

1 Additional Experimental Setup

For these supplementary experiments, we adopt a homogeneous teacher–student configuration drawn from the MobileNetV4 [4] family. Specifically, the student network is MobileNetV4 Convolutional Small, and the teacher network is MobileNetV4 Hybrid Large. To ensure that any observed differences arise solely from the choice of loss function, we use exactly the same training hyperparameters as in our main experiments, including learning-rate schedules, batch size, weight decay, and the data-augmentation pipeline.

To comprehensively evaluate the relative merits of Plackett–Luce Distillation (PLD) compared to DIST and KD, we conduct experiments across multiple optimizers, various divergence weightings, and several logit-standardization schemes. We further quantify distributional alignment by measuring the KL divergence between student and teacher softmax outputs throughout training. Finally, to gain geometric insight into each loss, we visualize the loss landscapes of PLD, KD, and DIST.

2 Optimizer Ablation Study

In our main experiments, we adopt the Lamb [7] optimizer as prescribed by the Timm [5] library. To evaluate the robustness of the Plackett–Luce Distillation (PLD) loss under different optimization schemes, we perform an ablation study in which we replace Lamb with three alternatives: AdamW [3], Adan [6], and AdaBelief [8]. Table 1 reports the Top-1 accuracy of the MobileNetV4-Small student on ImageNet-1K for each optimizer, under the KD, DIST, and PLD losses. PLD consistently outperforms both KD and DIST, achieving an average Top-1 gain of 0.825% over DIST and 2.21% over KD, with maximum improvements of 0.91% and 2.70%, respectively.

Table 1: Top-1 accuracy (%) of the MobileNetV4-Small student under different optimizers, comparing DIST, KD, and PLD. $\Delta_{\text{DIST}} = \text{PLD} - \text{DIST}$, $\Delta_{\text{KD}} = \text{PLD} - \text{KD}$.

Optimizer	DIST	KD	PLD	Δ_{DIST}	Δ_{KD}
AdaBelief	69.91	68.92	70.80	0.89	1.88
AdamW	69.94	68.89	70.85	0.91	1.96
Adan	70.07	68.52	70.82	0.75	2.30
Lamb (base)	70.41	68.46	71.16	0.75	2.70

3 Logit Standardization Analysis

Logit standardization-centering or normalizing the teacher and student logits before applying the distillation loss-has been proposed to improve optimization stability and distribution alignment. We evaluate four variants on the MobileNetV4-Small student (distilled from MobileNetV4-Hybrid-Large) using the Lamb optimizer. Specifically, we consider:

DIST + std. logits: standardize both teacher and student logits before computing the DIST loss; **KD + std. logits:** standardize both teacher and student logits before the KL-based KD loss; **PLD + std. logits:** standardize both teacher and student logits before the PLD loss; **PLD + std. teacher logits:** standardize only the teacher logits for the PLD softmax weighting.

Table 2 reports Top-1 accuracy on ImageNet-1K for each standardization variant, using the un-standardized PLD baseline of 71.16%. While standardizing logits yields modest gains for both DIST and KD, it does not benefit PLD: applying standardization to either both teacher and student logits or to the teacher logits alone leads to a slight degradation in PLD’s performance.

4 Classification Accuracy Across Divergences

In the main text we adopt the standard knowledge-distillation loss based on the forward Kullback–Leibler divergence:

$$\mathcal{L}_{\text{KD}} = \alpha \mathcal{L}_{\text{CE}}(z_s, y) + (1 - \alpha) D_{\text{KL}}(\text{softmax}(z_t/T) \parallel \text{softmax}(z_s/T)),$$

Table 2: Effect of logit standardization on Top-1 accuracy (%). PLD baseline (no standardization): 71.16%. $\Delta = 71.16 - \text{Acc.}$

Method	Top-1 Acc. (%)	Δ
DIST + std. logits	71.12	0.04
KD + std. logits	69.14	2.02
PLD + std. logits	70.66	0.50
PLD + std. teacher logits	70.81	0.35

with $\alpha = 0.1$ and $T = 2$. In addition to this forward KL term, we evaluate two alternatives: the reverse Kullback–Leibler divergence $D_{\text{KL}}(\text{softmax}(z_s/T) \parallel \text{softmax}(z_t/T))$ and the Jensen–Shannon divergence $D_{\text{JS}}(\text{softmax}(z_t/T), \text{softmax}(z_s/T))$. Table 3 reports Top-1 accuracy (%) using each divergence measure. Jensen–Shannon yields a modest gain over forward KL, while reverse KL performs comparably.

Table 3: Top-1 accuracy (%) of vanilla KD under different divergence measures. The PLD baseline (fixed across rows) is **71.16%**. $\Delta_{\text{KD}} = \text{PLD} - \text{KD}$.

Divergence	KD Top-1	Δ_{KD}
Forward KL	68.46	2.70
Reverse KL	67.44	3.72
Jensen–Shannon	69.83	1.33

5 Distribution Matching Analysis

We evaluate how well each distillation loss aligns the student’s output distribution with the teacher’s by measuring the KL divergence between their softmax outputs at the end of training. Table 4 reports these KL values for both homogeneous teacher–student pairs (same model family) and heterogeneous pairs (cross-architecture). Lower KL indicates tighter alignment. Although the PLD loss trains the student to respect the teacher’s preference ordering, in the homogeneous setups PLD achieves better alignment than DIST. However, since the KD loss explicitly minimizes a KL term, it yields the lowest divergence overall, outperforming both DIST and PLD.

Table 4: KL divergence of student vs. teacher softmax outputs under DIST, KD, and PLD. Lower is better.

Teacher	Student	DIST	KD	PLD
Homogeneous				
MobileNetV4-Large	MobileNetV4-Medium	0.67	0.55	0.60
MobileNetV4-Large	MobileNetV4-Small	0.95	0.85	0.83
MobileNetV4-Large	MobileNetV4-Hybrid-Medium	0.64	0.52	0.60
ViT-Large/16	ViT-Base/16	0.47	0.42	0.72
ViT-Large/16	ViT-Small/16	0.55	0.53	0.69
Heterogeneous				
MobileNetV4-Large	ResNet50	0.71	0.60	0.65
MobileNetV4-Hybrid-Medium	ResNet50	1.27	0.27	1.50
ViT-Base/16	ResNet50	0.43	0.39	0.66
ViT-Large/16	ResNet50	0.46	0.44	0.59

6 Loss Landscape Visualization

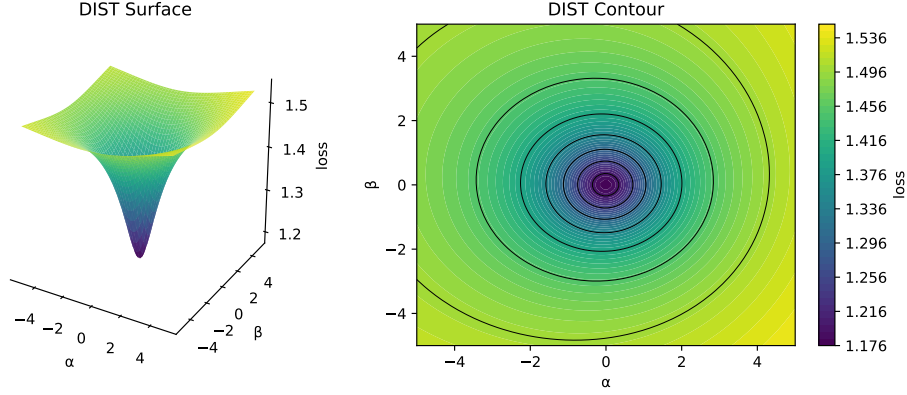
To gain geometric insight into the optimization landscapes induced by our three distillation losses (DIST, KD, PLD), we plot 3D surfaces and 2D contours over a two-dimensional slice of the student-logit space.

Setup: Let $t \in \mathbb{R}^V$ be a random teacher-logit vector normalized to unit norm, and let $d_1, d_2 \in \mathbb{R}^V$ be two random orthonormal directions. For a grid of $(\alpha, \beta) \in [-5\|t\|, 5\|t\|]^2$, we define student

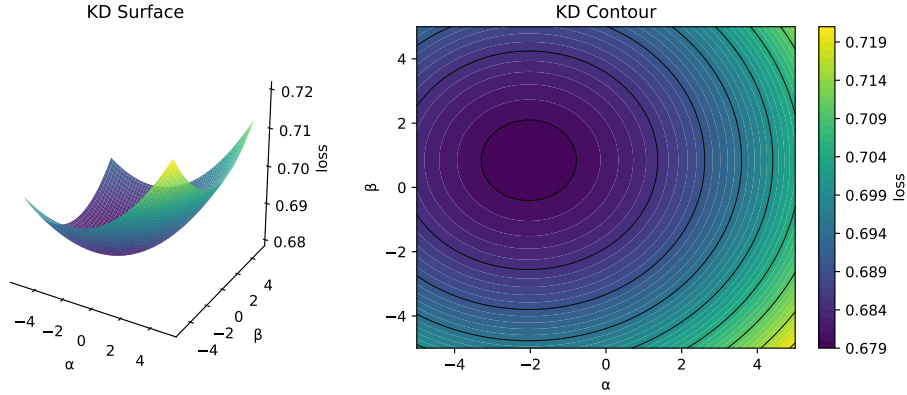
56 logits

$$s(\alpha, \beta) = t + \alpha d_1 + \beta d_2$$

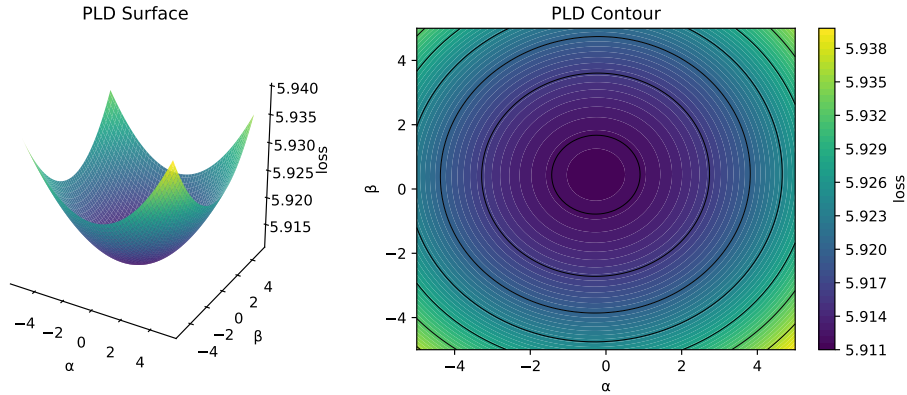
57 and compute each loss $L(s(\alpha, \beta), t)$ at every grid point. Figure 1 plots the loss landscapes for
 58 DIST [2], KD [1], and **PLD (ours)**. While KD and PLD both induce convex surfaces, DIST dips
 59 sharply yet remains effectively planar in the (α, β) slice. Moreover, the contour for PLD is more
 60 tightly centered around the origin than that of KD.



(a) DIST loss



(b) KD loss



(c) PLD loss

Figure 1: Loss landscapes of three distillation methods: (a) DIST exhibits a sharp dip yet remains effectively planar; (b) KD shows moderate convexity; (c) PLD (ours) exhibits better convexity with contours mostly centered at the origin.

6.1 Temperature Sensitivity of the PLD Loss Surface

To investigate the effect of the teacher-softmax temperature T on the geometry of the PLD loss landscape, we fix the same two-dimensional (α, β) slice and compute the PLD surface at four representative temperatures: $T \in \{2.0, 1.0, 0.5, 0.1\}$. Figure 2 shows both the 3D surface and 2D contour plots for each T . We observe that reducing T from 2.0 down to 1.0 produces only minor changes, whereas further lowering T below 1.0 flattens the curvature.

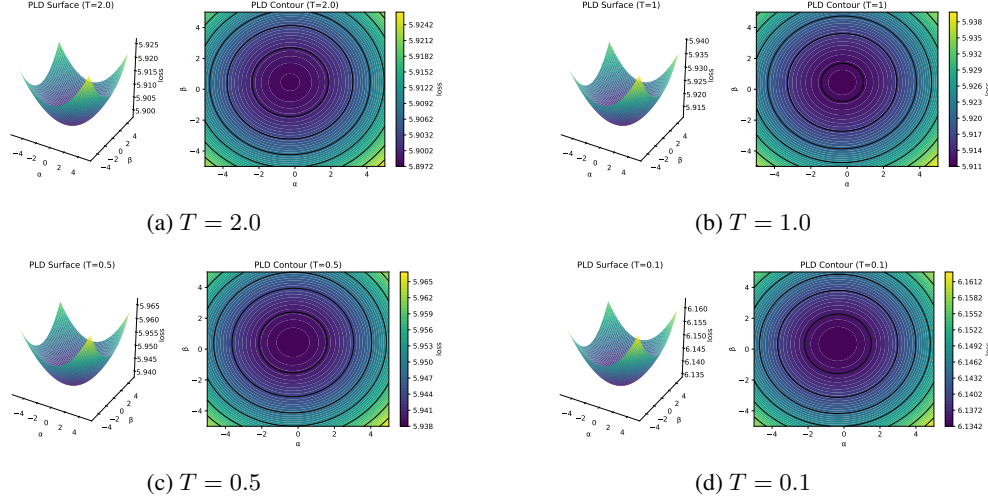


Figure 2: PLD loss surfaces at different teacher temperatures. (Top row) $T = 2.0$ and $T = 1.0$; (Bottom row) $T = 0.5$ and $T = 0.1$. Lowering T below 1.0 flattens convexity.

References

- [1] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [2] T. Huang, S. You, F. Wang, C. Qian, and C. Xu. Knowledge distillation from a stronger teacher. *NeurIPS*, 2022.
- [3] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [4] D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, W. Wang, C. Banbury, C. Ye, B. Akin, et al. Mobilenetv4: universal models for the mobile ecosystem. In *European Conference on Computer Vision*, pages 78–96. Springer, 2024.
- [5] R. Wightman. Pytorch image models. <https://github.com/huggingface/pytorch-image-models>, 2019.
- [6] X. Xie, P. Zhou, H. Li, Z. Lin, and S. Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [7] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.
- [8] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020.