复旦大学

Constructing Multiple Tasks for Augmentation: Improving Neural Image Classification With K-means Features

作者：Tao Gui, Lizhi Qing, Qi Zhang, Jiacheng Ye, Hang Yan, Zichu Fei, Xuanjing Huang

报告人：卿立之

单位：复旦大学

MOTIVATION

Color

- Mango、orange
- Apple、pitaya

Shape

- Mango、pitaya
- Apple、orange

THINKING

We need more related tasks!

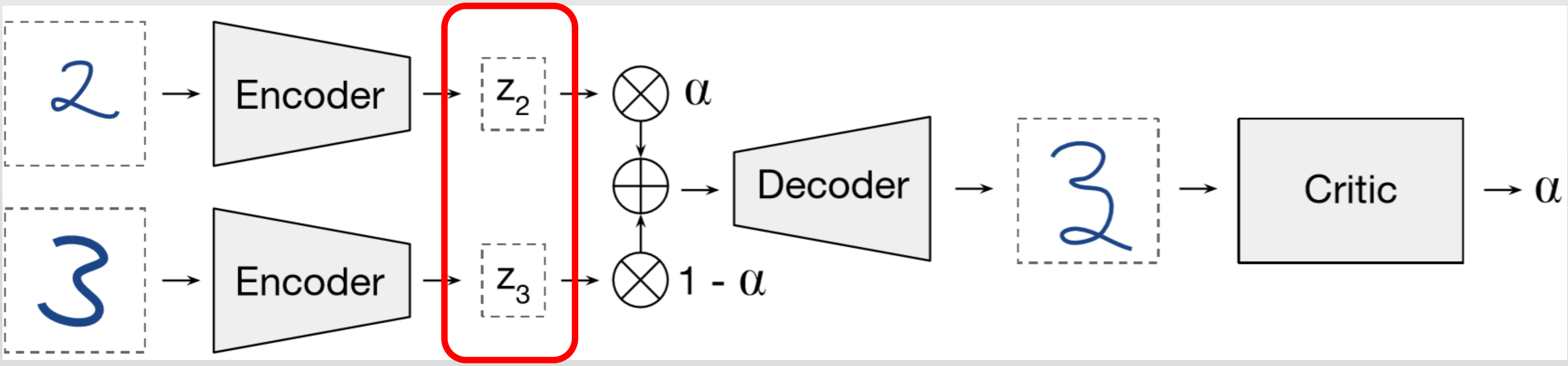
How to construct multiple related tasks?

《Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer》

Interpolation:

AUTOENCODER

《Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer》



《Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer》

Table 2: Single-layer classifier accuracy achieved by different autoencoders.

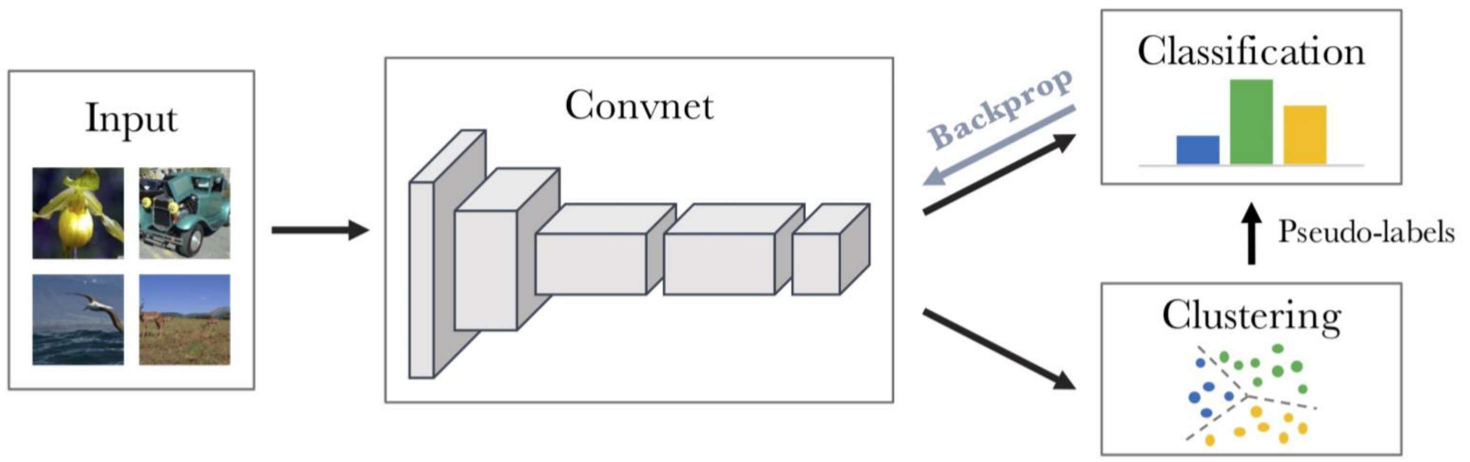
Dataset	d_z	Baseline	Dropout	Denoising	VAE	AAE	VQ-VAE	ACAI
MNIST	32	94.90±0.14	96.45±0.42	96.00±0.27	96.56±0.31	70.74±3.27	97.50±0.18	**98.25±0.11**
	256	93.94±0.13	94.50±0.29	98.51±0.04	98.74±0.14	90.03±0.54	97.25±1.42	**99.00±0.08**
SVHN	32	26.21±0.42	26.09±1.48	25.15±0.78	29.58±3.22	23.43±0.79	24.53±1.33	**34.47±1.14**
	256	22.74±0.05	25.12±1.05	77.89±0.35	66.30±1.06	22.81±0.24	44.94±20.42	**85.14±0.20**
CIFAR-10	256	47.92±0.20	40.99±0.41	**53.78±0.36**	47.49±0.22	40.65±1.45	42.80±0.44	52.77±0.45
	1024	51.62±0.25	49.38±0.77	60.65±0.14	51.39±0.46	42.86±0.88	16.22±12.44	**63.99±0.47**

EXAMPLES OF K-MEANS CLUSTERING

"Blue sky problem"

DEEPCLUSTER

《Deep Clustering for Unsupervised Learning of Visual Features》



$$\min_{\theta, W} \frac{1}{N} \sum_{n=1}^{N} \ell\left(g_W\left(f_\theta(x_n)\right), y_n\right)$$

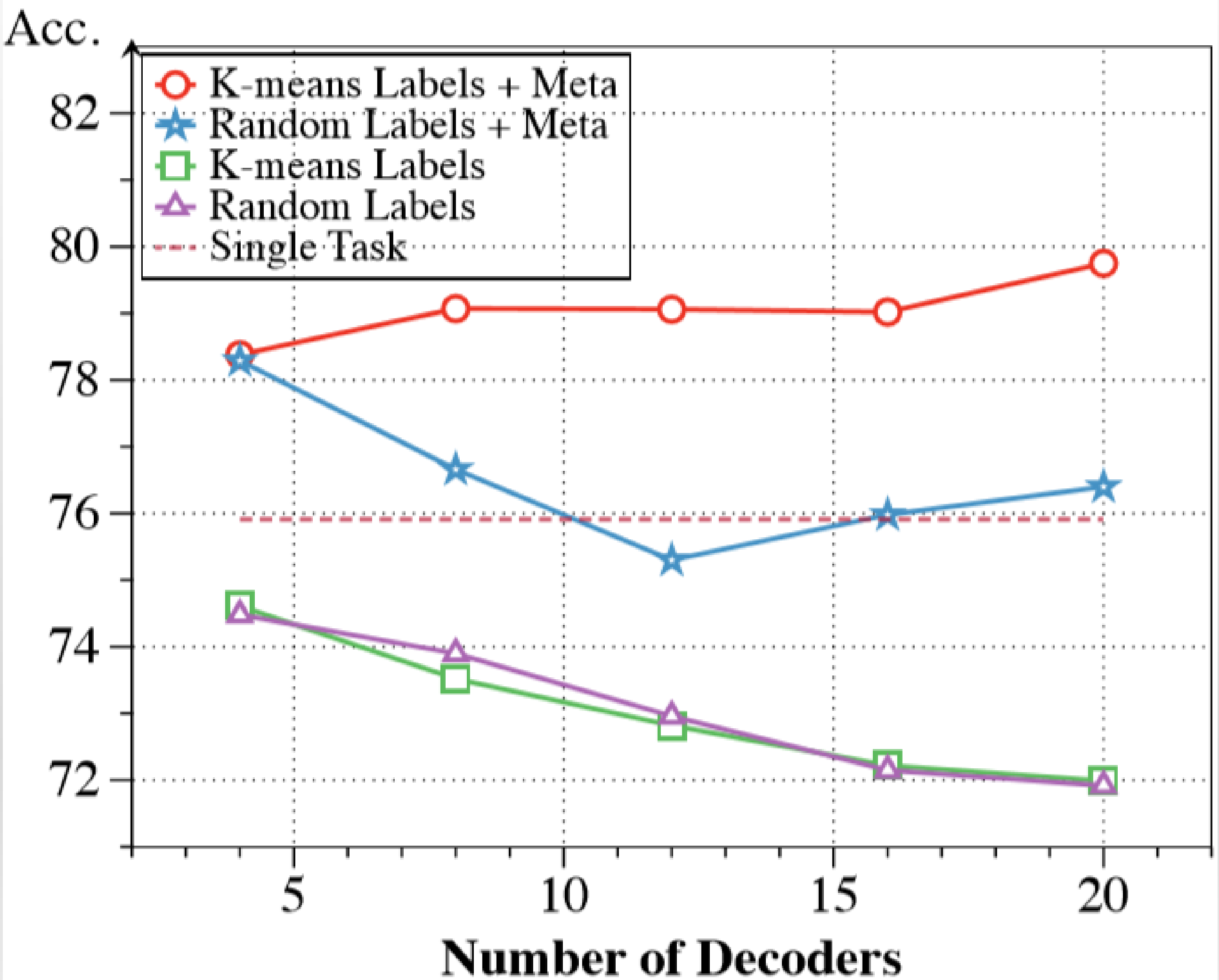
$$\min_{C \in \mathbb{R}^{d \times k}} \frac{1}{N} \sum_{n=1}^{N} \min_{y_n \in \{0,1\}^k} \| f_\theta(x_n) - C y_n \|_2^2 \quad \text{such that} \quad y_n^\top 1_k = 1$$

THINKING

We have enough related tasks!

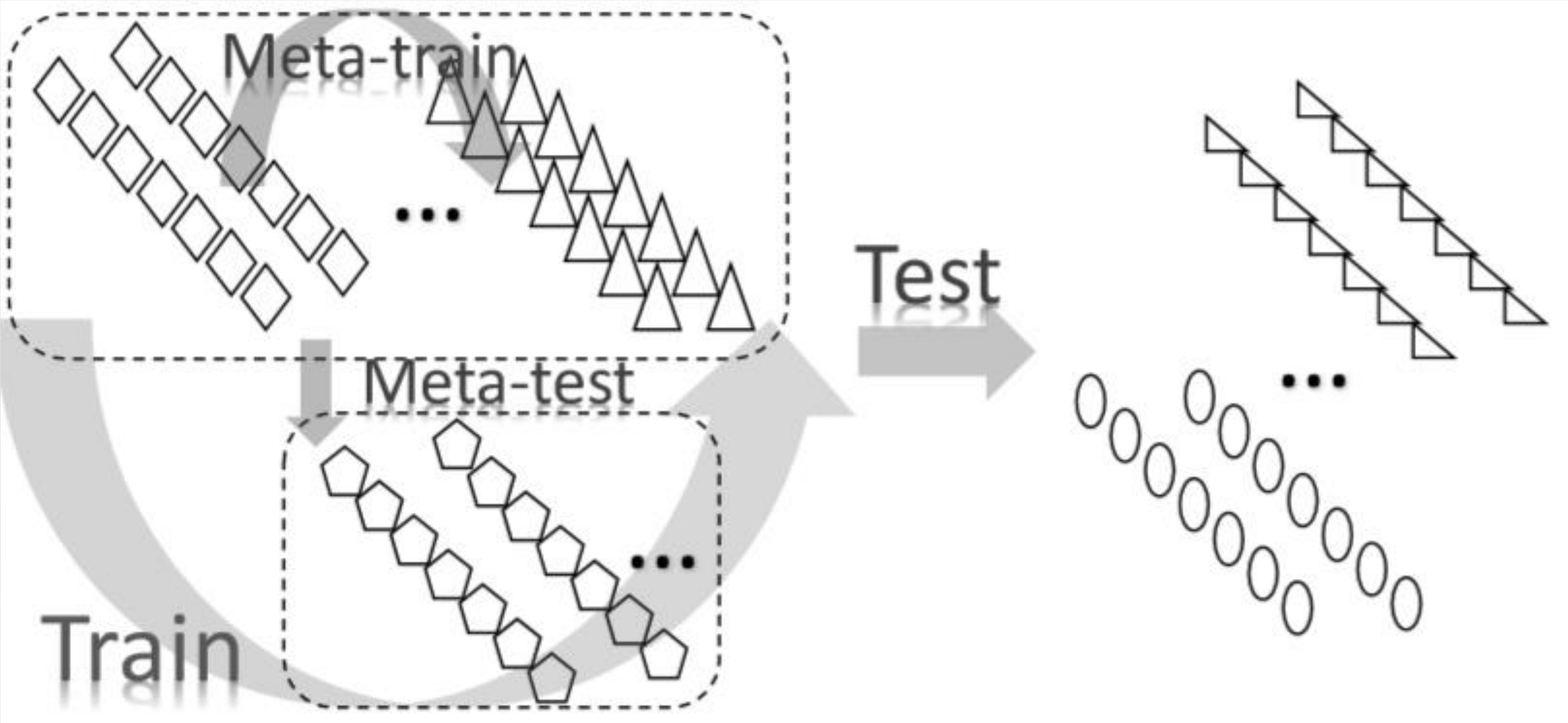
How to train them?

COMPARISON ON CIFAR10

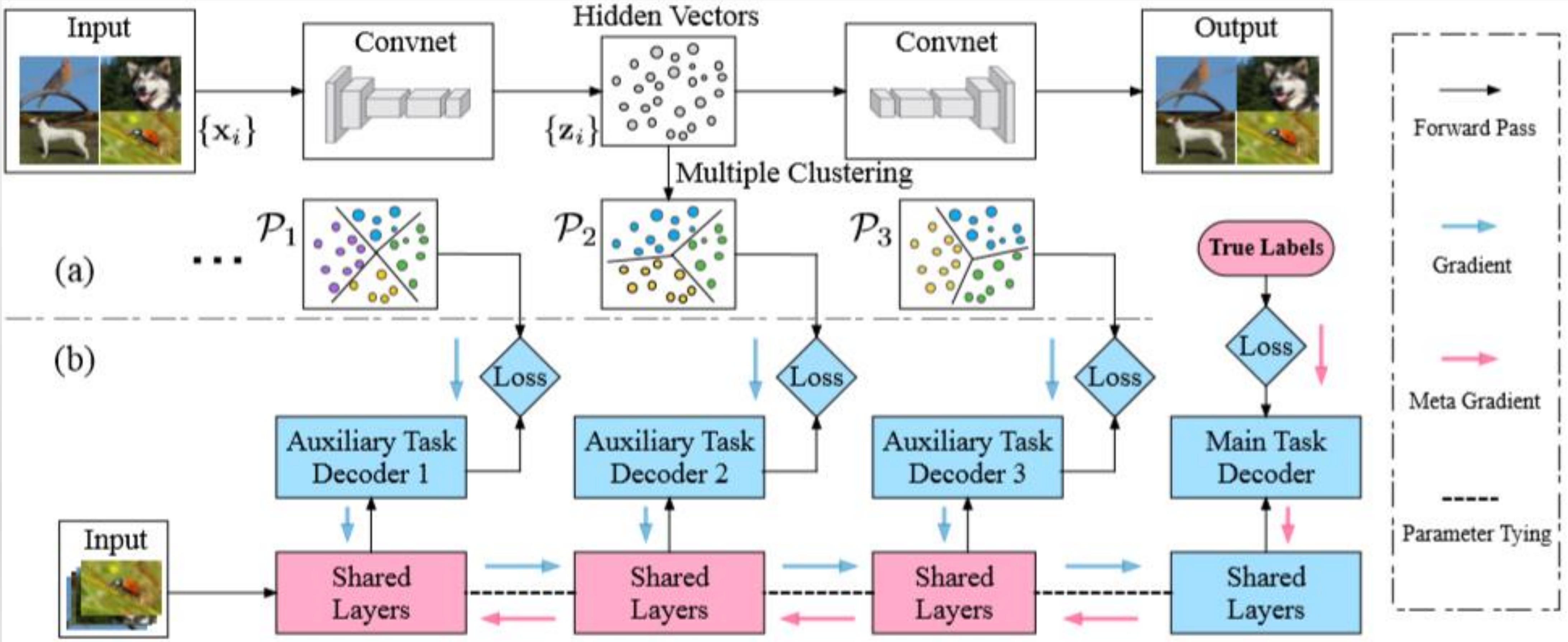


META-LEARNING

《*Learning to generalize: Meta-learning for domain generalization*》



META-MTL



Input
$\{x_i\}$
Convnet
Hidden Vectors
$\{z_i\}$
Convnet
Output
Multiple Clustering
(a)
...
$\mathcal{P}_1$
$\mathcal{P}_2$
$\mathcal{P}_3$
True Labels
(b)
Loss
Loss
Loss
Loss
Auxiliary Task Decoder 1
Auxiliary Task Decoder 2
Auxiliary Task Decoder 3
Main Task Decoder
Input
Shared Layers
Shared Layers
Shared Layers
Shared Layers
Forward Pass
Gradient
Meta Gradient
Parameter Tying

META-MTL

Algorithm 1 **Meta-MTL with K-means Augmentation**

Run embedding learning algorithm $\mathcal{E}$ on D_{aux} and produce embeddings $\{\mathbf{z}_i\}$ from observations $\{\mathbf{x}_i\}$.
Run k-means on $\{\mathbf{z}_i\}$ T times (with random scaling or random selection on dimensions) to generate a set of partitions $\{\mathcal{P}_t = \{C^l\}_{l=1}^{L_t}\}_{t=1}^{T}$, which correspond to a set of auxiliary tasks $\{\mathcal{T}_t\}_{t=1}^{T}$.
for episode = 1, M **do**
  Sample batch of tasks $\mathcal{T} \sim \{\mathcal{T}_t\}_{t=0}^{T}$.
  for all $\mathcal{T}$ **do**
    Sample K datapoints $D_{\mathcal{T}} = \{\mathbf{x}_j, \mathbf{y}_j\}$.
    Evaluate $\nabla_{\theta_{\mathcal{F}}}$ and $\nabla_{\theta_{\mathcal{D}_t}}$ using $D_{\mathcal{T}}$ based on Equation 1.
    Applying gradient decent to update the parameters of task-specific decoders $\theta_{\mathcal{D}_{\mathcal{T}}}$.
    Compute updated parameters $\theta^*_{\mathcal{F}}$ with gradient descent based on Equation 5.
    Sample datapoints $D_0 = \{\mathbf{x}_j, \mathbf{y}_j\}$ from $\mathcal{T}_0$ for the meta-update.
  end for
  Update the parameters of shared layers $\theta_{\mathcal{F}}$ based on Equation 6.
end for

$$\hat{\mathbf{y}}_i^t = \mathcal{D}_t(\mathcal{F}(\mathbf{x}_i^t; \theta_{\mathcal{F}}); \theta_{\mathcal{D}_t}), \qquad (1)$$

$$\begin{aligned} \theta_{\mathcal{D}_{\mathcal{T}}} &= \theta_{\mathcal{D}_{\mathcal{T}}} - \alpha \nabla \mathcal{L}_{\mathcal{D}_{\mathcal{T}}}(\theta_{\mathcal{D}_{\mathcal{T}}}) \\ \theta^*_{\mathcal{F}} &= \theta_{\mathcal{F}} - \alpha \nabla \mathcal{L}_{\mathcal{D}_{\mathcal{T}}}(\theta_{\mathcal{F}}). \end{aligned} \qquad (5)$$

$$\begin{aligned} \theta_{\mathcal{F}} &= \theta_{\mathcal{F}} - \beta \nabla \theta_{\mathcal{F}} \mathcal{L}_{\mathcal{D}_0}(\theta^*_{\mathcal{F}}) \\ &= \theta_{\mathcal{F}} - \beta \nabla \theta_{\mathcal{F}} \mathcal{L}_{\mathcal{D}_0}(\theta_{\mathcal{F}} - \alpha \nabla \mathcal{L}_{\mathcal{D}_{\mathcal{T}}}(\theta_{\mathcal{F}})), \end{aligned} \qquad (6)$$

EXPERIMENTS——Omniglot

1. Limited training data, multiple classes

Method	Acc.
STL (Yang and Hospedales 2016)	65.72
ACAI embedding finetune (Berthelot et al. 2019)	67.91
MTL on all alphabets (Yang and Hospedales 2016)	70.98
MTL + Tasks with random labels, $T = 4$	60.98
MTL + Tasks with k-means labels, $T = 4$	61.26
PTA-F, $T = 4$ (Meyerson and Miikkulainen 2018)	70.63
PTA-F, $T = 10$ (Meyerson and Miikkulainen 2018)	71.52
Meta-MTL, $T = 4$	**72.04**
Meta-MTL, $T = 10$	**74.80**

Table 1: Accuracy on Omniglot dataset. THe test accuracy averaged across 50 alphabets is shown.

2. Benefit of unlabeled data

Method	Acc.
STL (Yang and Hospedales 2016)	91.83
ACAI embedding finetune (Berthelot et al. 2019)	93.09
Self-training (Rosenberg, Hebert, and Schneiderman 2005)	92.20
Co-training (Chen, Weinberger, and Blitzer 2011)	91.87
MTL + Tasks with random labels, $T = 4$	92.27
MTL + Tasks with k-means labels, $T = 4$	92.86
PTA-F, $T = 4$ (Meyerson and Miikkulainen 2018)	92.67
PTA-F, $T = 10$ (Meyerson and Miikkulainen 2018)	91.98
Meta-MTL, $T = 4$	**93.37**
Meta-MTL, $T = 10$	**94.22**
Meta-MTL, $T = 4\dagger$	**93.76**
Meta-MTL, $T = 10\dagger$	**94.42**

Table 2: Accuracy on MNIST dataset. All of the models use 1% training data. The models marked with † use the remaining unlabeled data.

EXPERIMENTS——CIFAR10&100

3. Improve the performance of model with some data augmentation skills.

Model	Acc. CNN	Acc. CNN‡
STL (Yang and Hospedales 2016)	72.48	75.91
ACAI embedding finetune (Berthelot et al. 2019)	64.94	64.94
MTL + Tasks with random labels, $T = 4$	70.81	74.48
MTL + Tasks with k-means labels $T = 4$	71.06	74.61
PTA-F, $T = 4$ (Meyerson and Miikkulainen 2018)	69.20	72.54
PTA-F, $T = 12$ (Meyerson and Miikkulainen 2018)	68.48	70.96
Meta-MTL, $T = 4$	**75.02**	**78.65**
Meta-MTL, $T = 12$	**75.69**	**79.65**

Table 3: Accuracy on CIFAR-10 dataset. The models marked with ‡ apply data augmentation.

EXPERIMENTS——CIFAR10&100

4. Complex realistic images, multiple classes

Model	Acc.	
	20 C	100 C
STL (Yang and Hospedales 2016)	55.94	44.19
ACAI embedding finetune (Berthelot et al. 2019)	44.37	34.40
MTL + Tasks with random labels	51.00	41.30
MTL + Tasks with k-means labels	51.84	42.30
PTA-F, $T = 8$ (Meyerson and Miikkulainen 2018)	51.67	45.86
PTA-F, $T = 20$ (Meyerson and Miikkulainen 2018)	51.69	47.43
Meta-MTL, $T = 8$	**59.66**	**47.01**
Meta-MTL, $T = 20$	**60.39**	**47.94**

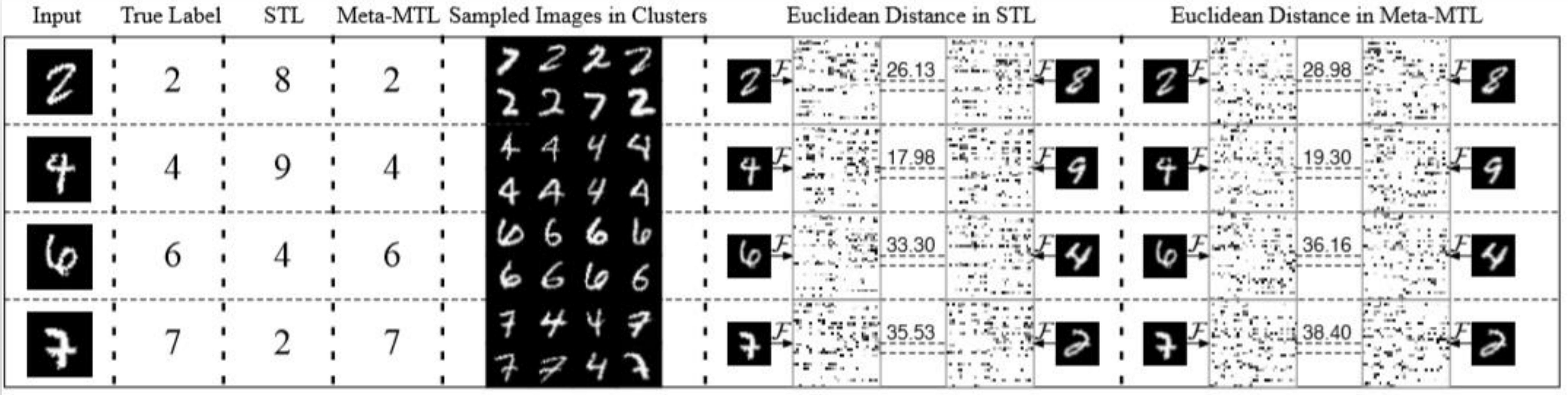
Table 4: Accuracy on CIFAR-100 dataset. The "20 C" means that the 100 classes in the CIFAR-100 are grouped into 20 superclasses, and the "100 C" means the original 100 classes.

5. Challenging benchmark

Embedding	Model	Acc.
Random initialization	STL	38.98
	MTL, Random Labels	36.70
	PTA-F, $T = 4$	34.93
DeepCluster	Embedding finetune	35.14
	MTL, k-means, $T = 4, \diamondsuit$	37.24
	MTL, k-means, $T = 4, \heartsuit$	36.78
	Meta-MTL, $T = 4, \diamondsuit$	**41.07**
	Meta-MTL, $T = 4, \heartsuit$	**40.86**
ACAI	Embedding finetune	22.90
	MTL, k-means, $T = 4, \diamondsuit$	37.36
	MTL, k-means, $T = 4, \heartsuit$	36.98
	Meta-MTL, $T = 4, \diamondsuit$	**40.43**
	Meta-MTL, $T = 4, \heartsuit$	**40.60**

Table 5: Accuracy on miniImageNet dataset. The models marked with $\diamondsuit$ apply the random scaling on the embeddings to obtain the different tasks, while those marked with $\heartsuit$ apply random selection for half of the dimensions on the embeddings.

Visualization



Thank you
& QA