NEURAL INFORMATION PROCESSING SYSTEMS

北京大学 PEKING UNIVERSITY

PLD: A Choice-Theoretic List-Wise Knowledge Distillation

Ejafa Bassam Dawei Zhu Kaigui Bian

School of Computer Science
Peking University

Knowledge Distillation

Goal: Teach a student model using a teacher model.

Both models take the same input x and produce logits or predictions.

We train the student using:

- Classification loss (e.g., Cross Entropy on the true label)
- Distillation loss (to mimic the teacher's behavior)

Together:

$$L = \alpha L_{CE} + (1 - \alpha) L_{KD}$$

Knowledge Distillation

Goal: Teach a student model using a teacher model.

Both models take the same input x and produce logits or predictions.

We train the student using:

- Classification loss (e.g., Cross Entropy on the true label)
- Distillation loss (to mimic the teacher's behavior)

Together:

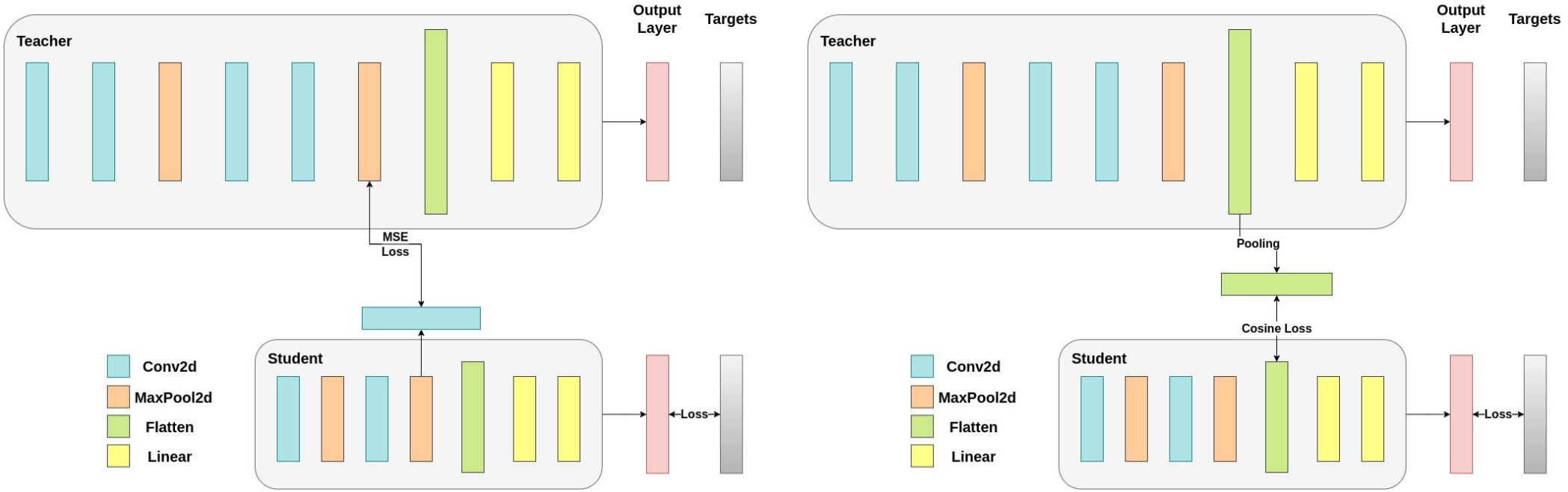
$$L = \alpha L_{CE} + (1 - \alpha) L_{KD}$$

Types of Distillation

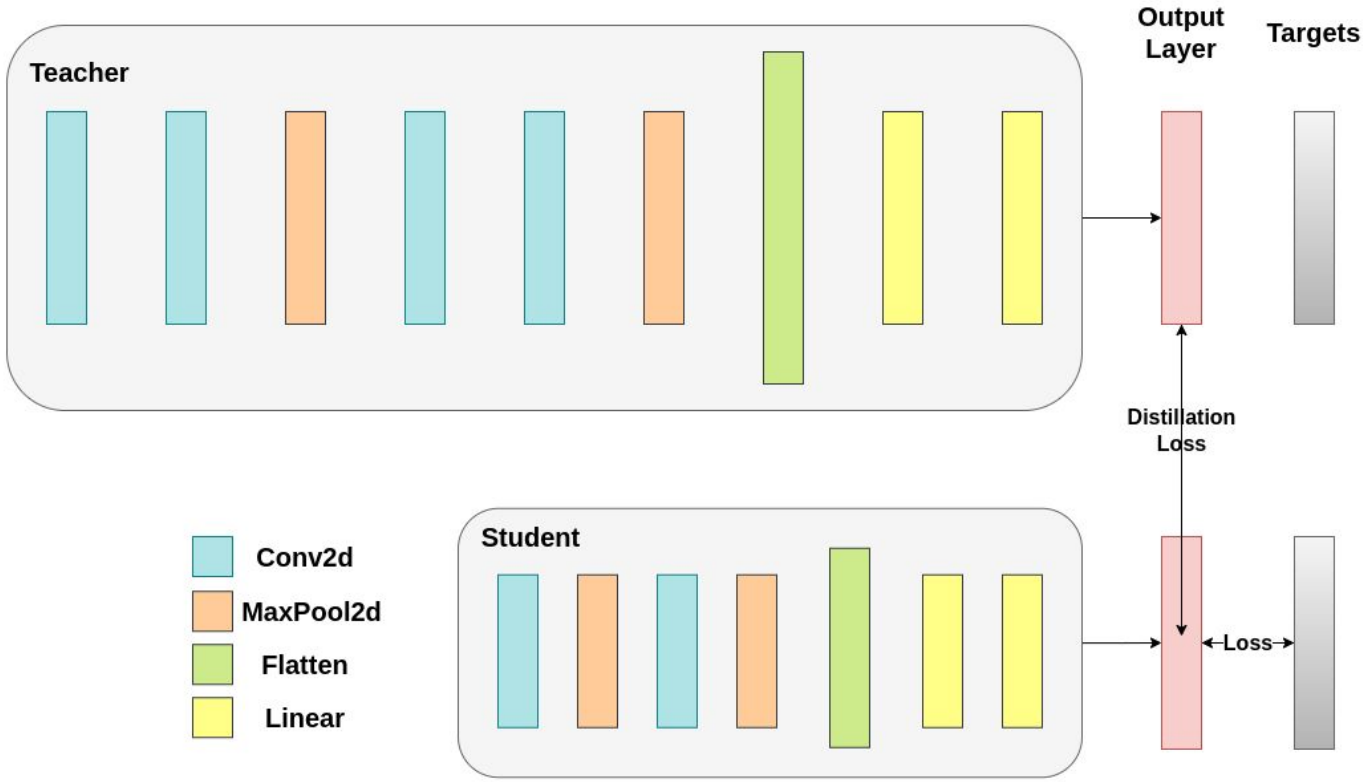
Two main kinds:

1. **Feature distillation:** student matches intermediate feature maps.
2. **Logit distillation:** student matches final logits (output probabilities).

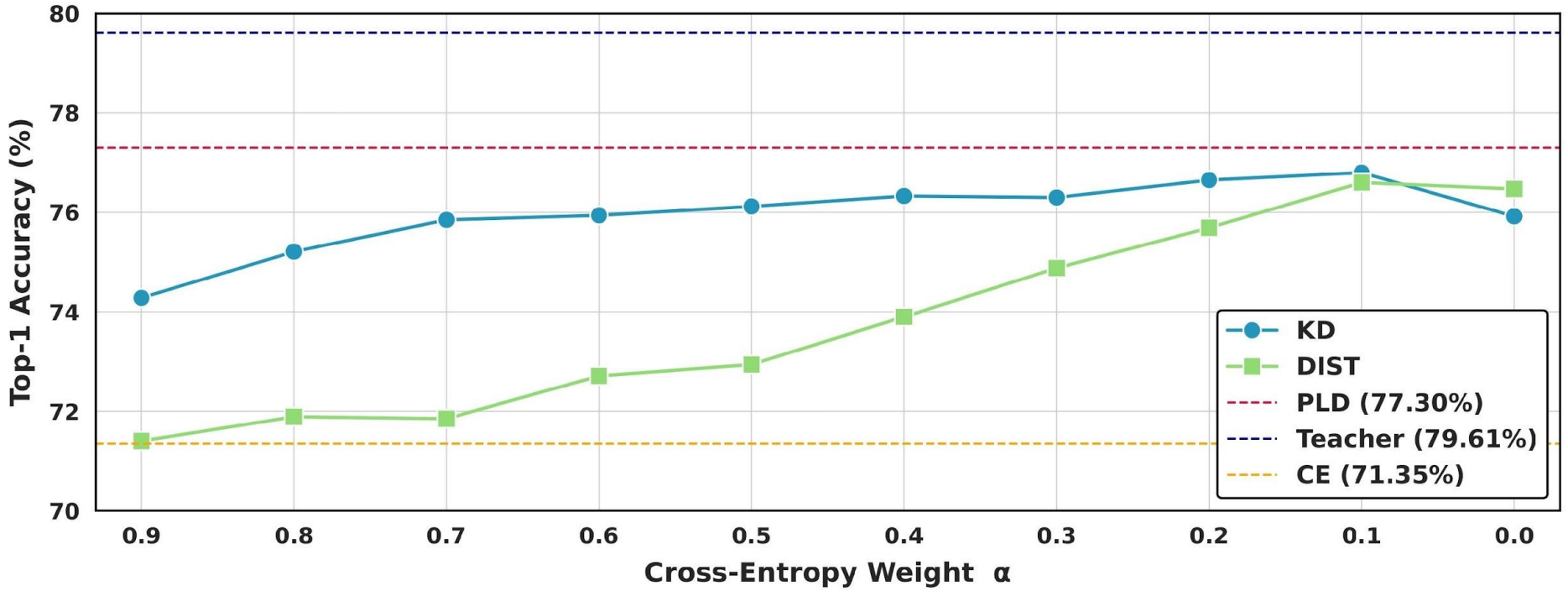
Feature distillation



Logit distillation

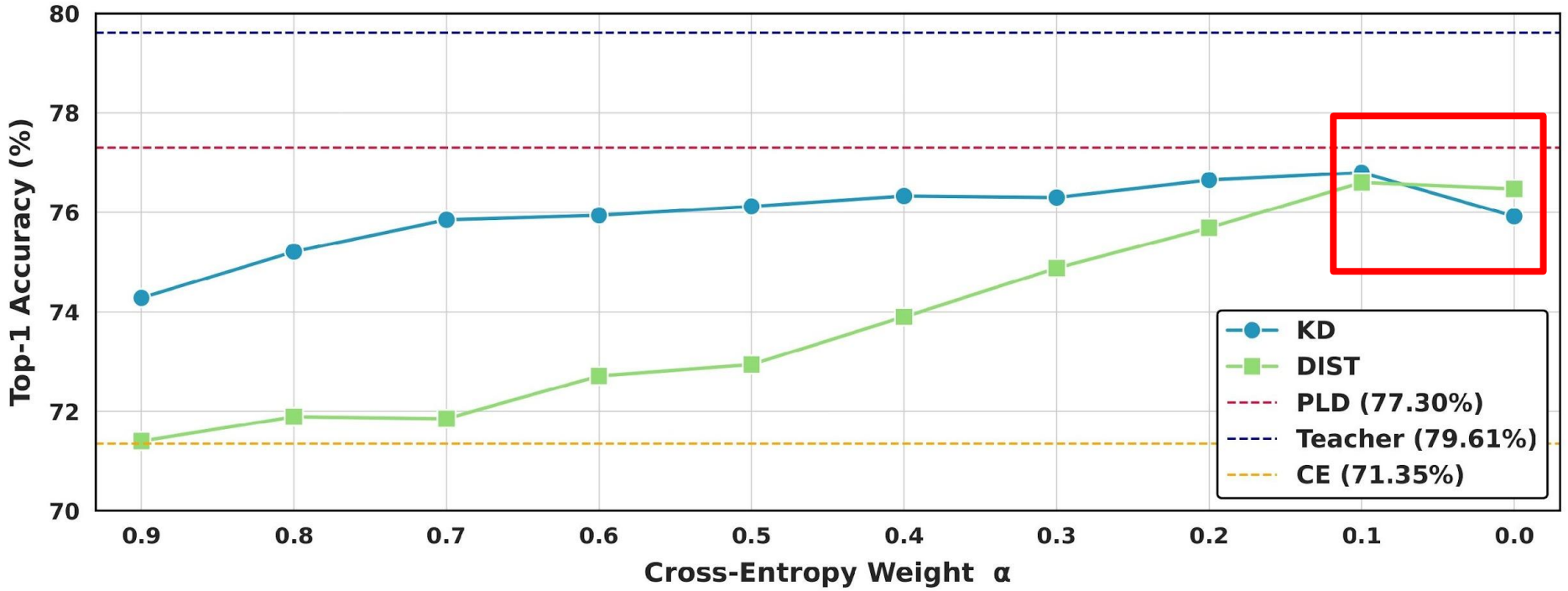


Challenge: Balancing the Two Losses



$$L = \alpha L_{CE} + (1 - \alpha) L_{KD}$$

Challenge: Balancing the Two Losses



$$L = \alpha L_{CE} + (1 - \alpha) L_{KD}$$

PLD: Plackett–Luce Distillation

STEP 1 – TEACHER-OPTIMAL PERMUTATION

1. Take teacher logits t and student logits s.
2. Remove the logit corresponding to the true label y.
3. Sort the teacher logits t in descending order.
4. Re-index the student logits s using that teacher order.
5. Prepend the true-label logit to the front.

STEP 2 – PLD CALCULATION

$$\pi^* = \big(y,\ \mathrm{argsort}(t) \setminus \{y\}\big)$$

$$P_{\mathrm{PL}}(\pi \mid s) = \prod_{k=1}^{C} \frac{w_{\pi_k}}{\sum_{l=k}^{C} w_{\pi_l}} = \prod_{k=1}^{C} \frac{\exp(s_{\pi_k})}{\sum_{l=k}^{C} \exp(s_{\pi_l})}$$

$$\mathcal{L}_{\mathrm{unweighted}}(s;\pi^*) = -\log P_{\mathrm{PL}}\big(\pi^* \mid s\big) = -\sum_{k=1}^{C} \log \frac{\exp\big(s_{\pi_k^*}\big)}{\sum_{\ell=k}^{C} \exp\big(s_{\pi_\ell^*}\big)}$$

$$q_i^T = \frac{\exp(t_i/\tau)}{\sum_{j=1}^{C} \exp(t_j/\tau)}, \alpha_k = q_{\pi_k^*}^T$$

$$\mathcal{L}_{\mathrm{PLD}}(s,t;y) = \sum_{k=1}^{C} q_{\pi_k^*}^T \Big[-s_{\pi_k^*} + \log \sum_{\ell=k}^{C} e^{s_{\pi_\ell^*}} \Big]$$

PLD: Plackett–Luce Distillation

STEP 1 – TEACHER-OPTIMAL PERMUTATION

1. Take teacher logits t and student logits s.
2. Remove the logit corresponding to the true label y.
3. Sort the teacher logits t in descending order.
4. Re-index the student logits s using that teacher order.
5. Prepend the true-label logit to the front.

STEP 2 – PLD CALCULATION

$$\pi^* = (y, \operatorname{argsort}(t) \setminus \{y\})$$

$$P_{\mathrm{PL}}(\pi \mid s) = \prod_{k=1}^{C} \frac{w_{\pi_k}}{\sum_{l=k}^{C} w_{\pi_l}} = \prod_{k=1}^{C} \frac{\exp(s_{\pi_k})}{\sum_{l=k}^{C} \exp(s_{\pi_l})}$$

$$\mathcal{L}_{\mathrm{unweighted}}(s;\pi^*) = -\log P_{\mathrm{PL}}(\pi^* \mid s) = -\sum_{k=1}^{C} \log \frac{\exp(s_{\pi_k^*})}{\sum_{\ell=k}^{C} \exp(s_{\pi_\ell^*})}$$

$$q_i^T = \frac{\exp(t_i/\tau)}{\sum_{j=1}^{C} \exp(t_j/\tau)}, \alpha_k = q_{\pi_k^*}^T$$

$$\mathcal{L}_{\mathrm{PLD}}(s,t;y) = \sum_{k=1}^{C} q_{\pi_k^*}^T \left[-s_{\pi_k^*} + \log \sum_{\ell=k}^{C} e^{s_{\pi_\ell^*}} \right]$$

PLD: Plackett–Luce Distillation

STEP 1 – TEACHER-OPTIMAL PERMUTATION

1. Take teacher logits t and student logits s.
2. Remove the logit corresponding to the true label y.
3. Sort the teacher logits t in descending order.
4. Re-index the student logits s using that teacher order.
5. Prepend the true-label logit to the front.

STEP 2 – PLD CALCULATION

$$\pi^* = \big(y,\ \mathrm{argsort}(t) \setminus \{y\}\big)$$

$$P_{\mathrm{PL}}(\pi \mid s) = \prod_{k=1}^{C} \frac{w_{\pi_k}}{\sum_{l=k}^{C} w_{\pi_l}} = \prod_{k=1}^{C} \frac{\exp(s_{\pi_k})}{\sum_{l=k}^{C} \exp(s_{\pi_l})}$$

$$\mathcal{L}_{\mathrm{unweighted}}(s;\pi^*) = -\log P_{\mathrm{PL}}\big(\pi^* \mid s\big) = -\sum_{k=1}^{C} \log \frac{\exp\big(s_{\pi_k^*}\big)}{\sum_{\ell=k}^{C} \exp\big(s_{\pi_\ell^*}\big)}$$

$$q_i^T = \frac{\exp(t_i/\tau)}{\sum_{j=1}^{C} \exp(t_j/\tau)}, \alpha_k = q_{\pi_k^*}^T$$

$$\mathcal{L}_{\mathrm{PLD}}(s,t;y) = \sum_{k=1}^{C} q_{\pi_k^*}^T \Big[-s_{\pi_k^*} + \log \sum_{\ell=k}^{C} e^{s_{\pi_\ell^*}}\Big]$$

Toy Example: Five-Class Illustration

We consider five classes (a, b, c, d, e).
Teacher logits: $(t_a, t_b, t_c, t_d, t_e)$.
Student logits: $(s_a, s_b, s_c, s_d, s_e)$.
The true label is (c).

Teacher ranking

For input x, the teacher's logits induce the permutation:

$$t_c \succ t_e \succ t_b \succ t_a \succ t_d.$$

Hence, the **teacher-optimal permutation:**

$$\pi^* = (c, e, b, a, d)$$

and the corresponding teacher softmax probabilities:

$$p_i^T = \frac{\exp(t_i/\tau)}{\sum_{j\in\{a,b,c,d,e\}} \exp(t_j/\tau)}, \qquad p_c^T \succ p_e^T \succ p_b^T \succ p_a^T \succ p_d^T.$$

Toy Example: Five-Class Illustration

We consider five classes (a, b, c, d, e).
Teacher logits: $(t_a, t_b, t_c, t_d, t_e)$.
Student logits: $(s_a, s_b, s_c, s_d, s_e)$.
The true label is (c).

Teacher ranking

For input x, the teacher's logits induce the permutation:

$$t_c \succ t_e \succ t_b \succ t_a \succ t_d.$$

Hence, the **teacher-optimal permutation:**

$$\pi^* = (c, e, b, a, d)$$

and the corresponding teacher softmax probabilities:

$$p_i^T = \frac{\exp(t_i/\tau)}{\sum_{j\in\{a,b,c,d,e\}} \exp(t_j/\tau)}, \qquad p_c^T \succ p_e^T \succ p_b^T \succ p_a^T \succ p_d^T.$$

Toy Example: Five-Class Illustration

We consider five classes (a, b, c, d, e).
Teacher logits: $(t_a, t_b, t_c, t_d, t_e)$.
Student logits: $(s_a, s_b, s_c, s_d, s_e)$.
The true label is (c).

Teacher ranking

For input x, the teacher's logits induce the permutation:

$$t_c \succ t_e \succ t_b \succ t_a \succ t_d.$$

Hence, the **teacher-optimal permutation:**

$$\pi^* = (c, e, b, a, d)$$

and the corresponding teacher softmax probabilities:

$$p_i^T = \frac{\exp(t_i/\tau)}{\sum_{j\in\{a,b,c,d,e\}} \exp(t_j/\tau)}, \qquad p_c^T \succ p_e^T \succ p_b^T \succ p_a^T \succ p_d^T.$$

Toy Example: Five-Class Illustration

We consider five classes (a, b, c, d, e).
Teacher logits: $(t_a, t_b, t_c, t_d, t_e)$.
Student logits: $(s_a, s_b, s_c, s_d, s_e)$.
The true label is (c).

Teacher ranking

For input x, the teacher's logits induce the permutation:

$$t_c \succ t_e \succ t_b \succ t_a \succ t_d.$$

Hence, the **teacher-optimal permutation:**

$$\pi^* = (c, e, b, a, d)$$

and the corresponding teacher softmax probabilities:

$$p_i^T = \frac{\exp(t_i/\tau)}{\sum_{j\in\{a,b,c,d,e\}} \exp(t_j/\tau)}, \qquad p_c^T \succ p_e^T \succ p_b^T \succ p_a^T \succ p_d^T.$$

Toy Example: Five-Class Illustration

Student Plackett-Luce ranking

For the student logits s, the Plackett–Luce probability of producing the same ranking π^* is:

$$P(\pi^*|s) = \frac{e^{s_c}}{e^{s_a}+e^{s_b}+e^{s_c}+e^{s_d}+e^{s_e}} \times \frac{e^{s_e}}{e^{s_e}+e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_b}}{e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_a}}{e^{s_a}+e^{s_d}} \times \frac{e^{s_d}}{e^{s_d}}.$$

The **negative log-likelihood** becomes:

$$\mathcal{L}_{PL}(s;\pi^*) = -\Big[\log \frac{e^{s_c}}{\sum_i e^{s_i}} + \log \frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}} + \log \frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}} + \log \frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}} + \log \frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\Big].$$

The first term is exactly the student's cross-entropy $(-\log p_c^S)$;
the later terms act as **distillation constraints**, encouraging: $s_c \succ s_e \succ s_b \succ s_a \succ s_d$.

Toy Example: Five-Class Illustration

Student Plackett-Luce ranking

For the student logits s, the Plackett–Luce probability of producing the same ranking π^* is:

$$P(\pi^*|s) = \frac{e^{s_c}}{e^{s_a}+e^{s_b}+e^{s_c}+e^{s_d}+e^{s_e}} \times \frac{e^{s_e}}{e^{s_e}+e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_b}}{e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_a}}{e^{s_a}+e^{s_d}} \times \frac{e^{s_d}}{e^{s_d}}.$$

The **negative log-likelihood** becomes:

$$\mathcal{L}_{PL}(s;\pi^*) = -\Big[\log \frac{e^{s_c}}{\sum_i e^{s_i}} + \log \frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}} + \log \frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}} + \log \frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}} + \log \frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\Big].$$

The first term is exactly the student's cross-entropy $(-\log p_c^S)$;
the later terms act as **distillation constraints**, encouraging: $s_c \succ s_e \succ s_b \succ s_a \succ s_d$.

Toy Example: Five-Class Illustration

Student Plackett-Luce ranking

For the student logits s, the Plackett–Luce probability of producing the same ranking π^* is:

$$P(\pi^*|s) = \frac{e^{s_c}}{e^{s_a}+e^{s_b}+e^{s_c}+e^{s_d}+e^{s_e}} \times \frac{e^{s_e}}{e^{s_e}+e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_b}}{e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_a}}{e^{s_a}+e^{s_d}} \times \frac{e^{s_d}}{e^{s_d}}.$$

The **negative log-likelihood** becomes:

$$\mathcal{L}_{PL}(s;\pi^*) = -\left[\log \frac{e^{s_c}}{\sum_i e^{s_i}} + \log \frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}} + \log \frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}} + \log \frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}} + \log \frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\right].$$

The first term is exactly the student's cross-entropy $(-\log p_c^S)$;
the later terms act as **distillation constraints**, encouraging: $s_c \succ s_e \succ s_b \succ s_a \succ s_d$.

Toy Example: Five-Class Illustration

Student Plackett-Luce ranking

For the student logits s, the Plackett–Luce probability of producing the same ranking π^* is:

$$P(\pi^*|s) = \frac{e^{s_c}}{e^{s_a}+e^{s_b}+e^{s_c}+e^{s_d}+e^{s_e}} \times \frac{e^{s_e}}{e^{s_e}+e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_b}}{e^{s_b}+e^{s_a}+e^{s_d}} \times \frac{e^{s_a}}{e^{s_a}+e^{s_d}} \times \frac{e^{s_d}}{e^{s_d}}.$$

The **negative log-likelihood** becomes:

$$\mathcal{L}_{PL}(s;\pi^*) = -\Big[\log \frac{e^{s_c}}{\sum_i e^{s_i}} + \log \frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}} + \log \frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}} + \log \frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}} + \log \frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\Big].$$

The first term is exactly the student's cross-entropy $(-\log p_c^S)$;
the later terms act as **distillation constraints**, encouraging: $s_c \succ s_e \succ s_b \succ s_a \succ s_d$.

Toy Example: Five-Class Illustration

From ListMLE → P-ListMLE → PLD

ListMLE: all ranking terms weighted equally.

P-ListMLE: multiply each rank by handcrafted decreasing weights (e.g., 2^{n-k}).

PLD: use the teacher's softmax $p^T_{\pi^*_k}$ as natural, data-dependent weights.

$$\boxed{\mathcal{L}_{PLD}(s,t;y) = -\sum_{k=1}^{K} p^T_{\pi^*_k}\left[s_{\pi^*_k} - \log\sum_{\ell\succeq k} e^{s_{\pi^*_\ell}}\right]}$$

Example (five classes):

$$\mathcal{L}_{PLD} = p^T_c\left(-\log\frac{e^{s_c}}{\sum_i e^{s_i}}\right) + p^T_e\left(-\log\frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}}\right) + p^T_b\left(-\log\frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}}\right) + p^T_a\left(-\log\frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}}\right) + p^T_d\left(-\log\frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\right).$$

Thus, PLD naturally merges the **cross-entropy** and **list-wise distillation** components into a single, convex objective that aligns the student's ranking:

$$s_c \succ s_e \succ s_b \succ s_a \succ s_d$$

Toy Example: Five-Class Illustration

From ListMLE → P-ListMLE → PLD

ListMLE: all ranking terms weighted equally.

P-ListMLE: multiply each rank by handcrafted decreasing weights (e.g., 2^{n-k}).

PLD: use the teacher's softmax $p^T_{\pi^*_k}$ as natural, data-dependent weights.

$$\mathcal{L}_{PLD}(s,t;y) = -\sum_{k=1}^{K} p^T_{\pi^*_k}\left[s_{\pi^*_k} - \log \sum_{\ell \succeq k} e^{s_{\pi^*_\ell}}\right]$$

Example (five classes):

$$\mathcal{L}_{PLD} = p^T_c\left(-\log \frac{e^{s_c}}{\sum_i e^{s_i}}\right) + p^T_e\left(-\log \frac{e^{s_e}}{\sum_{i\succeq e} e^{s_i}}\right) + p^T_b\left(-\log \frac{e^{s_b}}{\sum_{i\succeq b} e^{s_i}}\right) + p^T_a\left(-\log \frac{e^{s_a}}{\sum_{i\succeq a} e^{s_i}}\right) + p^T_d\left(-\log \frac{e^{s_d}}{\sum_{i\succeq d} e^{s_i}}\right)$$

Thus, PLD naturally merges the **cross-entropy** and **list-wise distillation** components into a single, convex objective that aligns the student's ranking:

$$s_c \succ s_e \succ s_b \succ s_a \succ s_d$$

Experiments (CIFAR-100)

Method	Homogeneous setup			Heterogeneous setup
	WRN-40-2 WRN-40-1	ResNet-56 ResNet-20	ResNet-32×4 ResNet-8×4	ResNet-50 MobileNetV2
Student (CE)	70.26±0.24	67.57±0.25	71.56±0.16	64.42±0.67
Feature-based methods				
AT [51]	71.99±0.34	68.42±0.16	72.51±0.37	52.28±0.84
FitNet [32]	70.86±0.33	66.75±0.27	72.59±0.30	62.77±0.05
PKT [27]	71.43±0.16	68.66±0.20	72.86±0.17	66.08±0.11
RKD [26]	11.32±1.84	68.53±0.18	71.45±0.41	65.16±0.21
SP [38]	73.49±0.20	69.36±0.27	72.45±0.08	65.30±0.29
VID [1]	70.61±0.39	68.52±0.12	71.70±0.32	64.14±0.82
Logits-based methods				
KD [12]	72.44±0.36	69.45±0.24	72.11±0.10	66.61±1.16
DIST [13]	72.74±0.33	69.66±0.49	73.21±0.10	66.98±0.74
DKD [52]	73.22±0.38	67.80±0.03	73.08±0.27	**69.51±0.42**
PLD (ours)	**73.50±0.30**	**70.28±0.21**	**73.97±0.05**	68.16±0.38

Experiments (ImageNet-1K)

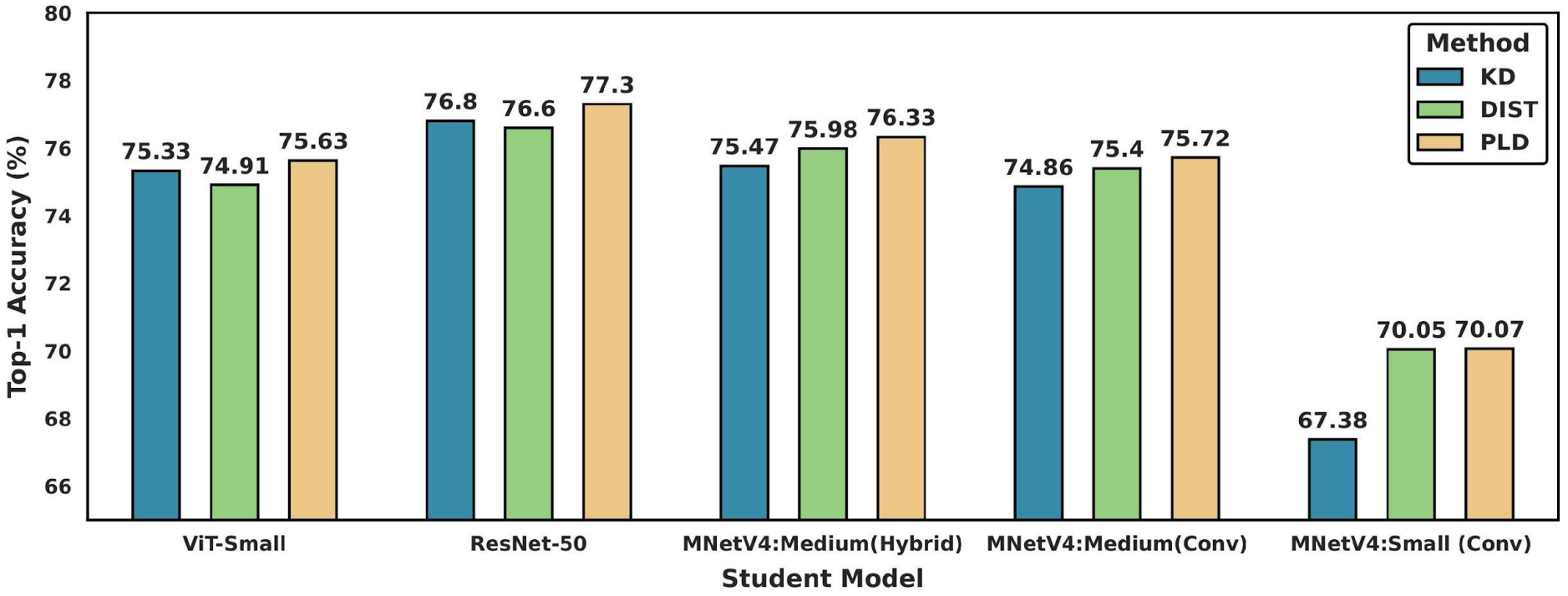
Method	Hyperparameters				Accuracy (%)	
	α	β	γ	τ	Top-1	Top-5
Teacher	–	–	–	–	79.61	–
CE	–	–	–	–	71.35	–
LS (ϵ=0.1)	–	–	–	–	73.92	–
KD hyperparameter sweep (τ=2)						
	0.00	–	–	2.00	75.92	92.82
KD	0.10	–	–	2.00	**76.80**	**93.16**
	0.20	–	–	2.00	76.65	93.11
	0.30	–	–	2.00	76.30	93.02
	0.40	–	–	2.00	76.33	93.10
	0.50	–	–	2.00	76.12	93.12
	0.60	–	–	2.00	75.94	92.79
	0.70	–	–	2.00	75.85	92.71
	0.80	–	–	2.00	75.21	92.39
	0.90	–	–	2.00	74.28	91.59

DIST hyperparameter sweep (τ=1)						
	1.00	2.00	2.00	1.00	75.77	92.67
	0.00	0.50	0.50	1.00	76.47	93.19
DIST	0.10	0.45	0.45	1.00	**76.60**	**93.30**
	0.20	0.40	0.40	1.00	75.69	92.63
	0.30	0.35	0.35	1.00	74.88	92.18
	0.40	0.30	0.30	1.00	73.90	91.35
	0.50	0.25	0.25	1.00	72.94	90.95
	0.60	0.20	0.20	1.00	72.71	90.58
	0.70	0.15	0.15	1.00	71.85	90.16
	0.80	0.10	0.10	1.00	71.89	90.16
	0.90	0.05	0.05	1.00	71.40	89.88

Experiments (ImageNet-1K)

Variant	τ	Top-1 Acc.%	Top-5 Acc.%
PLD	0.50	76.06	92.39
	1.00	**77.30**	93.28
	1.50	77.17	92.74
	2.00	76.46	92.34
	4.00	75.22	91.34
PLD–ListMLE	–	74.11	90.60
PLD–pListMLE	–	76.60	93.23

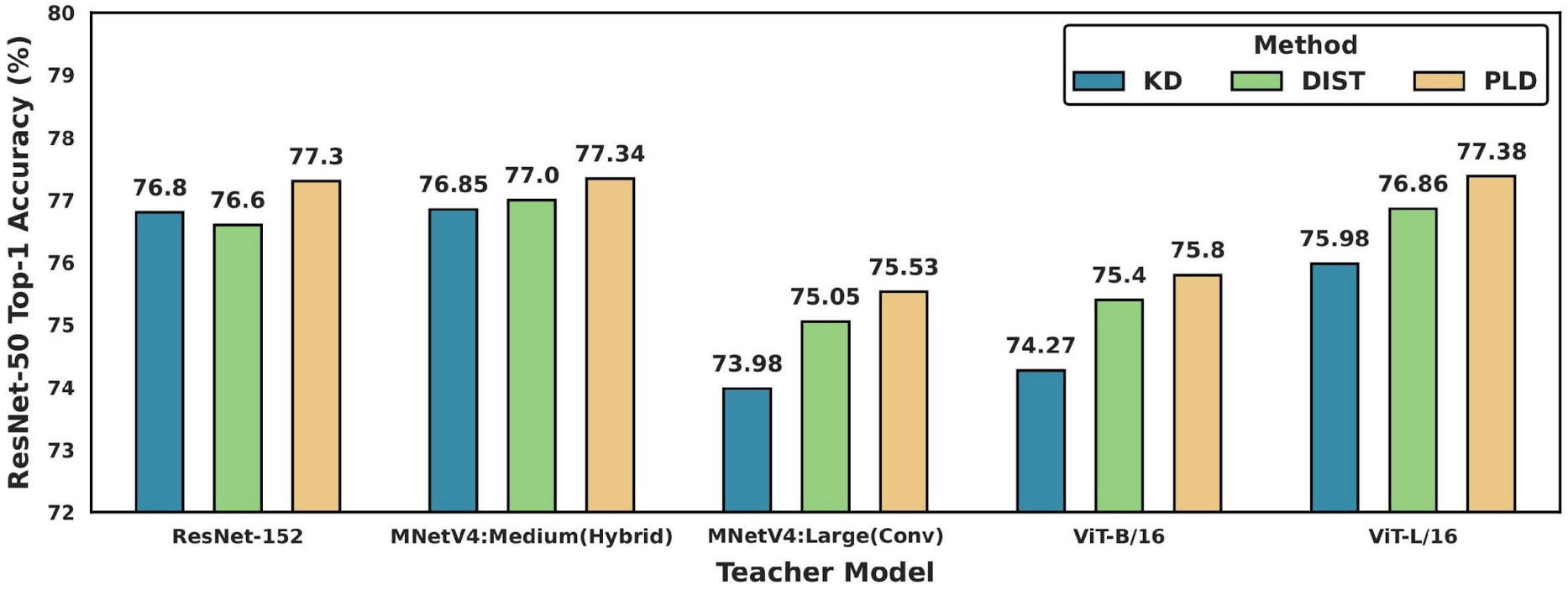
Experiments (ImageNet-1K)



Experiments (ImageNet-1K)

Student	Teacher	Params(M)	Top-1 % Accuracy					
			Teacher	KD	DIST	PLD	Δ_{DIST}	Δ_{KD}
ViT-Small	ViT-Large(304.33M)	22.05	84.80	75.33	74.91	**75.63**	**0.72**	**0.30**
ResNet-50	ResNet-152(60.19M)	25.56	79.61	76.80	76.60	**77.30**	**0.70**	**0.50**
MobileNet-v4								
Hybrid-Medium	Large (conv)	11.07	80.83	75.47	75.98	**76.33**	**0.35**	**0.86**
Medium (conv)	(32.59M)	9.72	80.83	74.86	75.40	**75.72**	**0.32**	**0.86**
Small (conv)		3.77	80.83	67.38	70.05	**70.07**	**0.02**	**2.69**

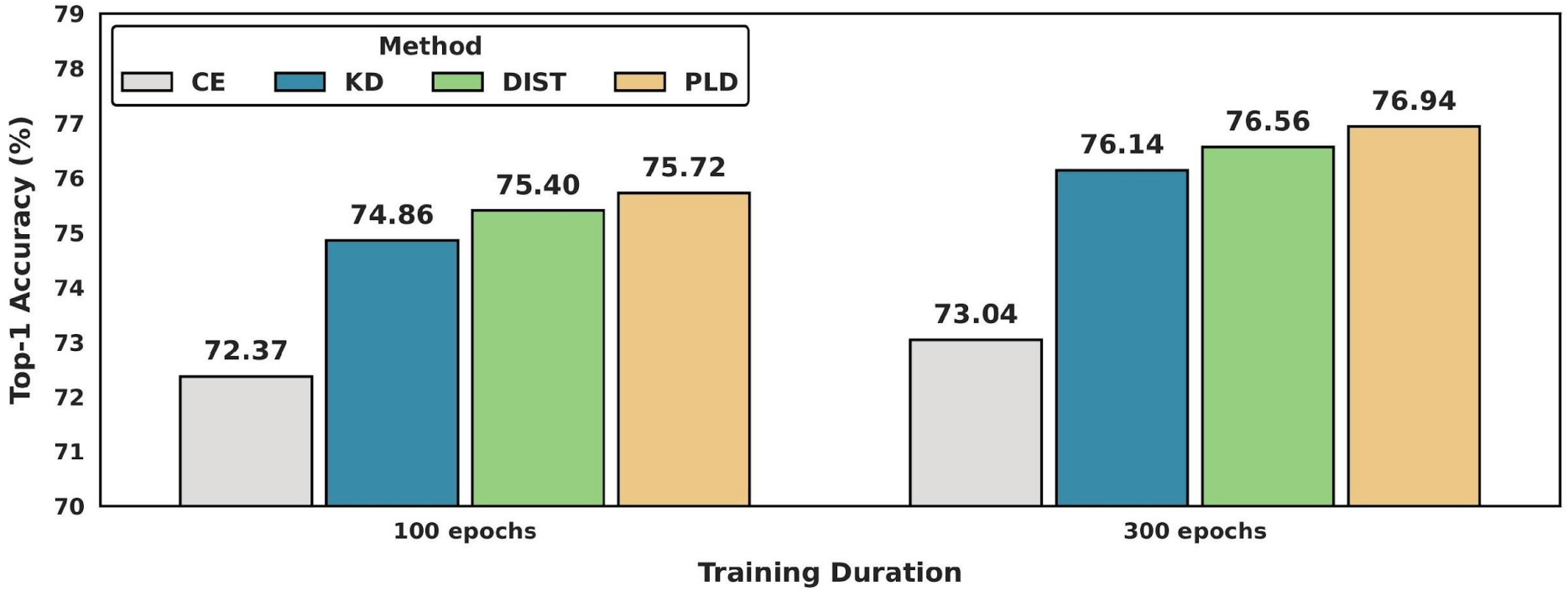
Experiments (ImageNet-1K)



Experiments (ImageNet-1K)

Teacher	Params (M)	Top-1 % Accuracy				Δ	
		Teacher	KD	DIST	PLD	Δ_{DIST}	Δ_{KD}
ResNet-152	60.19	79.61	76.80	76.60	**77.30**	**0.70**	**0.50**
MobileNet-v4 Hybrid Medium	11.07	78.66	76.85	77.00	**77.34**	**0.34**	**0.49**
MobileNet-v4 Conv Large	32.59	80.83	73.98	75.05	**75.53**	**0.48**	**1.55**
ViT-Base/16	86.57	82.07	74.27	75.40	**75.80**	**0.40**	**1.53**
ViT-Large/16	304.33	84.80	75.98	76.86	**77.38**	**0.52**	**1.40**

Experiments (ImageNet-1K)



Experiments (ImageNet-1K)

Epochs	CE	KD	DIST	PLD	Δ_{DIST}	Δ_{KD}
100 epochs	72.37	74.86	75.40	**75.72**	0.32	0.86
300 epochs	73.04	76.14	76.56	**76.94**	0.38	0.80
Δ (300–100)	0.67	1.28	1.16	1.22	–	–

Object detection (MS-COCO)

Teacher → Student	Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
ResNet-50 → MobileNetV2-FPN	DKD	29.24	50.42	29.87	16.15	31.11	38.07
ResNet-50 → MobileNetV2-FPN	PLD	28.91	49.63	**29.88**	16.05	30.41	**38.59**
ResNet-101 → ResNet-18-FPN	DKD	32.11	53.43	33.90	18.46	34.43	41.50
ResNet-101 → ResNet-18-FPN	PLD	**32.47**	**53.83**	**34.17**	18.50	**34.97**	**42.12**
ResNet-101 → ResNet-50-FPN	DKD	36.54	**58.57**	39.46	**21.74**	39.80	**47.31**
ResNet-101 → ResNet-50-FPN	PLD	**36.60**	58.28	**39.58**	21.37	39.76	47.24

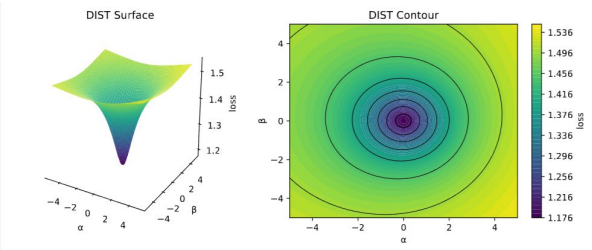
Optimizer Ablation Study (ImageNet-1K)

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**

Total training time (Cifar100)

Model (Teacher Student)	CE	PLD	KD	DIST	DKD	AT	FitNet	PKT	RKD	SP	VID
ResNet32×4 ResNet8×4	32.06	62.84	62.18	62.24	62.46	65.39	63.96	62.53	68.36	62.73	78.40
ResNet50 MobileNetV2	23.62	102.59	102.23	102.49	102.54	108.17	113.18	102.28	128.65	102.78	132.06
ResNet56 ResNet20	22.97	33.42	33.57	33.81	33.46	34.55	33.30	33.89	36.21	33.67	36.57
WideResNet-40-2 WideResNet-40-1	27.67	33.10	37.56	36.73	34.27	36.77	38.05	36.84	37.62	37.01	37.35

Loss landscapes & Implementation



(a) DIST loss

(b) KD loss

(c) PLD loss

C Implementation of the PLD Loss

```python
import torch
import torch.nn.functional as F

def create_adjusted_ranking(flat_logits: torch.Tensor,
                            flat_labels: torch.LongTensor
                           ) -> torch.LongTensor:
    """
    For each example, sort logits ascending, remove the true label,
    and append it at the end so it occupies the last (top-1) position.
    Returns a [N, V] LongTensor of class indices per rank.
    """
    _, sorted_idx = torch.sort(flat_logits, dim=-1, descending=False)#[N,V]
    mask = sorted_idx != flat_labels.unsqueeze(-1) #[N,V]
    V = flat_logits.size(-1)

    assert torch.all(mask.sum(dim=-1) == V - 1), \
        "Must remove exactly one true label per row."

    sorted_excl = sorted_idx[mask].view(-1, V - 1) #[N,V-1]
    return torch.cat([sorted_excl, flat_labels.unsqueeze(-1)], dim=-1)#[N,V]

def plackett_luce_loss(student_logits: torch.Tensor,
                       teacher_logits: torch.Tensor,
                       labels: torch.LongTensor,
                       temperature: float = 1.0
                      ) -> torch.Tensor:
    """
    Computes the PLD loss:
        L = sum_k alpha_k [ logsumexp_k - s_k ],
        alpha_k = softmax_teacher[pi*_k].
    """
    flat_s, flat_t, flat_lbl = prepare_for_classification(
        student_logits, labels, teacher_logits
    )
    ranking = create_adjusted_ranking(flat_t, flat_lbl) # [N, V]
    s_perm = torch.gather(flat_s, dim=-1, index=ranking) # [N, V]
    t_perm = torch.gather(flat_t, dim=-1, index=ranking) # [N, V]

    log_cumsum = torch.logcumsumexp(s_perm, dim=-1) # [N, V]
    per_pos_loss = log_cumsum - s_perm # [N, V]
    teacher_prob = F.softmax(t_perm / temperature, dim=-1) # [N, V]
    weighted = per_pos_loss * teacher_prob # [N, V]

    return weighted.sum(dim=-1).mean() # scalar
```

Listing 1: PyTorch implementation of Plackett-Luce Distillation (PLD) loss.

Conclusion

PLD = unified, convex, hyperparameter-light objective.

No α balancing needed.

Works across datasets and tasks.

Simple to implement ~ 40 lines of code.

Future Work: Extend to **NLP** and **reinforcement learning** domains.

Thanks for watching!

Ejafa Bassam
Email: ejafabassam@stu.pku.edu.cn
LinkedIn: ejafabassam