Gap Preserving Distillation by Building Bidirectional Mappings with A Dynamic Teacher

ICLR 2025

Yong Guo, Shulian Zhang, Haolin Pan, Jing Liu, Yulun Zhang, Jian Chen

Overview

Motivation:

- Knowledge distillation transfers knowledge from large teacher model to compact student.
- However, **a too large performance gap** between teacher and student hampers training.

💡 Idea & Method:

❖ We introduce a **dynamic teacher** model trained alonaside the student to maintain a reasonable performance gap.

❖ **Parameter sharing** between student and teacher for direct knowledge inheritance

❖ **Bidirectional mappings**:

🔄 Inverse Reparameterization (IR): Student → Teacher expansion

🔄 Channel-Branch Reparameterization (CBR): Teacher → Student extraction

📈 Results

- Consistently outperforms existing distillation methods (+1.58%)
- Generalizes well to training from scratch (+1.80%) and fine-tuning (+0.89%)

Method Overview

Dynamic teacher

Preserves appropriate performance gap during training

Bidirectional Mappings

- Inverse Reparameterization (IR):

Expands student to create dynamic teacher without sacrificing accuracy

- Channel-Branch Reparameteriztion(CBR):

Extracts effective student from dynamic teacher

Parameter Sharing

- Enables more direct knowledge inheritance

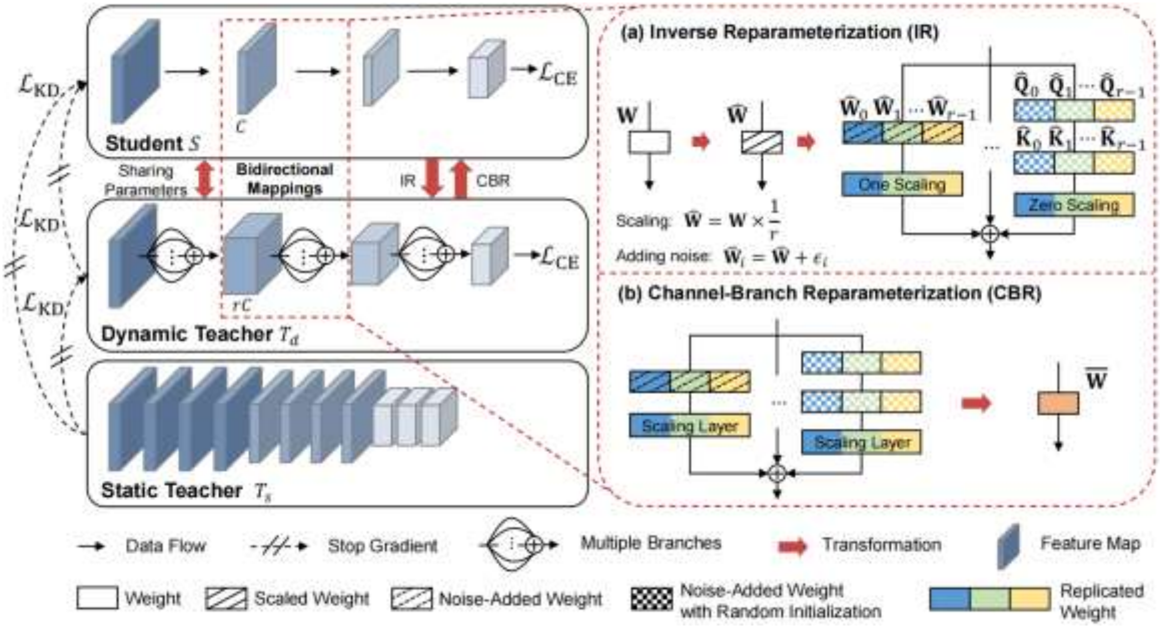

Fig. Overview of the proposed Gap Preserving Distillation (GPD) method.

Build Dynamic Teacher via Inverse Reparameterization

IR expands the student model along two dimensions while preserving accuracy

➢ **Channel-level:**
- Replicate weights and applies scaling to compensate for the increased number of channels
- Introduces noise for breaking symmetry between replications

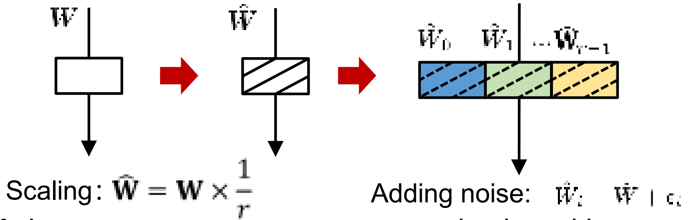

Fig. Illustration of channel-level inverse reparameterization with an expansion ratio of r.

➢ **Branch-level:**
- Transforms single conv layer into multi-branch structure
- Zeros-out additional branches initially to preserve original output

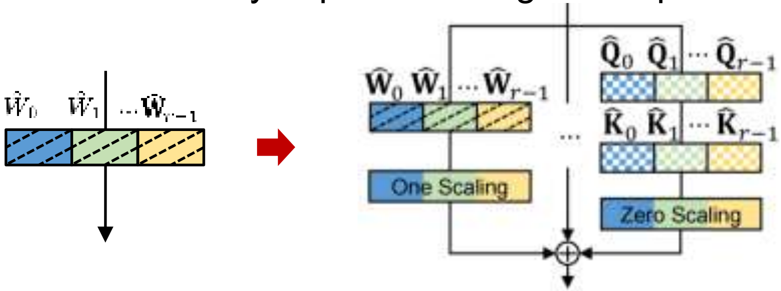

Fig. Illustration of branch-level inverse reparameterization.

Parameter Sharing via Channel-Branch Reparameterization

Student and dynamic teacher share parameters for direct knowledge inheritance

- Student conducts online reparameterization
- teacher performs direct forward pass

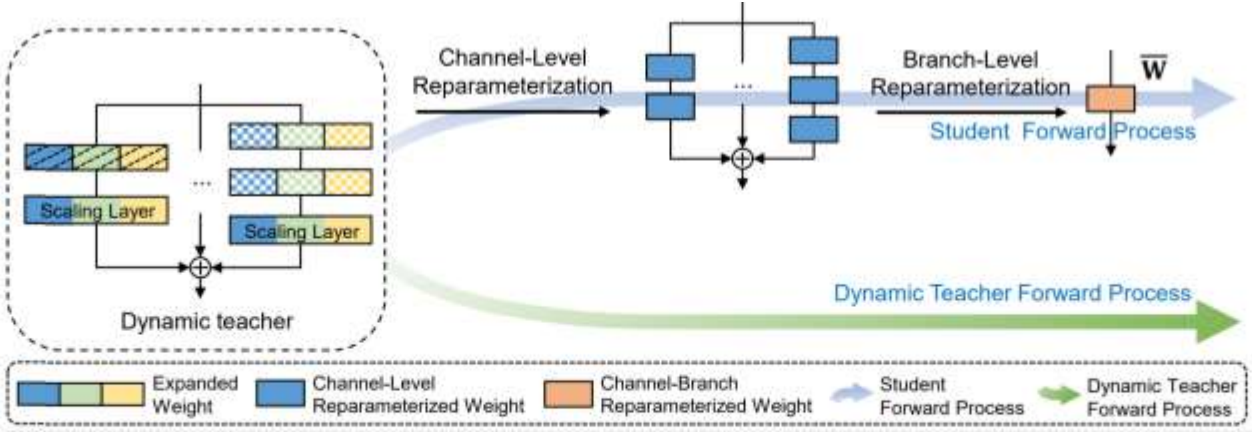

Fig. Illustration of the forward process for the student and dynamic teacher models.

Parameter Sharing via Channel-Branch Reparameterization

- **Channel-Level Reparameterization**

  Extract a channel-wise slice from $\mathbf{W}_m^l$ and apply a scaling operation

$$\bar{\mathbf{W}}_m^l = r\mathbf{W}_m^l[:C_m^l, :C_m^{l-1}, :, :]$$

- **Branch-Level Reparameterization**

  a) Reparameterize each branch:

$$\bar{\mathbf{W}}_m = \bar{\mathbf{W}}_m^1 \bar{\mathbf{W}}_m^2 \dots \bar{\mathbf{W}}_m^{L_m}$$

  b) Sum up all branches:

$$\bar{\mathbf{W}} = \sum_{m=1}^{M} \bar{\mathbf{W}}_m$$

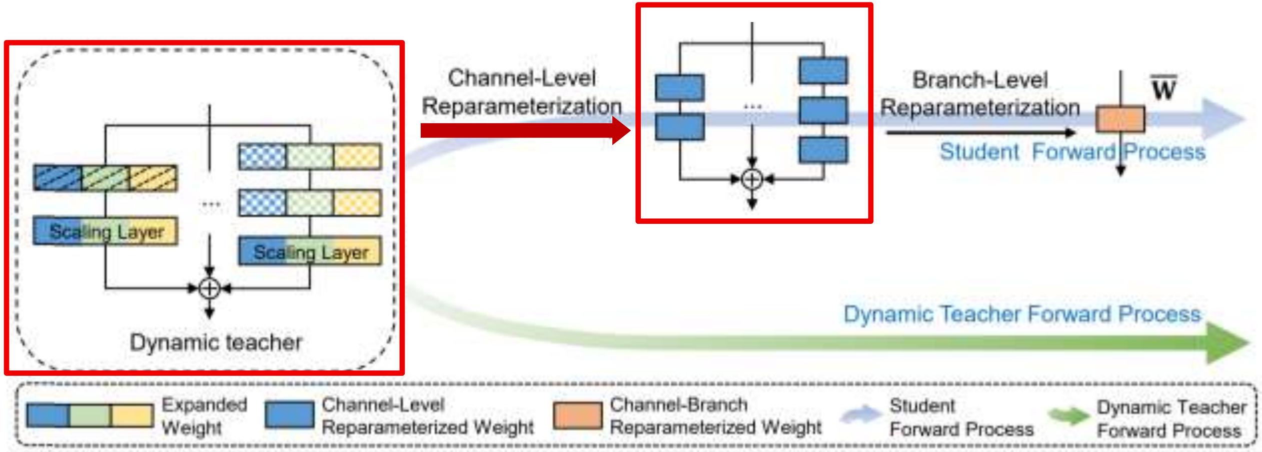

Fig. Illustration of the forward process for the student and dynamic teacher models.

Parameter Sharing via Channel-Branch Reparameterization

➢ **Channel-Level Reparameterization**

Extract a channel-wise slice from $\mathbf{W}_m^l$ and apply a scaling operation

$$\bar{\mathbf{W}}_m^l = r\mathbf{W}_m^l[:C_m^l, :C_m^{l-1}, :, :]$$

➢ **Branch-Level Reparameterization**

a) Reparameterize each branch:

$$\bar{\mathbf{W}}_m = \bar{\mathbf{W}}_m^1 \bar{\mathbf{W}}_m^2 \dots \bar{\mathbf{W}}_m^{L_m}$$

b) Sum up all branches:

$$\bar{\mathbf{W}} = \sum_{m=1}^{M} \bar{\mathbf{W}}_m$$

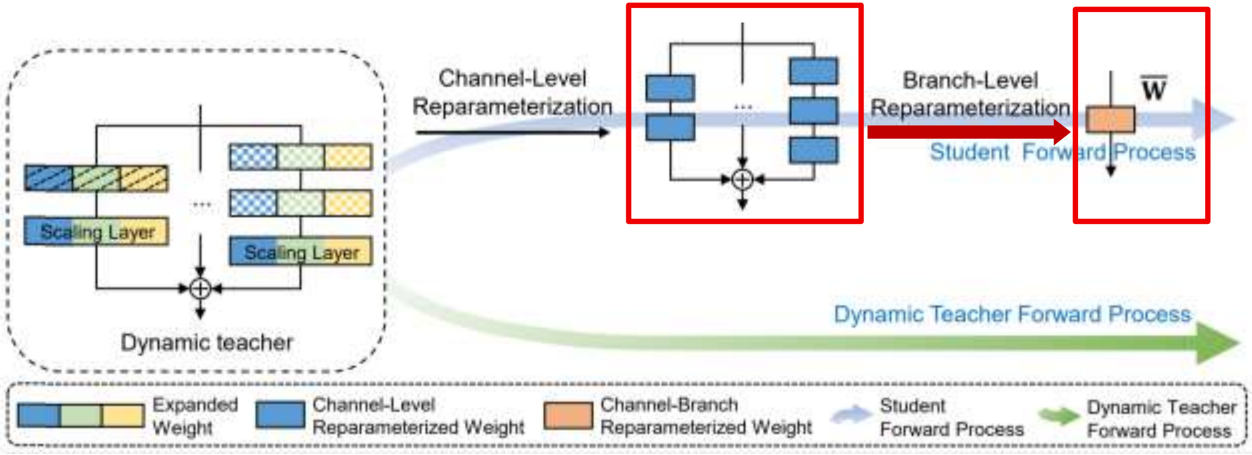

Fig. Illustration of the forward process for the student and dynamic teacher models.

Training Objective

- Total loss combines standard KD loss with GPD-specific loss:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{CE}}(S(x), y) + \lambda \mathcal{L}_{\text{KD}}(\psi(S(x)), \psi(T_s(x)))}_{\text{standard KD objective}} + \mathcal{L}_{\text{GPD}}$$

- GPD loss enables three-way knowledge transfer:

1. Train dynamic teacher with cross-entropy
2. Dynamic teacher guides student
3. Static teacher guides dynamic teacher

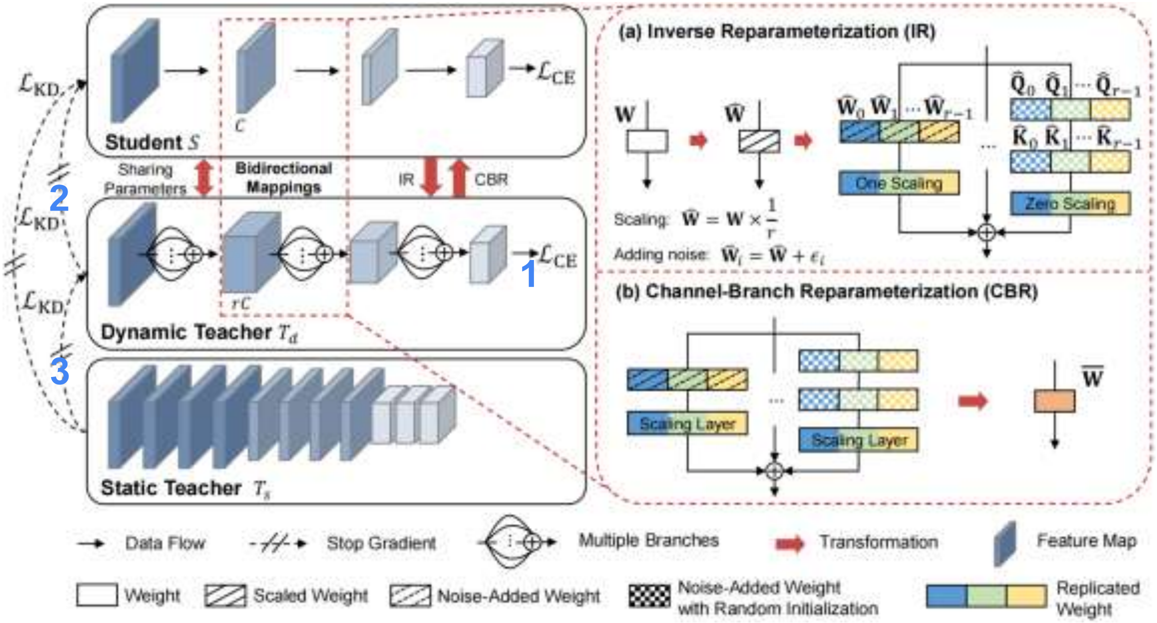

Fig. Overview of the proposed Gap Preserving Distillation (GPD) method.

Distillation with Static Teacher

GPD consistently improves performance across different architectures

Model	Teacher → Student		
	ResNet34 → ResNet18	ResNet50 → MobileNet	RVT-S → RVT-Ti
Teacher	73.31	76.16	81.69
Student	69.75	68.87	78.45
KD (Hinton et al., 2015)	70.66	68.58	-
AT (Zagoruyko & Komodakis, 2017a)	70.69	69.56	-
OFD (Heo et al., 2019a)	70.81	71.25	-
CRD (Tian et al., 2020)	71.17	71.37	-
RKD (Park et al., 2019)	70.40	68.5	-
WSLD (Zhou et al., 2021)	72.04	71.52	-
SRRL (Yang et al., 2021)	71.73	72.49	-
SimKD (Chen et al., 2022)	71.59	72.25	-
DIST (Huang et al., 2022)	72.07	73.24	-
NKD (Yang et al., 2023)	71.96	72.58	-
CAT-KD (Guo et al., 2023)	71.26	72.24	-
KD+CTKD (Li et al., 2023)	71.38	71.16	-
MLKD (Jin et al., 2023)	71.90	73.01	-
KD+CTKD+LS (Sun et al., 2024)	71.81	72.92	-
DKD+LSKD (Sun et al., 2024)	71.88	72.85	-
MLKD+LSKD (Sun et al., 2024)	72.08	73.22	-
CKD (Zhu et al., 2024b)	72.24	72.97	-
ReviewKD (Chen et al., 2021b)	71.61	72.56	78.92
ReviewKD + GPD	**72.50 (+0.89)**	**73.21 (+0.65)**	**80.01 (+1.09)**
DKD (Zhao et al., 2022a)	71.70	72.05	79.12
DKD + GPD	**72.71 (+1.01)**	**73.63 (+1.58)**	**80.14 (+1.02)**

Train from Scratch / Fine-Tuning

🏆 Even without using a pre-trained static teacher, GPD still generalizes well to various training scenarios

🔨 **Train from scratch**

Model	ResNet18	MobileNet	RVT-Ti
Baseline	70.07	71.68	78.45
GPD*	**71.87 (+1.80)**	**73.07 (+1.39)**	**79.85 (+1.40)**

🔧 **Model fine-tuning**

Model	ResNet18	MobileNet	RVT-Ti
Pre-trained Model	69.75	68.87	78.45
Longer Training	70.23	69.01	78.61
GPD*	**71.12 (+0.89)**	**69.47 (+0.46)**	**78.84 (+0.23)**

Thanks for your attention!