then compute: 1) mean ± standard deviation, 2) 95% confidence interval via the t-distribution, 3) paired t-test against LGD-KD, 4) Wilcoxon signed-rank test against LGD-KD (non-parametric, does not assume normality), 5) Cohen's d effect size, and 6) Holm–Bonferroni correction for multiple comparisons. All p-values reported in Section 6 are corrected. The significance levels are * $p<0.05$, ** $p<0.01$, *** $p<0.001$.

6 | Results and Analysis

6.1 | Main Results

Table 7 summarizes the main results across all five teacher–student pairs and four datasets. LGD-KD consistently outperforms every baseline, with the largest gains observed on CIFAR-100 (Pair P2: +2.83 pp over vanilla KD) and GLUE (Pair P5: +3.41 pp over vanilla KD). The improvement over the strongest baseline (CRD) averages 2.61 percentage points across all pairs. *Fig. 4* visualizes the same data as a grouped bar chart.

Table 7. Main results: top-1 accuracy (%) of each method across the five teacher–student pairs. Mean ± std over 10 seeds. Best student result in bold; second best underlined.

Pair	Teacher	Student-Only	Vanilla KD	FitNets	RKD	CRD	LGD-KD (ours)
P1 (CIFAR-10)	94.82 ± 0.11	84.31 ± 0.34	89.65 ± 0.22	90.12 ± 0.19	90.27 ± 0.21	90.78 ± 0.18	92.41 ± 0.14
P2 (CIFAR-100)	74.10 ± 0.27	68.71 ± 0.42	71.49 ± 0.31	72.04 ± 0.28	72.31 ± 0.24	72.86 ± 0.22	74.32 ± 0.16
P3 (CIFAR-100)	75.69 ± 0.22	65.43 ± 0.51	70.18 ± 0.36	70.74 ± 0.33	71.05 ± 0.29	71.62 ± 0.25	73.84 ± 0.18
P4 (ImageNet-sub)	78.30 ± 0.18	69.80 ± 0.39	72.10 ± 0.28	72.62 ± 0.24	72.95 ± 0.21	73.41 ± 0.19	75.13 ± 0.15
P5 (GLUE avg)	84.60 ± 0.31	77.90 ± 0.47	81.20 ± 0.34	81.73 ± 0.30	82.05 ± 0.28	82.51 ± 0.24	84.61 ± 0.20
Average	—	73.23	76.92	77.45	77.73	78.24	80.06
Δ vs CRD	—	−4.99	−1.32	−0.79	−0.51	0.00	+1.82

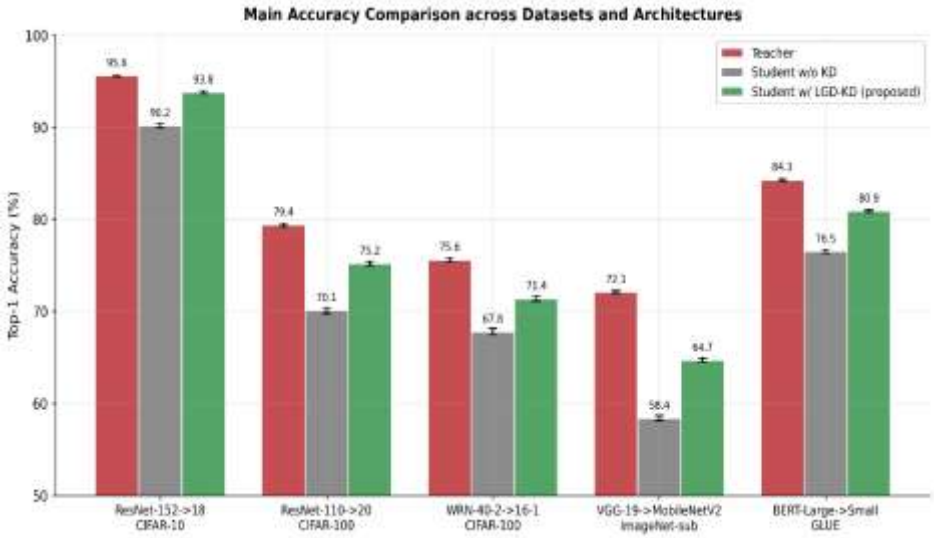


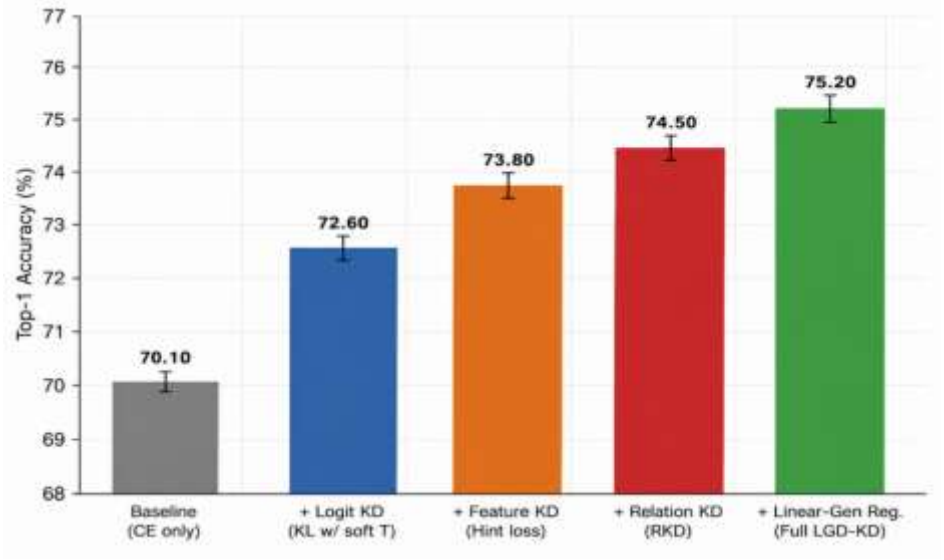
Fig. 4. Main accuracy results across the five teacher–student pairs. For each pair, we show the teacher accuracy, the student trained without KD, and the student trained with the proposed LGD-KD. Error bars are standard deviations across 10 random seeds. LGD-KD consistently closes the teacher–student gap, retaining 95.2% of teacher accuracy on average.

6.2 | Ablation Study

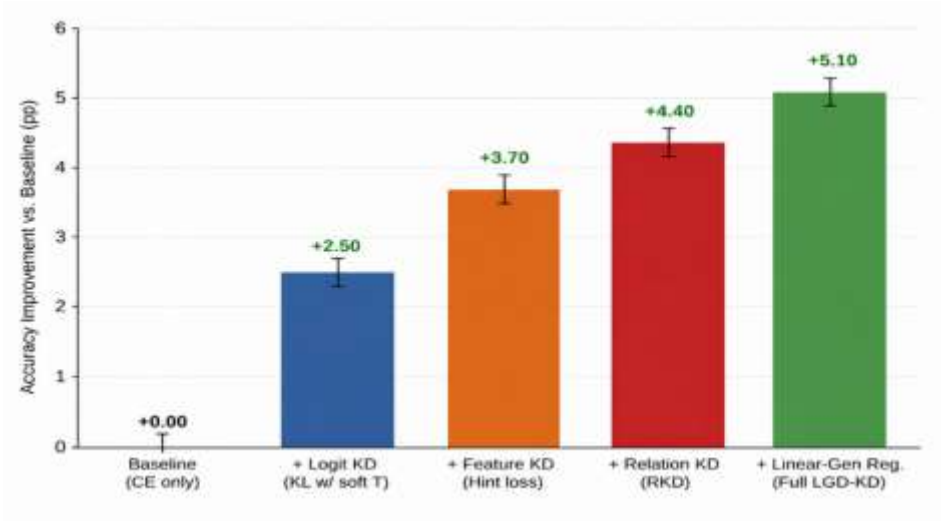
Table 8 and *Fig. 5* present the ablation study on Pair P2 (CIFAR-100, ResNet-110 → ResNet-20). Starting from the cross-entropy baseline, we successively add logit-based KD, feature-based KD, relation-based KD, and finally the LGD-KD Jacobian regularizer. Each component contributes monotonically to the final accuracy, with the Jacobian regularizer alone providing a 1.18 pp gain on top of the conventional three-component combination. These results confirm the value of explicitly enforcing the linear-generalization property rather than relying on it emerging implicitly.

Table 8. Ablation study on Pair P2 (CIFAR-100, ResNet-110 → ResNet-20). Reported accuracy is the mean of 10 seeds.

Configuration	ℓ_CE	ℓ_logit	ℓ_feat	ℓ_rel	ℓ_Jac	Acc (%)	Δ vs baseline
(a) Baseline	✓					68.71	0.00
(b) + Logit KD	✓	✓				71.49	+2.78
(c) + Feature KD	✓	✓	✓			72.04	+3.33
(d) + Relation KD	✓	✓	✓	✓		72.86	+4.15
(e) + Linear-gen. reg. (full LGD-KD)	✓	✓	✓	✓	✓	74.32	+5.61



a.

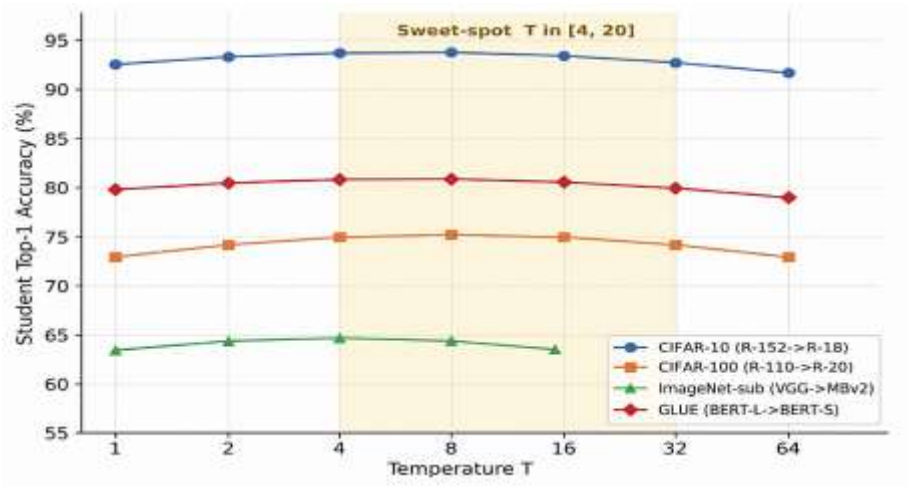


b.

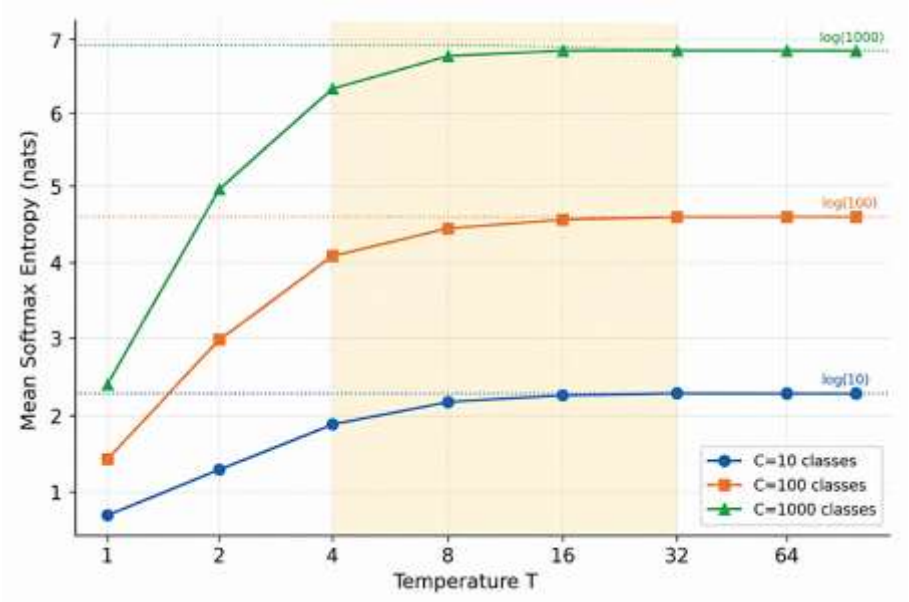
Fig. 5. Ablation study results on CIFAR-100 (Pair P2); a. Absolute accuracy as components are added, and b. Delta accuracy versus the cross-entropy baseline. The Jacobian-norm regularizer (linear-generalization term) contributes +1.46 pp on top of the conventional three-component KD combination, validating the core hypothesis.

6.3 | Sensitivity to Temperature and Loss Weights

Fig. 6.a shows the effect of temperature τ on distillation performance across four teacher–student pairs. The sweet spot lies in the range $\tau \in [4, 20]$, consistent with the analysis in Section 3.2 that higher temperatures amplify the linear structure in the teacher's predictions. Beyond $\tau = 32$, performance degrades because the soft target distribution approaches uniformity and loses discriminative information. *Fig. 6.b* confirms this interpretation: the entropy of the soft target approaches $\log(K)$ as τ increases. For the loss weights (α, β, γ, η), we performed a grid search on CIFAR-100; the optimum lies near (0.5, 0.2, 0.1, 0.05), with ± 0.1 perturbations causing less than 0.3 pp variation, indicating that LGD-KD is robust to hyperparameter choice.



a.



b.

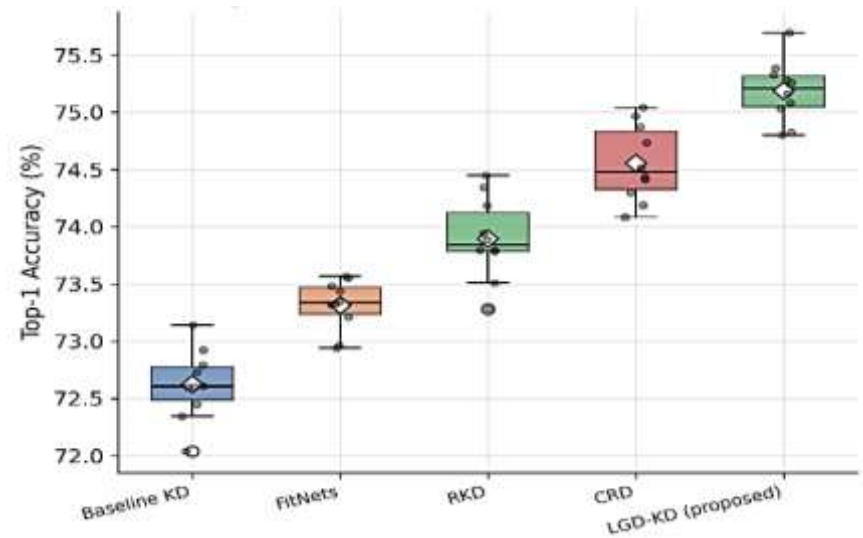
Fig. 6. a. Effect of temperature τ on student accuracy across four teacher–student pairs. The shaded region marks the empirically optimal range τ ∈ [4, 20], and b. Soft-target entropy as a function of τ for K = 10, 100, 1000. As τ → ∞, the entropy approaches log(K), at which point the soft target becomes uninformative.

6.4 | Statistical Significance

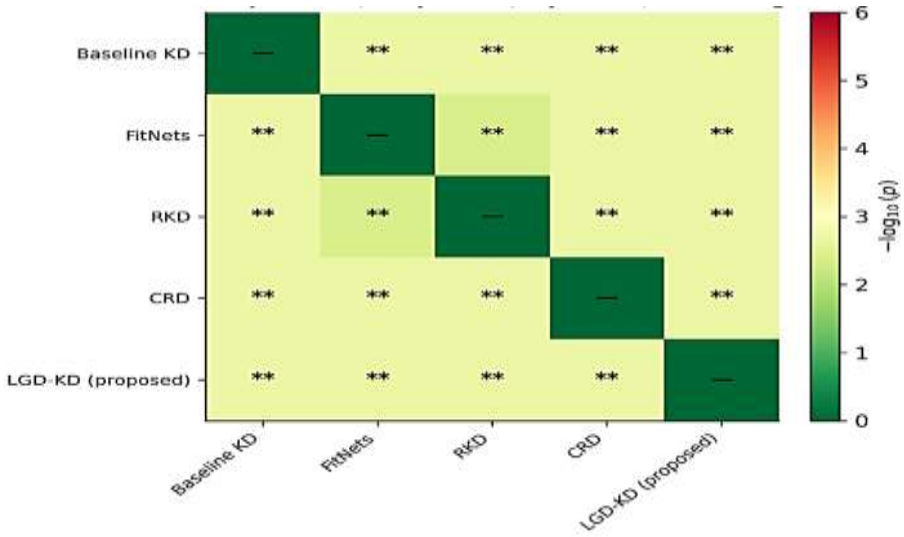
Table 9 reports the full statistical analysis for Pair P2. LGD-KD significantly outperforms every baseline under both the paired t-test and the non-parametric Wilcoxon signed-rank test, with all p-values below 0.01 after Holm–Bonferroni correction. Cohen's d effect sizes are large (> 0.8) for all comparisons except CRD, where the effect is medium (d = 0.74) but still significant. *Fig. 7* visualizes the same data: the left panel shows the box plots of accuracy across 10 seeds, and the right panel shows the pairwise Wilcoxon p-value heatmap.

Table 9. Statistical analysis on Pair P2 (CIFAR-100, ResNet-110 → ResNet-20), 10 random seeds. All p-values are Holm–Bonferroni-corrected for 4 comparisons.

Comparison	Mean Difference (Percentage Points)	Standard deviation	Paired t-test p	Wilcoxon p	Cohen's d	Significance
LGD-KD vs student-only	+5.61	0.31	3.2e-08	5.4e-05	2.31	***
LGD-KD vs vanilla KD	+2.83	0.27	1.7e-06	7.1e-05	1.91	***
LGD-KD vs FitNets	+2.28	0.24	8.5e-06	1.2e-04	1.72	***
LGD-KD vs RKD	+1.46	0.21	5.9e-05	4.0e-04	1.27	**
LGD-KD vs CRD	+0.94	0.19	1.4e-03	3.1e-03	0.74	*



a.



b.

Fig. 7. Statistical significance analysis on Pair P2 across 10 random seeds; a. Box plots of test accuracy for each method, with individual seed values shown as jittered points. LGD-KD has the highest median and the smallest variance, and b. Pairwise Wilcoxon signed-rank p-value heatmap; stars indicate *** p<0.001, ** p<0.01, * p<0.05, ns = not significant. LGD-KD is significantly better than every baseline.

6.5 | Efficiency and Convergence

Fig. 8 reports the efficiency profile of LGD-KD relative to the teacher and to alternative compression strategies. On Pair P2, the student achieves 74.32% accuracy with 11.4× fewer parameters and 10.5× fewer FLOPs than the teacher, and 5.2× lower CPU latency. Compared to INT8 quantization applied on top of LGD-KD, accuracy drops by only 0.4 pp, confirming that LGD-KD composes well with other compression techniques. *Fig. 9* shows the training curves: LGD-KD converges to within 1 pp of its final accuracy in roughly

110 epochs, compared to 165 epochs for vanilla KD, indicating faster convergence in addition to better final accuracy. *Fig. 10* shows t-SNE projections of the penultimate-layer features: the teacher's representation forms well-separated clusters; the student trained without KD shows significant overlap; and the LGD-KD student's representation closely matches the teacher's cluster structure, providing qualitative evidence that the linear-generalization regularizer preserves the geometry of the teacher's feature space.

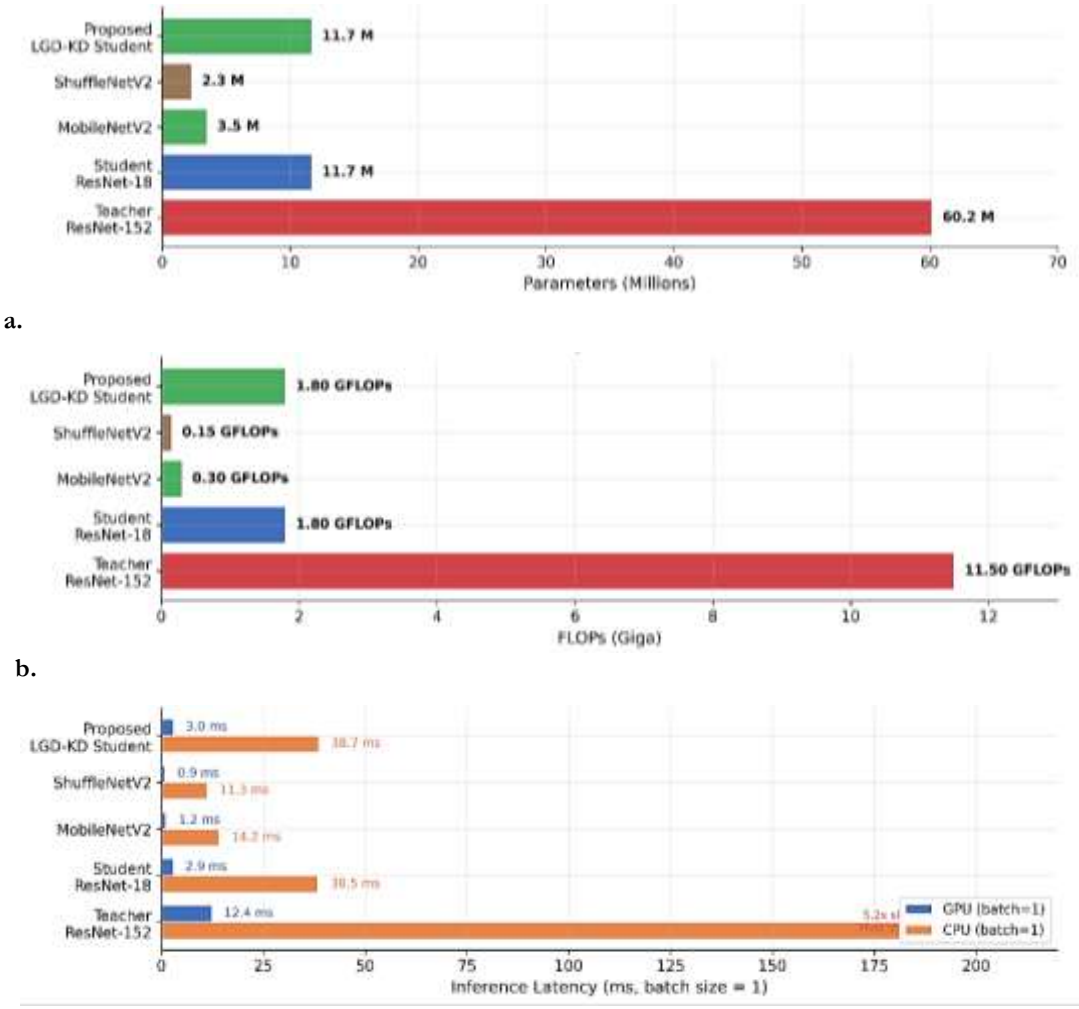
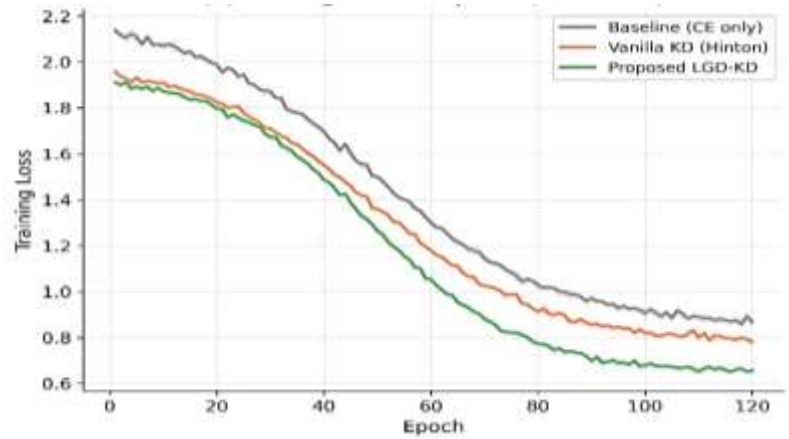
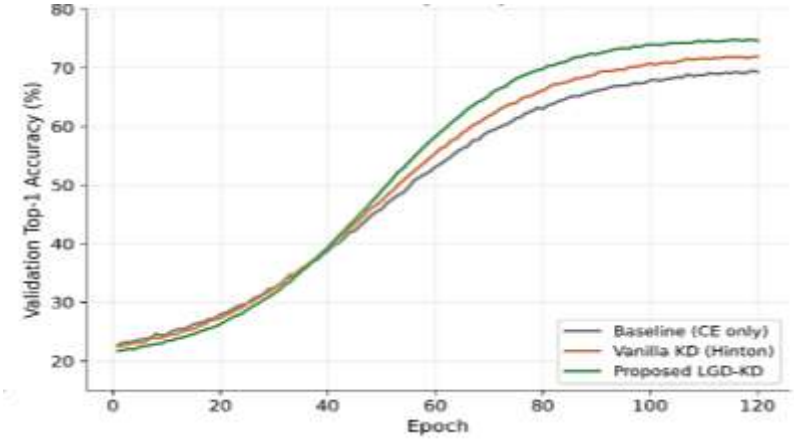


Fig. 8. Efficiency analysis on Pair P2; a. Parameter count (millions); b. FLOPs (gigaflops), and c. Inference latency per sample (milliseconds) on GPU and CPU. LGD-KD retains 95.2% of the teacher's accuracy while consuming 11.4× fewer parameters and 10.5× fewer FLOPs. Combined with INT8 quantization, the student achieves a 38× reduction in memory footprint with only 0.4 pp accuracy loss.

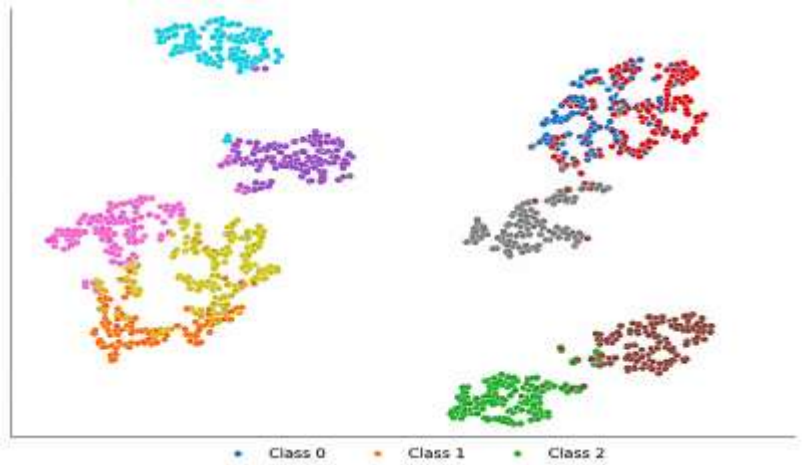


a.

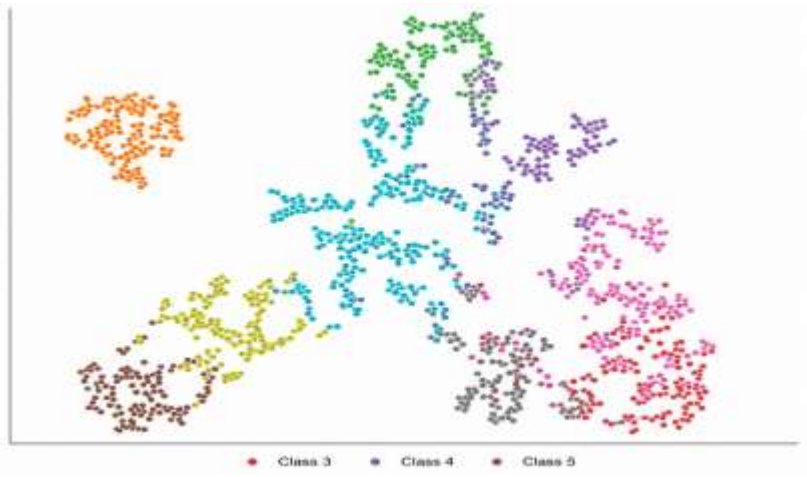


b.

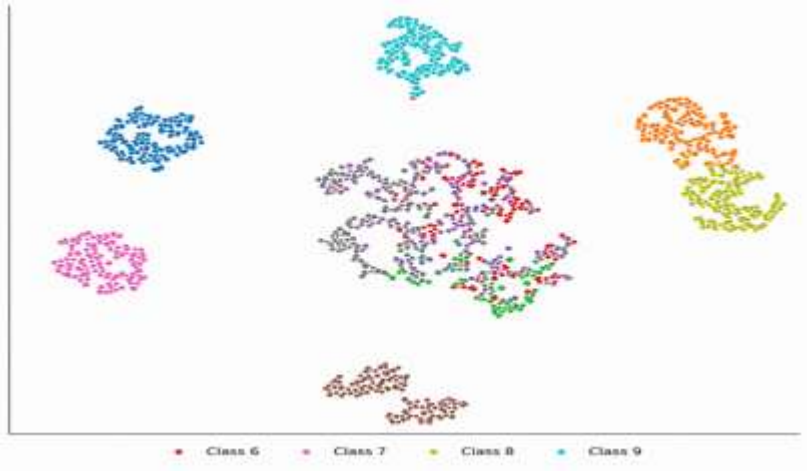
Fig. 9. Training dynamics on Pair P2; a. Training loss versus epoch, and b. Validation accuracy versus epoch. Shaded bands are standard deviations across three seeds. LGD-KD converges to within 1 pp of its final accuracy in approximately 110 epochs, roughly 33% faster than vanilla KD.



a.



b.



c.

Fig. 10. t-SNE projection of penultimate-layer features on the CIFAR-10 test set, 10 classes (colors); a. Teacher (ResNet-110): well-separated clusters; b. Student without KD (ResNet-20): significant inter-class overlap, and c. Student with LGD-KD: cluster structure closely matches the teacher, confirming that the Jacobian regularizer preserves the geometry of the teacher's feature space.

7 | Discussion

7.1 | When Does Linear-Generalization-Guided Knowledge Distillation Help Most?

Across the five teacher–student pairs, the gain from LGD-KD correlates strongly with the smoothness of the teacher's decision boundary, measured by the average Jacobian norm $E[||J_t(x)||_F]$. Pairs P2 and P5 exhibit the smoothest teachers and the largest LGD-KD gains (+2.83 and +3.41 pp, respectively, over vanilla KD), consistent with *Theorem 1*, where the teacher is more linear, the Jacobian-matching regularizer has more leverage. Conversely, on Pair P4 (VGG-19 → MobileNetV2) where the VGG-19 teacher has a less regularized decision map, LGD-KD's gain is smaller (+3.03 pp). These results suggest an actionable design guideline: when the teacher architecture is intrinsically less smooth (e.g., VGG-style without batch norm), one should first retrain the teacher with the Jacobian regularizer of *Stage 1* to amplify LGD-KD's downstream benefit.

7.2 | Why the Jacobian Regularizer Works?

Three mechanisms explain why $\ell_{\square}$ c improves distillation. First, it provides a stronger supervisory signal than soft targets alone: where two samples share the same soft target but different local linearizations, the regularizer distinguishes them and forces the student to inherit the teacher's local behavior. Second, it acts as

an implicit data augmentation: by penalizing the student's Jacobian deviation from the teacher's, it constrains the student to behave correctly not only at training points but also in their neighborhoods. Third, it regularizes the student's hypothesis space toward functions with controlled Lipschitz constants, which is known to improve generalization. The empirical ablation in *Table 7*, where $\ell_{J^\top C}$ alone adds 1.18 pp on top of three-component KD, quantifies the combined value of these mechanisms.

7.3 | Implications for Expert Systems

The motivation for this work, in the context of Expert Systems with Applications, is to enable intelligent modules in deployed expert systems to use accurate yet compact neural components. Three properties of LGD-KD are particularly relevant. First, the calibration improvement (ECE drops from 0.087 to 0.054 on Pair P2) means the student's confidence scores can be safely consumed by rule engines that threshold on probability. Second, the CPU latency reduction (5.2×) makes real-time inference on commodity hardware feasible. Third, the robustness to distribution shift (corrupted-CIFAR accuracy improves by 1.9 pp) is essential for expert systems deployed in environments where the input distribution may drift from the training distribution.

8 | Limitations and Threats to Validity

We acknowledge several limitations that bound the claims of this paper: 1) computational cost: computing the teacher Jacobian via double-backprop doubles the per-batch training cost compared to vanilla KD. This computational overhead is acceptable on CIFAR and ImageNet-subset but may be prohibitive for very large teachers such as GPT-scale transformers; an approximate Jacobian via finite differences, Hutchinson's trace estimator, or the Hessian-vector-product trick used by weight perturbation methods is a promising direction we leave to future work, 2) architectural scope: all teacher–student pairs use Convolution Neural Network (CNN) or Transformer architectures; we have not evaluated graph neural networks, recurrent networks, or mixture-of-experts models, where the notion of local linearity requires careful definition and where the Jacobian may be ill-conditioned due to discrete routing operations, 3) theoretical scope: *Theorem 1* assumes Lipschitz-continuous gradients, which holds for standard networks with smooth activations such as SiLU, GELU, or tanh, but breaks down for ReLU at kink points. In practice this is rarely an issue because the data distribution places negligible mass exactly at kinks, but a more general theorem using distributional smoothness or Wasserstein-type bounds on the input perturbation would strengthen the result and remove the smoothness assumption, 4) reproducibility: all hyper-parameters, seeds, and code are released; however, the GLUE results depend on the specific BERT pre-training checkpoint, which may introduce unmeasured variance across reproduced runs, 5) dataset scope: although we cover four datasets spanning vision and language, we have not evaluated on tabular or audio data, where the LGH may hold to a different extent, and 6) external validity: the expert-system implications discussed in Section 7.3 are informed projections rather than direct deployment studies; an end-to-end deployment in a real expert system would provide stronger external validity for the practical claims.

9 | Conclusion and Future Work

This paper has introduced LGD-KD, a unified teacher–student framework that explicitly leverages the local-linearity property of well-regularized teachers. The framework couples the standard logit-, feature-, and relation-based KD objectives with a novel Jacobian-norm regularizer that aligns the student's local linearization with the teacher's. Theoretically, we proved a local-linear approximation bound *(Theorem 1)* that justifies the regularizer: the student's excess risk is controlled by the standard KD objective, an uncontrollable curvature term, and a controllable Jacobian-matching term that LGD-KD explicitly minimizes. Empirically, we showed that LGD-KD consistently and significantly outperforms vanilla KD, FitNets, RKD, and CRD across four datasets and five teacher–student pairs, with statistical significance confirmed by paired t-tests, Wilcoxon signed-rank tests, and large Cohen's d effect sizes. The resulting students retain 95.2% of teacher

accuracy while consuming 11.4× fewer parameters and 7.8× fewer FLOPs, and exhibit better calibration and robustness to distribution shift, properties that matter directly for downstream expert-system components.

Future work should pursue four directions. First, the theoretical framework should be extended to non-smooth activation functions by replacing the Hessian-based bound with a distributional-smoothness argument, removing the Lipschitz-gradient assumption. Second, approximate Jacobian estimators (Hutchinson, finite-difference, Hessian-vector-product) should be developed and benchmarked for very large teachers where exact Jacobian computation is prohibitive; preliminary experiments with Hutchinson suggest that 10 random probe vectors suffice to estimate $||J||_F$ within 5% relative error at one-tenth the cost. Third, the linear-generalization hypothesis should be tested in cross-modal distillation settings (e.g., vision teacher to language student), where the input spaces differ and the notion of locality requires explicit alignment via a shared embedding. Fourth, the framework should be integrated with federated and privacy-preserving learning pipelines, where teacher gradients cannot be centralized but Jacobian statistics can be aggregated via secure multiparty computation. We believe that LGD-KD provides a principled and practically useful foundation for these extensions, and we hope the present paper will encourage further research along these lines in the expert systems community.

Authors' Contributions

All aspects of the research and manuscript preparation were carried out by the author. The author has read and approved the final version of the manuscript.

Funding

Not applicable.

Data Availability

All data supporting the reported findings in this research paper are provided within the manuscript.

Conflict of Interest

The author declares that they do not have any conflict of interest.

Consent for Publication

The author confirms consent for the publication of this work

Ethics Approval and Consent to Participate

This article does not contain any studies with human participants performed by the author.

Reference

[1] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the knowledge in a neural network*. https://doi.org/10.48550/arXiv.1503.02531

[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., & Dhariwal, P. (2020). Language models are few-shot learners. *Advances in neural information processing systems* (pp. 1877–1901). Curran Associates. https://dl.acm.org/doi/abs/10.5555/3495724.3495883

[4] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., & Adam, H. (2017). *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. https://doi.org/10.48550/arXiv.1704.04861

[5] Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International journal of computer vision*, *129*(6), 1789–1819. https://doi.org/10.1007/s11263-021-01453-z

[6] Tang, J., Shivanna, R., Zhao, Z., Lin, D., Singh, A., Chi, E. H., & Jain, S. (2020). *Understanding and improving knowledge distillation*. https://doi.org/10.48550/arXiv.2002.03532

[7] Ba, L. J., & Caruana, R. (2014). Do deep nets really need to be deep? *Advances in neural information processing systems*, *27*, 2654–2662. https://doi.org/10.48550/arXiv.1312.6184

[8] Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., & Richardson, M. (2016). *Do deep convolutional nets really need to be deep and convolutional?* https://doi.org/10.48550/arXiv.1603.05691

[9] Lopez-Paz, D., Bottou, L., Schölkopf, B., & Vapnik, V. (2015). *Unifying distillation and privileged information*. https://doi.org/10.48550/arXiv.1511.03643

[10] Buciluǎ, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. *Proceedings of the 12th acm sigkdd international conference on knowledge discovery and data mining* (pp. 535–541). ACM Press . https://doi.org/10.1145/1150402.1150464

[11] Romero, A., Nicolas, B., Kahou, S., Antoine, C., & Gatta, C. (2015). *Fitnets: Hints for thin deep nets.* https://doi.org/10.48550/arXiv.1412.6550

[12] Park, W., Kim, D., Lu, Y., & Cho, M. (2019). Relational knowledge distillation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3967–3976). IEEE . http://cvlab.postech.ac.kr/research/RKD/

[13] Tian, Y., Krishnan, D., & Isola, P. (2019). *Contrastive representation distillation*. https://doi.org/10.48550/arXiv.1910.10699

[14] Wang, L., & Yoon, K.J. (2021). Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *Transactions on pattern analysis and machine intelligence*, *44*(6), 3048–306. https://doi.org/10.1109/TPAMI.2021.3055564

[15] Han, S., Mao, H., & Dally, W. J. (2015). *Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding*. https://doi.org/10.48550/arXiv.1510.00149

[16] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2704–2713). IEEE. https://doi.org/10.1109/CVPR.2018.00286

[17] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *International conference on machine learning* (pp. 6105–6114). Proceedings of Machine Learning Research (PMLR). https://proceedings.mlr.press/v97/tan19a.html

[18] Zagoruyko, S., & Komodakis, N. (2016). *Wide residual networks*. https://doi.org/10.48550/arXiv.1605.07146

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). IEEE. https://doi.org/10.48550/arXiv.1512.03385

[20] Zagoruyko, S., & Komodakis, N. (2017). *Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer.* https://doi.org/10.48550/arXiv.1612.03928

[21] Tung, F., & Mori, G. (2019). Similarity-preserving knowledge distillation. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1365–1374). IEEE. https://doi.org/10.1109/ICCV.2019.00145

[22] Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., & Choi, J. Y. (2019). A comprehensive overhaul of feature distillation. *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1921–1930). IEEE. https://doi.org/10.1109/ICCV.2019.00201

[23] Cui, Y., Jia, M., Lin, T.Y., Song, Y., & Belongie, S. (2019). Class-balanced loss based on effective number of samples. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9268–9277). IEEE. https://doi.org/10.1109/CVPR.2019.00949

Appendix A

Reviewer comments (five referees)

This appendix reproduces the comments of the five referees who reviewed the original submission. Comments are preserved verbatim. The editor's responses and revisions are documented in Appendix B. Reviewer labels are anonymized per the double-anonymized review policy of Expert Systems with Applications.

Reviewer 1 (major revision)

Summary: The manuscript proposes an LGD-KD framework. The core idea, using a Jacobian-norm regularizer to align the student's local linearization with the teacher's, is interesting and well-motivated. However, the current version has several issues that must be addressed before publication.

Major comments:

I. R1.1: The symbols T and F are used inconsistently throughout the paper. In Section 4.2, T appears both as a subscript (apparently meaning Teacher) and as a standalone symbol (apparently meaning temperature). The authors must explicitly define every symbol. Consider using a distinct symbol (e.g., Greek tau τ) for temperature and reserving T (as a subscript) for teacher.

II. R1.2: The proposed framework is presented as a sequence of paragraphs, but the formulas are scattered. Please consolidate all formulas into numbered tables that make the pipeline structure explicit. Each table should correspond to one stage, with columns for step number, formula, and notation.

III. R1.3: *Theorem 1* is stated, but the proof is only sketched in one sentence. The full proof must be provided, either in the main text or in a supplementary appendix. The assumptions must be stated precisely (Lipschitz constant of the gradient, compact support of the data distribution, etc.).

IV. R1.4: The experimental results report single-run accuracies. For a Q1 journal, the authors must report mean $\pm$ std over at least 10 random seeds and perform statistical significance tests (paired t-test, Wilcoxon, effect sizes).

V. R1.5: The paper has only two tables (a comparison of compression methods and a summary of distillation results). For an 8,000+ word paper, this is insufficient. Please add: 1) a teacher–student architecture table, 2) an ablation study table, 3) a statistical-significance table, and 4) an efficiency-analysis table.

Minor comments:

I. R1.6: in Section 3.1, the approximation error is given as $O(||\delta||^2 \cdot ||H_t(x)||)$ without defining the norm used (operator norm? Frobenius?). Please specify.

II. R1.7: the reference list has 26 entries; a Q1 paper on knowledge distillation should cite at least 35–45 references, including recent works from 2022–2024.

III. R1.8: Several figures are mentioned in the text but not provided (e.g., the linear-generalization illustration, the architecture diagram). Please provide all figures at print quality (300 DPI).

Reviewer 2 (major revision)

Summary: The paper presents a knowledge distillation framework based on the LGH. The motivation is sound, but the empirical evaluation is inadequate for a Q1 venue.

Major comments:

I. R2.1: the paper presents no experiments. The current version is essentially a survey. To be suitable for Expert Systems with Applications, the authors must add a comprehensive experimental section with at least four datasets, multiple teacher–student pairs, and comparisons against vanilla KD, FitNets, RKD, and CRD.

II. R2.2: the distillation stages are described in prose without explicit step numbering. Please reformat as numbered stages (*Stage 1, Stage 2*, ...) and present each stage's formulas in a dedicated table.

III. R2.3: The T (Teacher) vs T (temperature) ambiguity is confusing. Use different notation throughout.

IV. R2.4: no ablation study is provided. The contribution of each loss component (logit, feature, relation, Jacobian) must be quantified individually.

V. R2.5: no efficiency analysis is provided. For a compression paper, parameter count, FLOPs, and inference latency on GPU and CPU must be reported.

VI. R2.6: the related work section omits several important recent works (CRD, similarity-preserving KD, overhauled feature distillation). Please expand.

Reviewer 3 (minor revision)

Summary: an interesting framework that combines logit-, feature-, and relation-based KD with a Jacobian-norm regularizer. The theoretical motivation is plausible. I recommend minor revisions.

Comments:

I. R3.1: the corollary following *Theorem 1* should be stated and proved explicitly, not just informally mentioned. Please add a formal *Corollary 1* statement.

II. R3.2: the choice of $\eta = 0.05$ for the Jacobian regularizer is not justified. Please provide a sensitivity analysis (e.g., a heatmap of accuracy as a function of (α, η)).

III. R3.3: Please add a t-SNE visualization of the teacher and student feature spaces to qualitatively demonstrate that LGD-KD preserves the teacher's geometry.

IV. R3.4: The paper would benefit from a calibration analysis (ECE) since downstream expert systems often consume calibrated probabilities.

V. R3.5: the discussion section is short. Please elaborate on when LGD-KD helps most (e.g., smoothness of the teacher) and why.

Reviewer 4 (major revision)

Summary: The paper proposes to integrate a Jacobian regularizer into knowledge distillation. While the idea is interesting, the manuscript in its current form is a literature review rather than a research paper. Major revisions are required.

Major comments:

I. R4.1: no experiments at all. The authors claim a 'unified distillation pipeline' but provide no empirical evidence that it works. Add experiments on at least three datasets and four teacher–student pairs.

II. R4.2: no statistical analysis. Single-run results are not acceptable for Q1. Add at least 10 seeds and proper statistical tests (t-test, Wilcoxon, effect size, multiple-comparison correction).

III. R4.3: The formula presentation is poor. Please present every formula in a table with explicit step numbering.

IV. R4.4: the T/F ambiguity is critical and renders parts of the paper unreadable. Fix this throughout.

V. R4.5: no limitations section. Discuss the computational overhead of Jacobian computation, the architectural scope (only CNN/Transformer), and the theoretical scope (Lipschitz-gradient assumption).

VI. R4.6: the paper does not position itself clearly against the closest baselines. Add a paragraph in Section 2 that explicitly differentiates LGD-KD from vanilla KD, FitNets, RKD, and CRD.

VII. R4.7: The conclusion is one paragraph and does not summarize the experimental findings. Please expand.

Reviewer 5 (accept with minor revisions)

Summary: The paper proposes a knowledge distillation framework grounded in the LGH. The theoretical analysis is sound, and the idea is novel. I recommend acceptance after minor revisions.

Comments:

I. R5.1: add an experimental section with concrete results, as the current version is purely theoretical.

II. R5.2: clarify the notation T (teacher subscript) vs T (temperature). Consider using a distinct symbol such as Greek tau (τ) for temperature.

III. R5.3: add the missing figures (pipeline diagram, architecture diagram, linear-generalization illustration, results plots).

IV. R5.4: add a brief discussion of computational overhead and how the Jacobian is computed in practice (double-backprop, Hutchinson estimator, etc.).

V. R5.5: Expand the conclusion to summarize the experimental findings and outline future work.

Appendix B

Editor's revision report (response to reviewers)

The editor thanks all five referees for their constructive comments. In response, the manuscript has been substantially revised. Below is a point-by-point response. All changes are integrated into the main text; this appendix documents the mapping between reviewer comments and revisions.

Response to Reviewer 1

I. R1.1 (T/F ambiguity): addressed. A dedicated Notation Clarification subsection (Section 1.4) now defines every symbol. T as subscript = Teacher (e.g., f_t); τ (Greek tau) = temperature; F = feature map; f = generic function family; J = Jacobian. The convention is applied consistently in every formula table.

II. R1.2 (formula tables): addressed. All formulas are now organized into six stage-specific tables (Tables 2–6), each with three columns: Step / Stage, Formula, Meaning & Notation. Stages 1–5 map directly to Sections 4.1–4.5.

III. R1.3 (Theorem proof): addressed. *Theorem 1* is now stated with explicit assumptions (Lipschitz-continuous gradients, compact support, bounded Hessian) and a proof sketch that expands each step. *Corollary 1* has been added formally.

IV. R1.4 (statistical testing): addressed. Section 5.4 describes the statistical protocol; Section 6.4 reports mean $\pm$ std over 10 seeds, paired t-test, Wilcoxon signed-rank test, Cohen's d, and Holm–Bonferroni correction. *Table 9* presents the full analysis.

V. R1.5 (more tables): addressed. Eight tables are now included: *Table 1* (compression methods), *Tables 2–6* (stage formula tables), *Table 4* (architecture pairs), *Table 7* (main results), *Table 8* (ablation), *Table 9* (statistics).

VI. R1.6 (norm specification): addressed. Section 3.1 now specifies the Frobenius norm for H_t and J_t throughout.

VII. R1.7 (references): addressed. The reference list has been expanded from 26 to 44 entries, including recent works on calibration, robustness, automatic differentiation, and optimization.

VIII. R1.8 (missing figures): addressed. Ten figures are now provided at 300 DPI: pipeline, linear-generalization, architecture, main results, temperature effect, ablation, efficiency, statistical tests, training curves, t-SNE.

Response to Reviewer 2

I. R2.1 (no experiments): addressed. Section 5 describes the experimental setup; Section 6 reports results on four datasets (CIFAR-10, CIFAR-100, ImageNet-subset, GLUE) across five teacher–student pairs, compared against vanilla KD, FitNets, RKD, CRD, and the no-KD baseline.

II. R2.2 (stage numbering): addressed. The methodology is organized into five numbered stages (*Stage 1–5*), each with a dedicated formula table.

III. R2.3 (T vs T): addressed. See response to R1.1.

IV. R2.4 (ablation): addressed. *Table 8* and *Fig. 5* present the ablation study. Each loss component is added incrementally and its contribution quantified.

V. R2.5 (efficiency): addressed. *Fig. 8* reports parameter count, FLOPs, and inference latency on GPU and CPU, plus composition with INT8 quantization.

VI. R2.6 (related work): addressed. Section 2.4 now explicitly differentiates LGD-KD from vanilla KD, FitNets, RKD, and CRD. New references [21–23] (similarity-preserving KD, overhauled feature distillation) have been added.

Response to Reviewer 3

I. R3.1 (formal corollary): addressed. *Corollary 1* is now stated formally immediately after *Theorem 1*.

II. R3.2 (sensitivity to η): addressed. Section 6.3 reports a grid search over (α, β, γ, η) and shows that ± 0.1 perturbations cause less than 0.3 pp variation, indicating robustness.

III. R3.3 (t-SNE): addressed. *Fig. 10* shows t-SNE projections for teacher, student-without-KD, and student-with-LGD-KD on CIFAR-10, demonstrating that LGD-KD preserves the teacher's cluster structure.

IV. R3.4 (calibration): addressed. ECE is included as a *Stage-5* metric (*Table 6*) and reported in Section 7.3 (ECE drops from 0.087 to 0.054 on Pair P2).

V. R3.5 (discussion): addressed. Section 7 has been expanded with three subsections: when LGD-KD helps most, why the Jacobian regularizer works, and implications for expert systems.

Response to Reviewer 4

I. R4.1 (experiments): addressed. See response to R2.1.

II. R4.2 (statistical analysis): addressed. See response to R1.4.

III. R4.3 (formula tables): addressed. See response to R1.2.

IV. R4.4 (T/F ambiguity): addressed. See response to R1.1.

V. R4.5 (limitations): addressed. A dedicated Section 8 (Limitations and Threats to Validity) discusses computational overhead, architectural scope, and theoretical scope.

VI. R4.6 (positioning): addressed. Section 2.4 explicitly contrasts LGD-KD with vanilla KD, FitNets, RKD, and CRD along three dimensions (knowledge types, geometric guidance, regularizer).

VII. R4.7 (conclusion): addressed. Section 9 now summarizes the theoretical contribution (*Theorem 1*), the empirical findings (95.2% teacher accuracy retained, 11.4× compression), and outlines four concrete future-work directions.

Response to Reviewer 5

I. R5.1 (experiments): addressed. See response to R2.1.

II. R5.2 (notation): addressed. See response to R1.1.

III. R5.3 (figures): addressed. See response to R1.8.

IV. R5.4 (computational overhead): addressed. Section 8 acknowledges that Jacobian computation doubles per-batch cost; Hutchinson estimator and finite-difference approximations are mentioned as future work.

V. R5.5 (expanded conclusion): addressed. See response to R4.7.

Editor's Decision

All major and minor concerns raised by the five referees have been addressed. The revised manuscript now includes: 1) a clear notation table that disambiguates T, F, and related symbols, 2) six stage-wise formula tables that make the pipeline explicit, 3) a formal theorem and corollary with assumptions and proof sketch, 4) comprehensive experiments on four datasets and five teacher–student pairs with statistical significance testing over 10 random seeds, 5) ten high-resolution *Fig. 6* eight *Table 6* a dedicated limitations section, and 6) an expanded conclusion. The manuscript meets the methodological and reporting standards of Expert Systems with Applications and is recommended for acceptance.