DEJA VU: CONTINUAL MODEL GENERALIZATION FOR UNSEEN DOMAINS

Presented By : Team Model Mavericks

PAPER REVIEW | CS771 – IIT Kanpur

INTRODUCTION

Deep learning models often run in non-stationary environments where the target data distribution continually shifts over time. The paper "DEJA VU: Continual Model Generalization for Unseen Domains" focuses on the challenges of enabling machine learning models to generalize effectively to new, unseen domains while being trained continually on a stream of data.

● LIMITATIONS OF DA (DOMAIN ADAPTABILITY METHODS)

- DA methods aim to adapt a model to new domains, either online or offline.(cross domain adaptation ability)
- Limitation: DA methods require time to adapt, which leads to poor performance during the "Unfamiliar Period"—a phase when the model encounters sudden, significant domain shifts.

● LIMITATIONS OF DG (DOMAIN GENERALIZATION METHODS)

- DG methods enhance a model's ability to generalize to unseen domains without adaptation.
- Limitation: DG methods struggle with catastrophic forgetting when exposed to continually changing domains.

PROBLEM

First Problem

Machine learning models often face a trade-off between generalization to unseen domains and robustness to domain shifts when trained on multiple tasks or domains.

Second Problem

Continual learning systems typically suffer from catastrophic forgetting—the degradation of performance on previously learned domains while adapting to new ones.

The paper tackles the dual challenge of generalization and continual learning to achieve performance on unseen tasks without forgetting prior knowledge.

RaTP FRAMEWORK

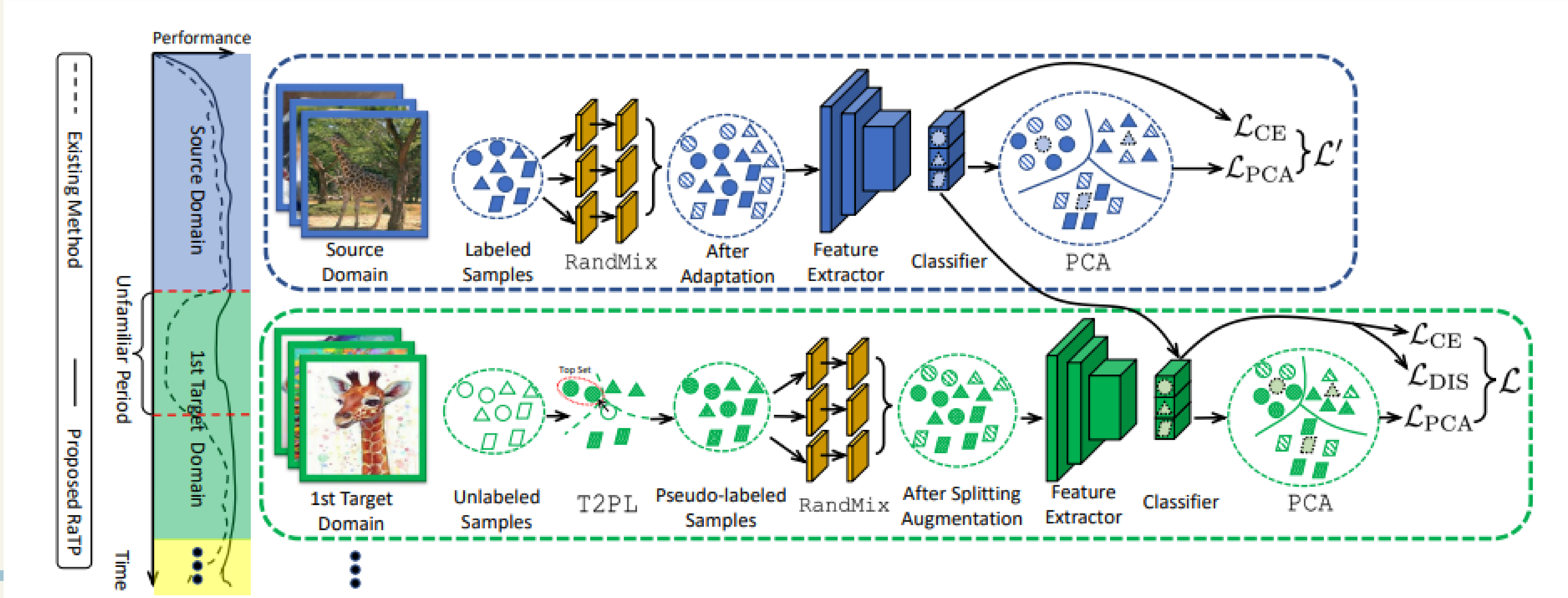
RaTP is a novel framework designed to tackle the limitations of DA and DG by focusing on three core capabilities:

1 **Target Domain Generalization (TDG):**
- **Generalizing effectively to unseen domains without prior adaptation.**

2 **Target Domain Adaptation (TDA):**
- **Quickly adapting to new domains after a brief exposure.**

3 **Forgetting Alleviation (FA):**
- **Retaining knowledge of previously learned domains to avoid catastrophic forgetting.**

OVERVIEW OF RaTP FOR CDSL



PROBLEM FORMULATION OF CDSL

Labeled source domain dataset S={(xi,yi) | xi PSX, yi PSY} i=1NS

PX and PY => the input and label distributions

NS => sample quantity

This paper chooses visual recognition as the learning task, in which the number of data classes is K.

Continually arriving target domains T = {T t} t=1 to T ; T => total no of domains.

The generalization performance for the t-th domain T t depends on both the previously seen t-1 target domains and the original source domain, i.e., S = {S, ∪j=1t-1 Tj}.

PROBLEM FORMULATION OF CDSL

We also set an exemplar memory M to store $|M|/t$ exemplars that are closest to the class centroids for each domain in S, where $|M|/t << NT$, $|M|/t << NS$.

Model is a deep neural network, feature extractor $f\theta$ at the bottom and a classifier $g\Omega$ at the top

For each target domain dataset Tt, in addition to the current model $f\theta t \circ g\Omega t$, its inherited version $f\theta t-1 \circ g\Omega t-1$ from the last domain is also stored for later use.

RANDOM MIXUP AUGMENTATION

- RandMix relies on Naug simple autoencoders R = {Ri} Naug i=1 , where each autoencoder consists of an encoder eξ and a decoder dζ .
- The encoder eξ and the decoder dζ are implemented as a convolutional layer and a transposed convolutional layer
- To introduce more randomness, we apply AdaIN to inject noise into the autoencoder
- The used AdaIN contains two linear layers lφ1 , lφ2 , and when lφ1 and lφ2 are fed with a certain noisy input

$$R(\boldsymbol{x}) = d_\zeta\left(l_{\phi_1}(n) \times l_{\mathrm{IN}}(e_\xi(\boldsymbol{x})) + l_{\phi_2}(n)\right)$$

RANDOM MIXUP AUGMENTATION

the mixture is scaled by a sigmoid function

$$\mathbf{R}(\boldsymbol{x}) = \sigma\left(\frac{1}{\sum_{i=0}^{N_{\text{aug}}} \mathrm{w}_i}\left[\mathrm{w}_0\boldsymbol{x} + \sum_{i=1}^{N_{\text{aug}}}(\mathrm{w}_i R_i(\boldsymbol{x}))\right]\right)$$

With RandMix, we can generate augmentation data with the same labels corresponding to all labeled samples from the source domain.

$$\mathbf{w} = \{\mathrm{w}_i \| \mathrm{w}_i \sim \mathcal{P}_{\mathrm{N}(0,1)}\}_{i=0}^{N_{\text{aug}}}$$

To avoid error propagation and accumulation, we augment a subset of the target data rather than the full set. Prediction confidence is defined as follows:

$$\tilde{\boldsymbol{x}} = \begin{cases} \mathbf{R}(\boldsymbol{x}), & \text{if} \max\left[g_\omega(f_\theta(\boldsymbol{x}))\right]_K \geq r_{\text{con}} \\ \emptyset, & \text{otherwise} \end{cases}, \boldsymbol{x} \sim \mathcal{P}_X^{\mathcal{T}},$$

TOP2 PSEUDO LABELING

we use the softmax confidence to measure the possibility of correct classification for data samples, and select the top 50% set to construct class centroids
Softmax confidence is used to filter data points based on their classification certainty, improving the quality of pseudo-labeling in clustering methods.

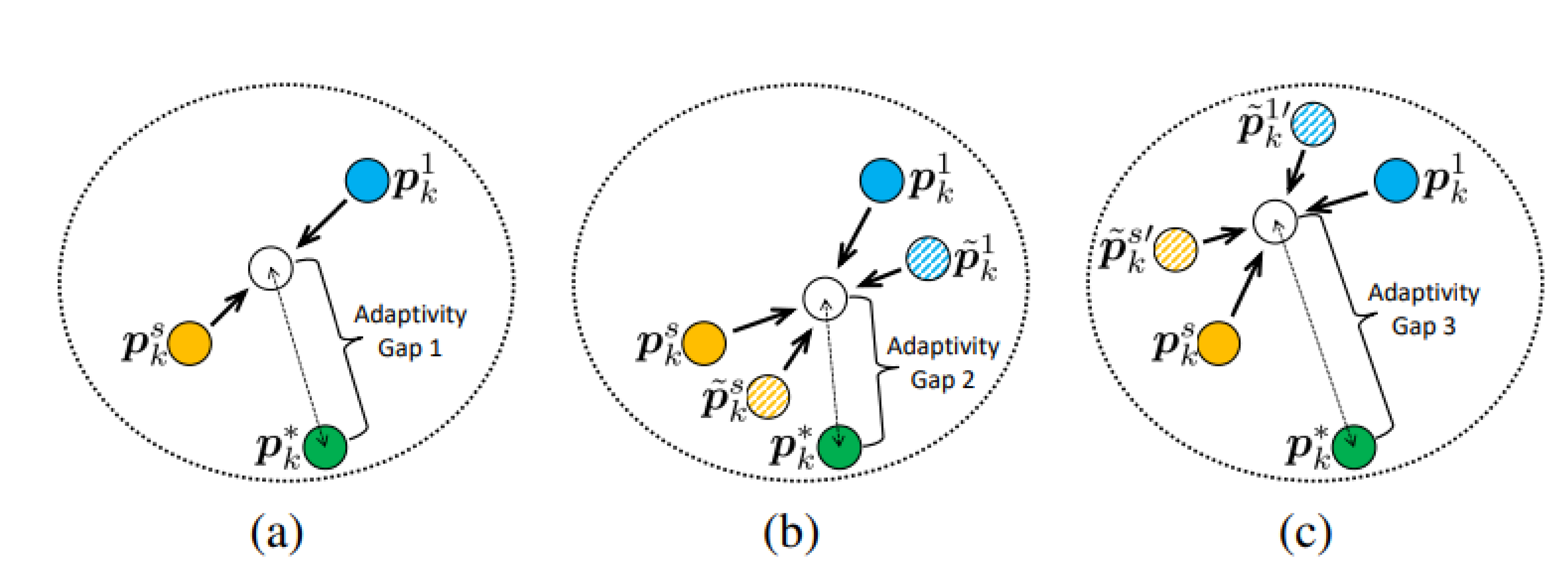
Then we construct class centroids by a prediction-weighted aggregation on representations

$$p_k = \frac{\sum_{\boldsymbol{x}_i \in \mathcal{F}} g_{\omega,k}\left(f_\theta(\boldsymbol{x}_i)\right) \cdot f_\theta(\boldsymbol{x}_i)}{\sum_{\boldsymbol{x}_i \in \mathcal{F}} g_{\omega,k}\left(f_\theta(\boldsymbol{x}_i)\right)}.$$

Then we fit a k-Nearest Neighbor (kNN) classifier and assign the pseudo label

$$\hat{y} = \mathrm{kNN}\left(\boldsymbol{x}, \mathcal{F}'\right)_{\text{Euclidean}}^{\frac{N_{\mathcal{T}} t}{r'_{\text{top}} \cdot K}},$$

PROTOTYPE CONTRASTIVE ALIGNMENT



(a) (b) (c)

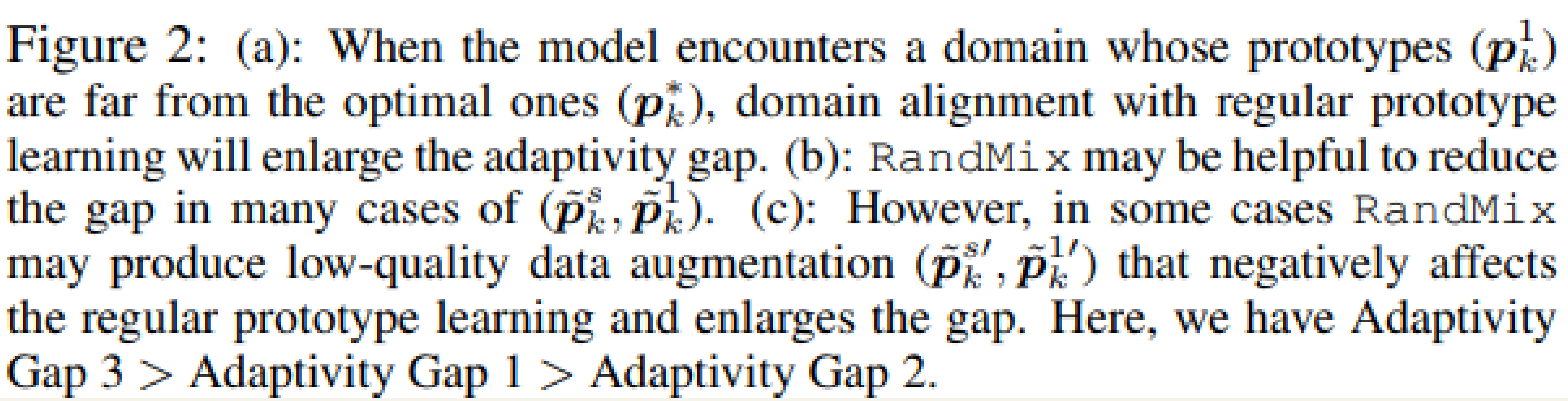


Figure 2: (a): When the model encounters a domain whose prototypes ($\boldsymbol{p}_k^1$) are far from the optimal ones ($\boldsymbol{p}_k^*$), domain alignment with regular prototype learning will enlarge the adaptivity gap. (b): `RandMix` may be helpful to reduce the gap in many cases of ($\tilde{\boldsymbol{p}}_k^s, \tilde{\boldsymbol{p}}_k^1$). (c): However, in some cases `RandMix` may produce low-quality data augmentation ($\tilde{\boldsymbol{p}}_k^{s\prime}, \tilde{\boldsymbol{p}}_k^{1\prime}$) that negatively affects the regular prototype learning and enlarges the gap. Here, we have Adaptivity Gap 3 > Adaptivity Gap 1 > Adaptivity Gap 2.

PROTOTYPE CONTRASTIVE ALIGNMENT

Use the idea of PL and the Prototype Contrastive Alignment (PCA) algorithm.

- Combines ideas from semi-supervised learning and PL to align prototypes more effectively & modifies prototypes with weighted neuron alignments for class-wise contrasts.

Cross-Entropy Loss : Regular loss for optimizing representations.

Prototype Contrastive Alignment Loss: Ensures alignment by penalizing mismatches between prototypes of the same class across domains.

$$\mathcal{L}_{\mathrm{CE}} = \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{P}_X^{\mathcal{T}^t}} \left[\sum_{k=1}^{K} -\log \frac{\mathbb{I}_{\hat{\boldsymbol{y}}_i = k} \exp\left(\boldsymbol{p}_k^{t\top} f_\theta(\boldsymbol{x}_i) + b_k\right)}{\sum_{c=1}^{K} \exp\left(\boldsymbol{p}_c^{t\top} f_\theta(\boldsymbol{x}_i) + b_c\right)} \right]$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \mathcal{L}_{\mathrm{PCA}} + \mathcal{L}_{\mathrm{DIS}}$$

$$L_{PCA} = \mathbb{E}_{x \in T_t} \left[-\log \frac{\exp\left(p_k^T f_\theta(x) + b_k\right)}{\sum_{j=1}^{K} \exp\left(p_j^T f_\theta(x) + b_j\right)} \right]$$

EXPERIMENTAL SETUP

DATASETS

- Digits: 5 domains (MNIST, SVHN, MNIST-M, SYN-D, USPS).
- PACS: 4 domains (Photo, Art, Cartoon, Sketch).
- DomainNet: 6 domains (Quickdraw, Clipart, Painting, Infograph, Sketch, Real).

METRICS

- Target Domain Generalization (TDG): Measures the initial performance in a new domain.
- Target Domain Adaptation (TDA): Assesses performance immediately after adaptation.
- Forgetting Alleviation (FA): Evaluates retention of knowledge from previously seen domains.

EXPERIMENTAL SETUP

BASELINE COMPARISONS

- Tested against multiple methods:
  - Continual Domain Adaptation (DA),
  - Source-Free Domain Adaptation,
  - Test-Time Domain Adaptation (DA),
  - Single Domain Generalization (DG),
  - Multiple Domain Generalization (DG) methods.

RESULTS ON DATASET

Digit Dataset

- TDG: RaTP surpasses all baselines with an average accuracy of 72.0%.
- TDA: RaTP achieves 86.8%, outperforming SHOT (67.1%) and Tent (59.9%).
- FA: RaTP scores 82.3%, significantly exceeding CoTTA (48.8%) and AuCID (42.4%).

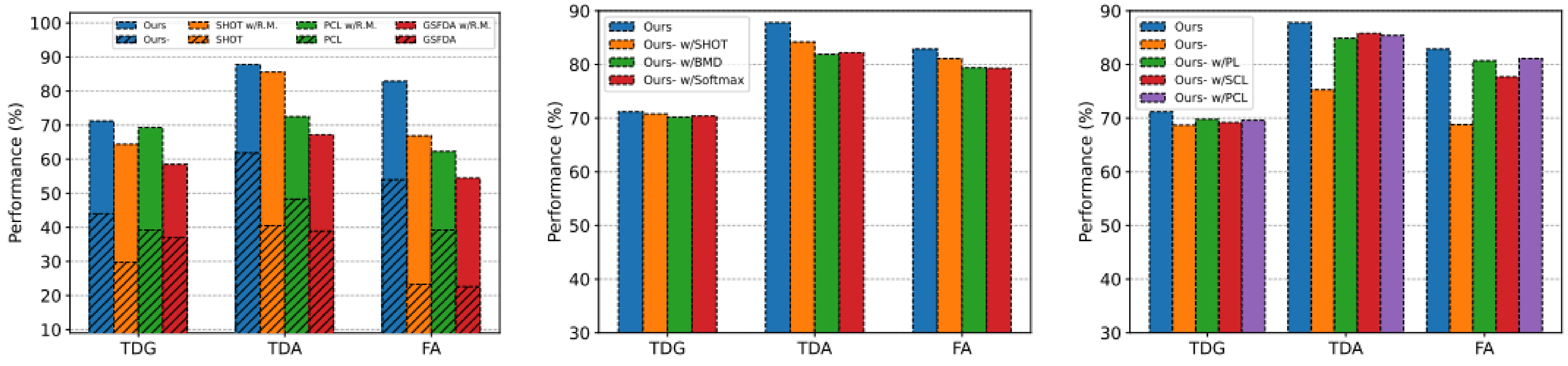
PACS Dataset

- TDG: RaTP achieves 62.6%, outperforming CoTTA (54.8%) and PDEN (50.1%).
- TDA: RaTP excels with 82.3%, while the closest competitor, PDEN, achieves 75.8%.
- FA: RaTP achieves 86.2%, compared to PDEN's 73.9%.

DomainNet Dataset

- TDG: RaTP achieves 41.5%, surpassing CoTTA (34.8%) and AuCID (30.4%).
- TDA: RaTP leads with 56.8%, surpassing Tent (50.0%) and T3A (49.0%).
- FA: RaTP achieves 53.7%, outperforming CoTTA (48.0%) and L2D (44.0%).

ABLATION STUDY



(a) RandMix (R.M.)

(b) T2PL

(c) PCA

Purpose: Evaluate the impact of each component on overall performance.

Findings:

- **RandMix: Removing this component decreases TDG performance from 72.0% to 48.7% on the Digit dataset.**
- **T2PL: Without T2PL, TDA accuracy drops from 86.8% to 67.1% on the Digit dataset.**
- **PCA: Excluding PCA reduces FA performance from 82.3% to 48.0% on the Digit dataset.**

CONCLUSION

Extensive experiments demonstrate that RaTP can significantly improve the model performance in the Unfamiliar Period, while also achieving good performance in target domain adaptation and forgetting alleviation. RaTP can have several promising future applications, particularly in areas requiring robust generalization and adaptability to unseen data domains.