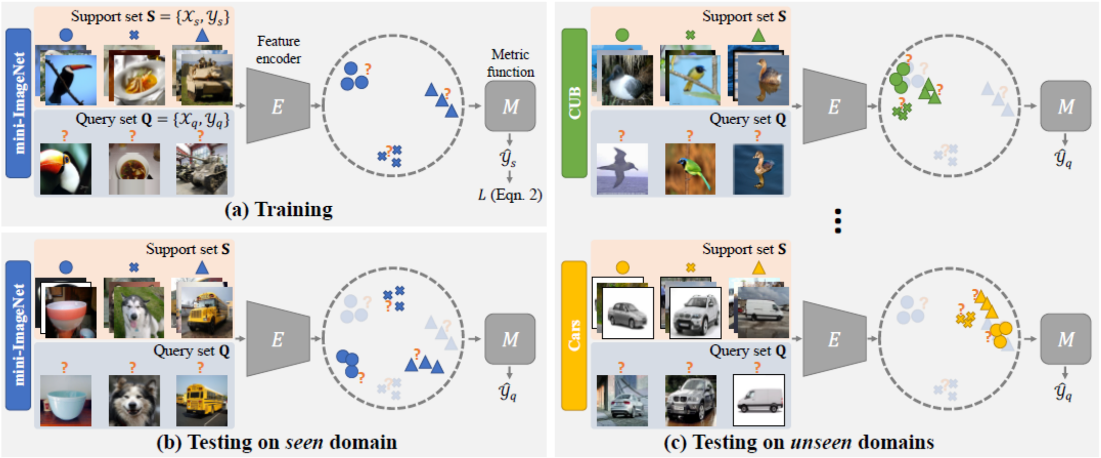
Cross-Domain Few-Shot Classification via Learned Feature-Wise Transformation

现有的基于度量的小样本分类算法通过使用学习度量函数将查询图像的特征嵌入与少数标记图像的特征嵌入进行比较来预测类别。由于域之间的特征分布存在很大的差异，这些方法往往不能推广到不可见的域。在这项工作中，作者主要解决基于度量的方法在域转移下的少样本分类问题。作者提出了一种Feature-Wise转换层，在训练阶段利用仿射变换增强图像的特征，模拟不同领域下的各种特征分布。为了捕获不同领域中特征分布的变化，我们进一步应用了一种元学习方法来搜索Feature-Wise转换层的超参数。作者使用5个小样本分类数据集:mini-ImageNet、CUB、Cars、Places和Plantae，在域迁移设置下进行了大量的实验和消融研究。实验结果表明，所提出的特征变换层适用于各种基于度量的模型，并对域转移下的小样本分类性能提供了一致的改进。


mini-ImageNet
Support set $\mathbf{S} = \{\mathcal{X}_s, \mathcal{Y}_s\}$
Query set $\mathbf{Q} = \{\mathcal{X}_q, \mathcal{Y}_q\}$
Feature encoder
E
Metric function
M
$\hat{y}_s$
L (Eqn. 2)
Support set S
Query set Q
$\hat{y}_q$
CUB
Cars

(a) Training

(b) Testing on *seen* domain

(c) Testing on *unseen* domains

Figure 1: **Problem formulation and motivation.** Metric-based meta-learning models usually consist of a feature encoder E and metric function M. We aim to improve the generalization ability of the models training from seen domains to arbitrary unseen domains. The key observation is that the distributions of the image features extracted from tasks in the unseen domains are significantly different from those in the seen domains.

方法

将一组包含小样本分类任务的域表示为 $\mathcal{T} = \{T_1, T_2, \cdots, T_n\}$

假设训练阶段有N个可见的域 $\{\mathcal{T}_1^{\text{seen}}, \mathcal{T}_2^{\text{seen}}, \cdots, \mathcal{T}_N^{\text{seen}}\}$

目标是学习一个基于度量的小样本分类模型，该模型可以很好地泛化到一个不可见的域 $\mathcal{T}^{\text{unseen}}$.

Feature-Wise转换层

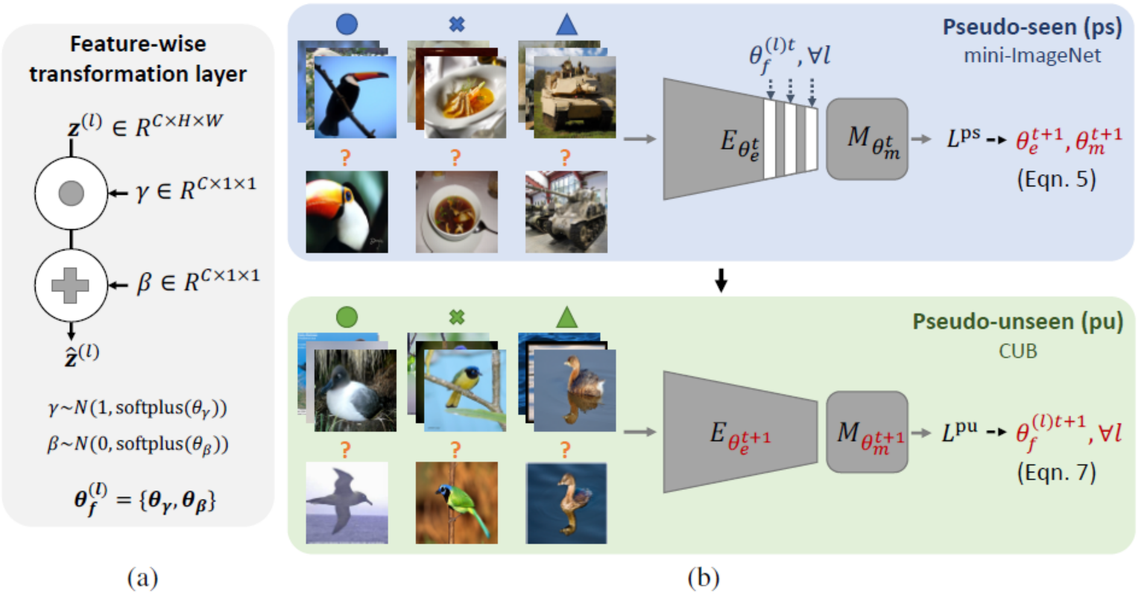

Feature-wise transformation layer
$\mathbf{z}^{(l)} \in R^{C\times H\times W}$
$\gamma \in R^{C\times 1\times 1}$
$\beta \in R^{C\times 1\times 1}$
$\hat{\mathbf{z}}^{(l)}$
$\gamma \sim N(1, \text{softplus}(\theta_\gamma))$
$\beta \sim N(0, \text{softplus}(\theta_\beta))$
$\boldsymbol{\theta}_f^{(l)} = \{\boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_\beta\}$
Pseudo-seen (ps)
mini-ImageNet
$\theta_f^{(l)t}, \forall l$
?
?
?
$E_{\theta_e^t}$
$M_{\theta_m^t}$
L^{ps}
$\theta_e^{t+1}, \theta_m^{t+1}$
(Eqn. 5)
Pseudo-unseen (pu)
CUB
?
?
?
$E_{\theta_e^{t+1}}$
$M_{\theta_m^{t+1}}$
L^{pu}
$\theta_f^{(l)t+1}, \forall l$
(Eqn. 7)
(a)
(b)

Figure 2: **Method overview.** (a) We propose a feature-wise transformation layer to modulate intermediate feature activation $\mathbf{z}$ in the feature encoder E with the scaling and bias terms sampled from the Gaussian distributions parameterized by the hyper-parameters θ_γ and θ_β. During the training phase, we insert a collection of feature-wise transformation layers into the feature encoder to simulate feature distributions extracted from the tasks in various domains. (b) We design a learning-to-learn algorithm to optimize the hyper-parameters θ_γ and θ_β of feature-wise transformation layers by maximizing the performance of the applied metric-based model on the pseudo-unseen domain (*bottom*) after it is optimized on the pseudo-seen domain (*top*).

基于度量的模型通常包括一个特征编码器E和度量函数M。

作者将Feature-Wise转换层集成到特征编码器E中，来产生更多样的特征分布，提高度量函数M的泛化能力。

给定特征编码器E输出的中间特征z，维度为$C \times H \times W$。首先从高斯分布采样得到Feature-Wise转换层的参数

$$\gamma \sim N(\mathbf{1}, \text{softplus}(\theta_\gamma)) \quad \beta \sim N(\mathbf{0}, \text{softplus}(\theta_\beta)). \tag{3}$$

其中超参数 $\theta_\gamma \in R^{C\times 1\times 1}$ and $\theta_\beta \in R^{C\times 1\times 1}$ 是对仿射变换参数进行抽样的高斯分布的标准差。

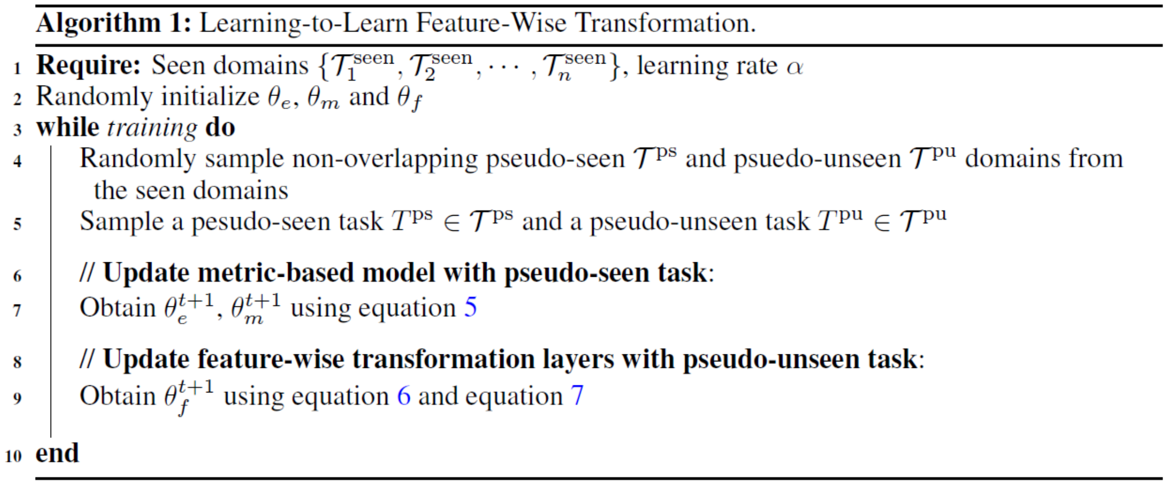
然后通过仿射变换计算出调整后的激活特征

$$\hat{z}_{c,h,w} = \gamma_c \times z_{c,h,w} + \beta_c, \tag{4}$$

实践中，作者在特征编码器E的多个层后都添加了Feature-Wise转换层。

学习Feature-Wise转换层

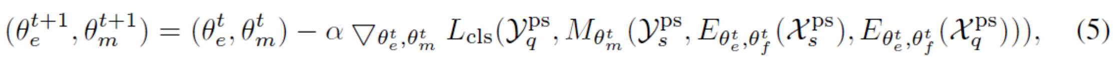
作者设计一个元学习算法优化Feature-Wise转换层的超参数。



Algorithm 1: Learning-to-Learn Feature-Wise Transformation.

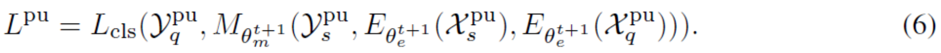
```
Require: Seen domains $\{\mathcal{T}_1^{\text{seen}}, \mathcal{T}_2^{\text{seen}}, \cdots, \mathcal{T}_n^{\text{seen}}\}$, learning rate $\alpha$
Randomly initialize $\theta_e$, $\theta_m$ and $\theta_f$
while training do
    Randomly sample non-overlapping pseudo-seen $\mathcal{T}^{\text{ps}}$ and psuedo-unseen $\mathcal{T}^{\text{pu}}$ domains from
      the seen domains
    Sample a pesudo-seen task $T^{\text{ps}} \in \mathcal{T}^{\text{ps}}$ and a pseudo-unseen task $T^{\text{pu}} \in \mathcal{T}^{\text{pu}}$
    // Update metric-based model with pseudo-seen task:
    Obtain $\theta_e^{t+1}$, $\theta_m^{t+1}$ using equation 5
    // Update feature-wise transformation layers with pseudo-unseen task:
    Obtain $\theta_f^{t+1}$ using equation 6 and equation 7
end
```

首先将Feature-Wise转换层集成到E中，然后根据伪可见任务的分类Loss更新原模型的参数：



$$(\theta_e^{t+1}, \theta_m^{t+1}) = (\theta_e^t, \theta_m^t) - \alpha \bigtriangledown_{\theta_e^t, \theta_m^t} L_{\text{cls}}(\mathcal{Y}_q^{\text{ps}}, M_{\theta_m^t}(\mathcal{Y}_s^{\text{ps}}, E_{\theta_e^t, \theta_f^t}(\mathcal{X}_s^{\text{ps}}), E_{\theta_e^t, \theta_f^t}(\mathcal{X}_q^{\text{ps}}))), \quad (5)$$

然后从模型中删除Feature-Wise转换层，在伪不可见任务上计算更新后的模型的分类损失



$$L^{\text{pu}} = L_{\text{cls}}(\mathcal{Y}_q^{\text{pu}}, M_{\theta_m^{t+1}}(\mathcal{Y}_s^{\text{pu}}, E_{\theta_e^{t+1}}(\mathcal{X}_s^{\text{pu}}), E_{\theta_e^{t+1}}(\mathcal{X}_q^{\text{pu}}))). \quad (6)$$

由于L^{pu}损失反映了Feature-Wise变换层的有效性，所以就使用这个损失来更新Feature-Wise转换层的超参数

$$\theta_f^{t+1} = \theta_f^t - \alpha \bigtriangledown_{\theta_f^t} L^{\text{pu}}. \quad (7)$$

实验

Resnet10，在所有残差块的最后一个BN层后添加了Feature-Wise转换层。根据经验确定超参数。在mini-ImageNet上训练，在其余4个数据集上进行测试。

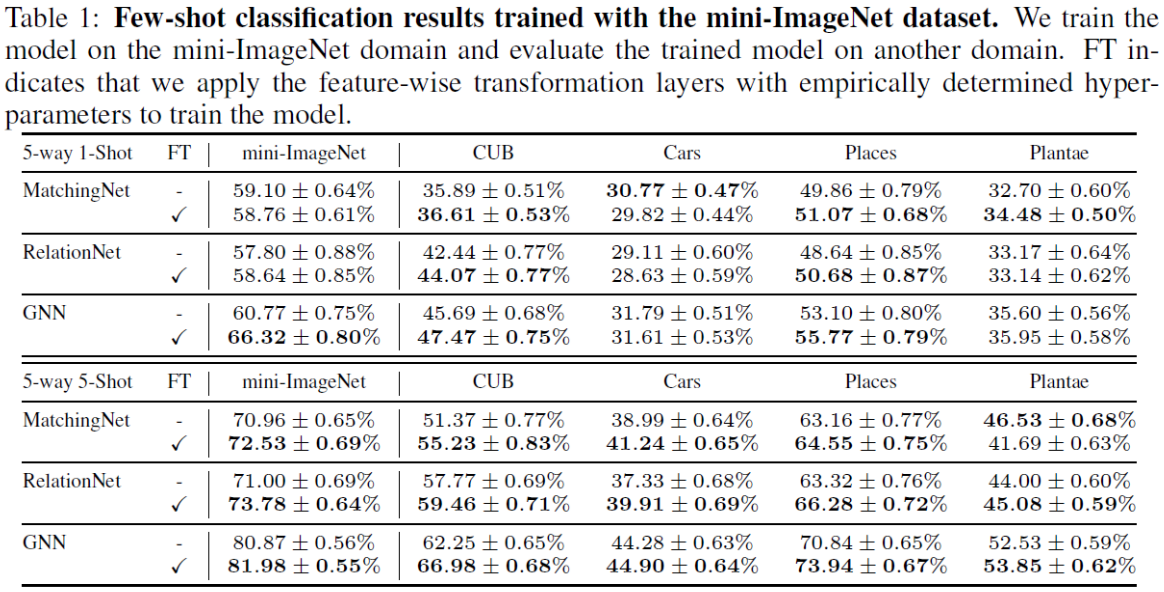


Table 1: **Few-shot classification results trained with the mini-ImageNet dataset.** We train the model on the mini-ImageNet domain and evaluate the trained model on another domain. FT indicates that we apply the feature-wise transformation layers with empirically determined hyper-parameters to train the model.

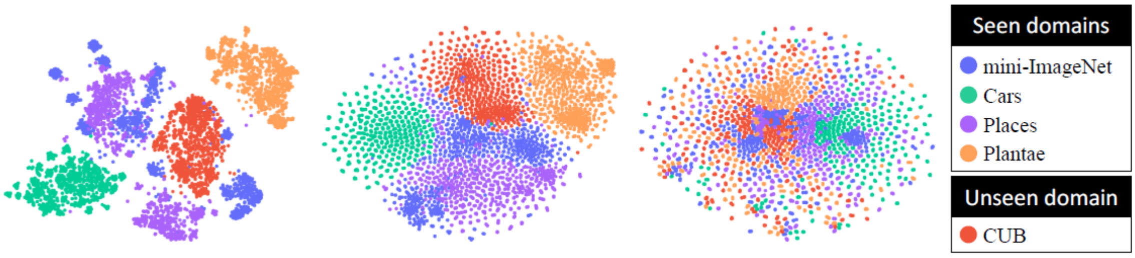
5-way 1-Shot	FT	mini-ImageNet	CUB	Cars	Places	Plantae
MatchingNet	-	59.10 ± 0.64%	35.89 ± 0.51%	**30.77 ± 0.47%**	49.86 ± 0.79%	32.70 ± 0.60%
	✓	58.76 ± 0.61%	**36.61 ± 0.53%**	29.82 ± 0.44%	**51.07 ± 0.68%**	**34.48 ± 0.50%**
RelationNet	-	57.80 ± 0.88%	42.44 ± 0.77%	29.11 ± 0.60%	48.64 ± 0.85%	33.17 ± 0.64%
	✓	58.64 ± 0.85%	**44.07 ± 0.77%**	28.63 ± 0.59%	**50.68 ± 0.87%**	33.14 ± 0.62%
GNN	-	60.77 ± 0.75%	45.69 ± 0.68%	31.79 ± 0.51%	53.10 ± 0.80%	35.60 ± 0.56%
	✓	**66.32 ± 0.80%**	**47.47 ± 0.75%**	31.61 ± 0.53%	**55.77 ± 0.79%**	35.95 ± 0.58%
5-way 5-Shot	FT	mini-ImageNet	CUB	Cars	Places	Plantae
MatchingNet	-	70.96 ± 0.65%	51.37 ± 0.77%	38.99 ± 0.64%	63.16 ± 0.77%	**46.53 ± 0.68%**
	✓	**72.53 ± 0.69%**	**55.23 ± 0.83%**	**41.24 ± 0.65%**	**64.55 ± 0.75%**	41.69 ± 0.63%
RelationNet	-	71.00 ± 0.69%	57.77 ± 0.69%	37.33 ± 0.68%	63.32 ± 0.76%	44.00 ± 0.60%
	✓	**73.78 ± 0.64%**	**59.46 ± 0.71%**	**39.91 ± 0.69%**	**66.28 ± 0.72%**	**45.08 ± 0.59%**
GNN	-	80.87 ± 0.56%	62.25 ± 0.65%	44.28 ± 0.63%	70.84 ± 0.65%	52.53 ± 0.59%
	✓	**81.98 ± 0.55%**	**66.98 ± 0.68%**	**44.90 ± 0.64%**	**73.94 ± 0.67%**	**53.85 ± 0.62%**

验证提出的学习算法的有效性。从除mini-ImageNet外的四个数据集中选择一个作为不可见域，其余数据集作为训练集的可见域。

Table 2: **Few-shot classification results trained with multiple datasets.** We use the leave-one-out setting to select the unseen domain and train the model as well as the feature-wise transformation layers using Algorithm 1. FT and LFT indicate applying the pre-determined and learning-to-learned feature-wise transformation, respectively.

5-way 1-Shot		CUB	Cars	Places	Plantae
MatchingNet	-	$37.90 \pm 0.55\%$	$28.96 \pm 0.45\%$	$49.01 \pm 0.65\%$	$33.21 \pm 0.51\%$
	FT	$41.74 \pm 0.59\%$	$28.30 \pm 0.44\%$	$48.77 \pm 0.65\%$	$32.15 \pm 0.50\%$
	LFT	$\mathbf{43.29 \pm 0.59\%}$	$\mathbf{30.62 \pm 0.48\%}$	$\mathbf{52.51 \pm 0.67\%}$	$\mathbf{35.12 \pm 0.54\%}$
RelationNet	-	$44.33 \pm 0.59\%$	$29.53 \pm 0.45\%$	$47.76 \pm 0.63\%$	$33.76 \pm 0.52\%$
	FT	$44.67 \pm 0.58\%$	$30.38 \pm 0.47\%$	$48.40 \pm 0.64\%$	$35.40 \pm 0.53\%$
	LFT	$\mathbf{48.38 \pm 0.63\%}$	$\mathbf{32.21 \pm 0.51\%}$	$\mathbf{50.74 \pm 0.66\%}$	$35.00 \pm 0.52\%$
GNN	-	$49.46 \pm 0.73\%$	$32.95 \pm 0.56\%$	$51.39 \pm 0.80\%$	$37.15 \pm 0.60\%$
	FT	$48.24 \pm 0.75\%$	$33.26 \pm 0.56\%$	$54.81 \pm 0.81\%$	$37.54 \pm 0.62\%$
	LFT	$\mathbf{51.51 \pm 0.80\%}$	$\mathbf{34.12 \pm 0.63\%}$	$\mathbf{56.31 \pm 0.80\%}$	$\mathbf{42.09 \pm 0.68\%}$
5-way 5-Shot		**CUB**	**Cars**	**Places**	**Plantae**
MatchingNet	-	$51.92 \pm 0.80\%$	$39.87 \pm 0.51\%$	$61.82 \pm 0.57\%$	$47.29 \pm 0.51\%$
	FT	$56.29 \pm 0.80\%$	$39.58 \pm 0.54\%$	$62.32 \pm 0.58\%$	$46.48 \pm 0.52\%$
	LFT	$\mathbf{61.41 \pm 0.57\%}$	$\mathbf{43.08 \pm 0.55\%}$	$\mathbf{64.99 \pm 0.59\%}$	$\mathbf{48.32 \pm 0.57\%}$
RelationNet	-	$62.13 \pm 0.74\%$	$40.64 \pm 0.54\%$	$64.34 \pm 0.57\%$	$46.29 \pm 0.56\%$
	FT	$63.64 \pm 0.77\%$	$42.24 \pm 0.57\%$	$65.42 \pm 0.58\%$	$47.81 \pm 0.51\%$
	LFT	$\mathbf{64.99 \pm 0.54\%}$	$\mathbf{43.44 \pm 0.59\%}$	$\mathbf{67.35 \pm 0.54\%}$	$\mathbf{50.39 \pm 0.52\%}$
GNN	-	$69.26 \pm 0.68\%$	$48.91 \pm 0.67\%$	$72.59 \pm 0.67\%$	$58.36 \pm 0.68\%$
	FT	$70.37 \pm 0.68\%$	$47.68 \pm 0.63\%$	$74.48 \pm 0.70\%$	$57.85 \pm 0.68\%$
	LFT	$\mathbf{73.11 \pm 0.68\%}$	$\mathbf{49.88 \pm 0.67\%}$	$\mathbf{77.05 \pm 0.65\%}$	$\mathbf{58.84 \pm 0.66\%}$

可视化



(a) RelationNet (b) RelationNet + FT (c) RelationNet + LFT

Figure 3: **T-SNE visualization of the image features extracted from tasks in different domains.** We show the t-SNE visualization of the features extracted by the (a) original feature encoder E, (b) feature encoder with pre-determined feature-wise transformation layers, and (c) feature encoder with learning-to-learned feature-wise transformation.

在Feature-Wise转换层的帮助下，从不同域提取的特征之间的距离变得更小。

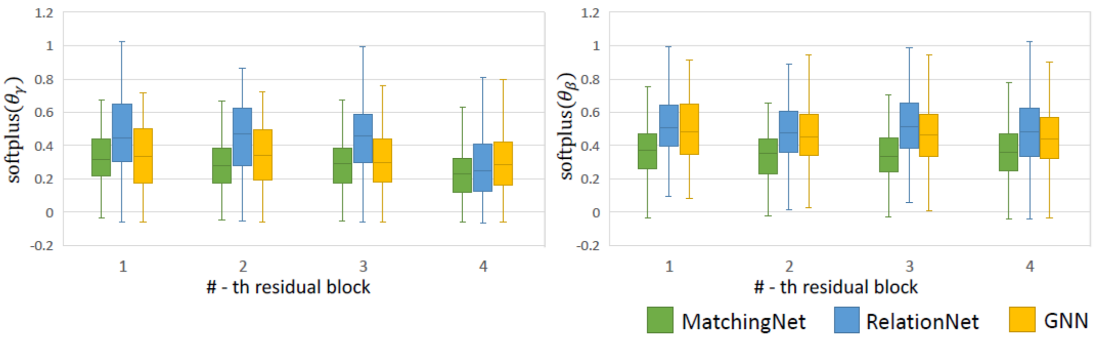


Figure 4: **Visualization of the feature-wise transformation layers.** We show the quartile visualization of the activations $\text{softplus}(\theta_\gamma)$ and $\text{softplus}(\theta_\beta)$ from each feature-wise transformation layer that are optimized by the proposed learning-to-learn algorithm.

θ_γ在较深的层中值变小，特别是在最后的残差块中。层的深度似乎对偏置项的分布没有明显的影响。