Advancements in **ML** via **Efficient** Generative Modeling, **Robust** Domain Adaptation, and **Explainable** Multimodal Retrieval

Name: Prasanna Reddy Pulakurthi
Advisor: Dr. Majid Rabbani
Co-advisor: Dr. Sohail Dianat
Date: 11/21/2025

This work was partially supported by a DEVCOM ARL contract

RIT

Publications

1. **Prasanna Reddy Pulakurthi**, Jiamian Wang, Majid Rabbani, Sohail Dianat, Raghuveer Rao, Zhiqiang Tao. "X-CoT: Explainable Text-to-Video Retrieval via Large Language Models (LLM) Based Chain-of-Thought Reasoning." *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (**EMNLP**)*, 2025.

2. **Prasanna Reddy Pulakurthi**, Majid Rabbani, Jamison Heard, Sohail Dianat, Celso M. de Melo, Raghuveer M. Rao. "Shuffle PatchMix Augmentation with Confidence-Margin Weighted Pseudo-Labels for Enhanced Source-Free Domain Adaptation." *Proceedings of the IEEE International Conference on Image Processing (**ICIP**)*, IEEE, 2025.

3. **Prasanna Reddy Pulakurthi**, Majid Rabbani, Celso M. de Melo, Sohail A. Dianat, Raghuveer M. Rao. "Effective Dual-Region Augmentation for Reduced Reliance on Large Amounts of Labeled Data." *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications III*, Vol. 13459, pp. 210–218, SPIE, 2025.

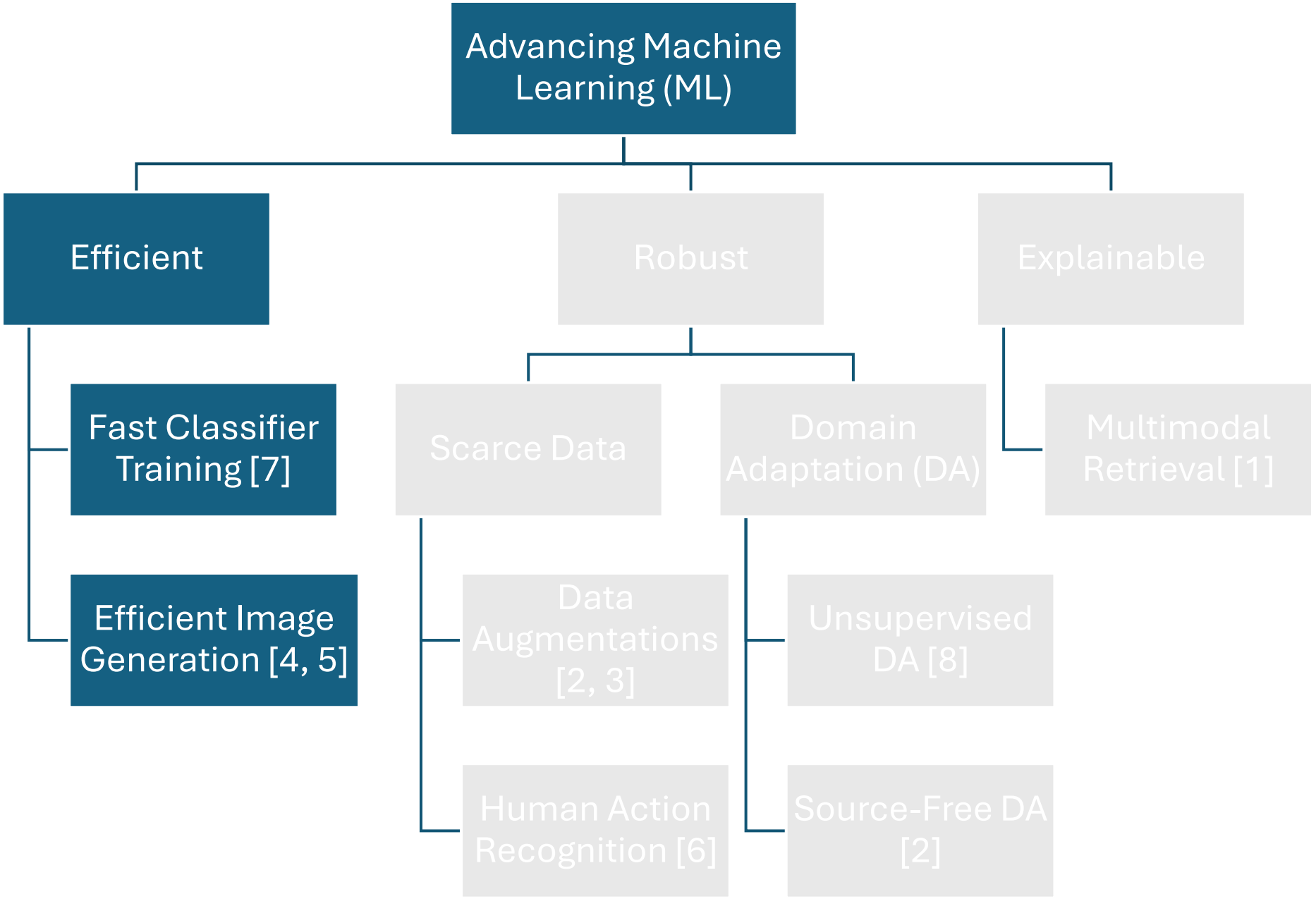
4. **Prasanna Reddy Pulakurthi**, Mahsa Mozaffari, Sohail A. Dianat, Jamison Heard, Raghuveer M. Rao, Majid Rabbani. "Enhancing GANs with MMD Neural Architecture Search, PMish Activation Function, and Adaptive Rank Decomposition." ***IEEE Access Journal***, Vol. 12, pp. 174222–174244, 2024.

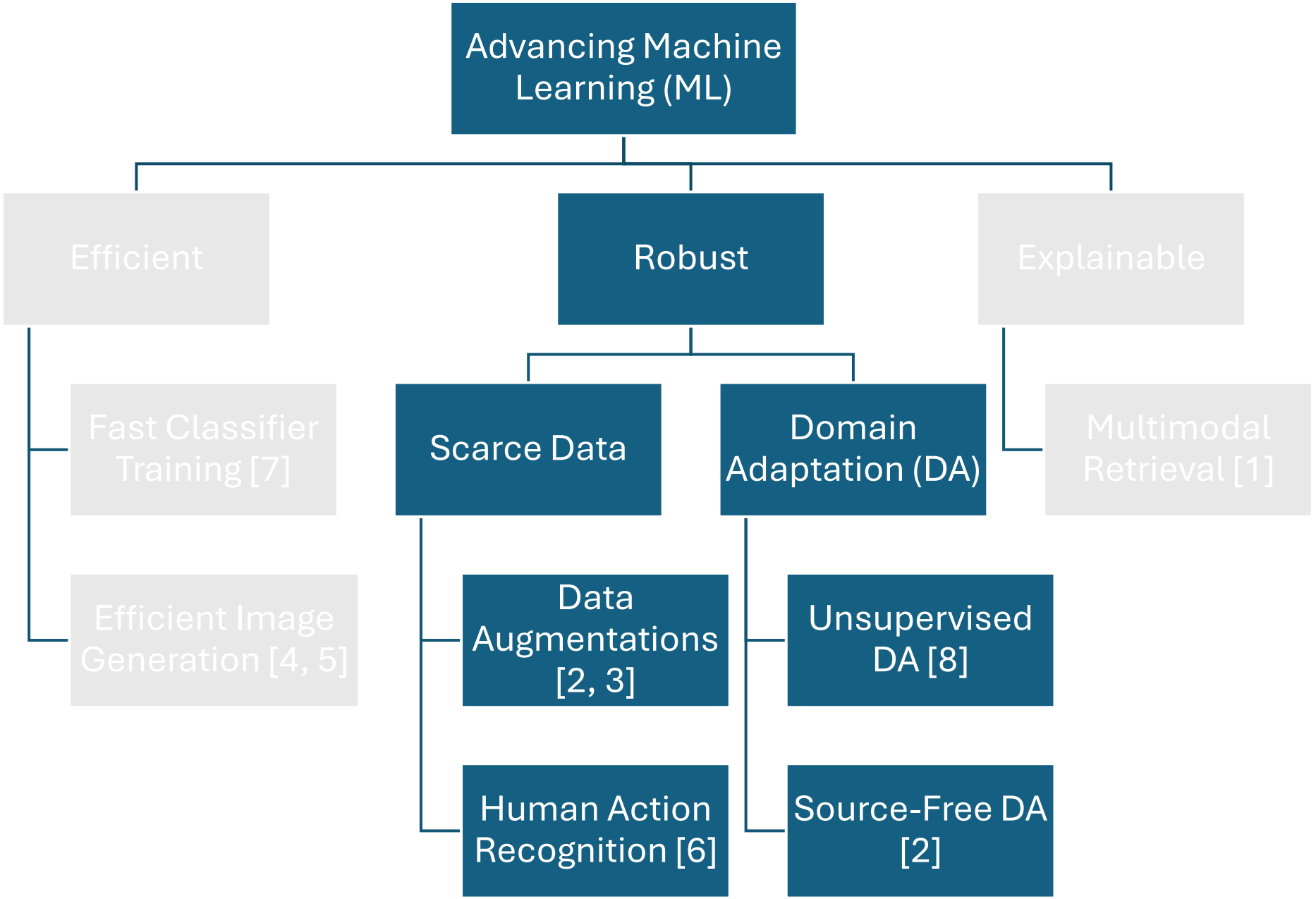
5. **Prasanna Reddy Pulakurthi**, Mahsa Mozaffari, Sohail A. Dianat, Majid Rabbani, Jamison Heard, Raghuveer M. Rao. "Enhancing GAN Performance through Neural Architecture Search and Tensor Decomposition." *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (**ICASSP**)*, pp. 7280–7284, 2024.

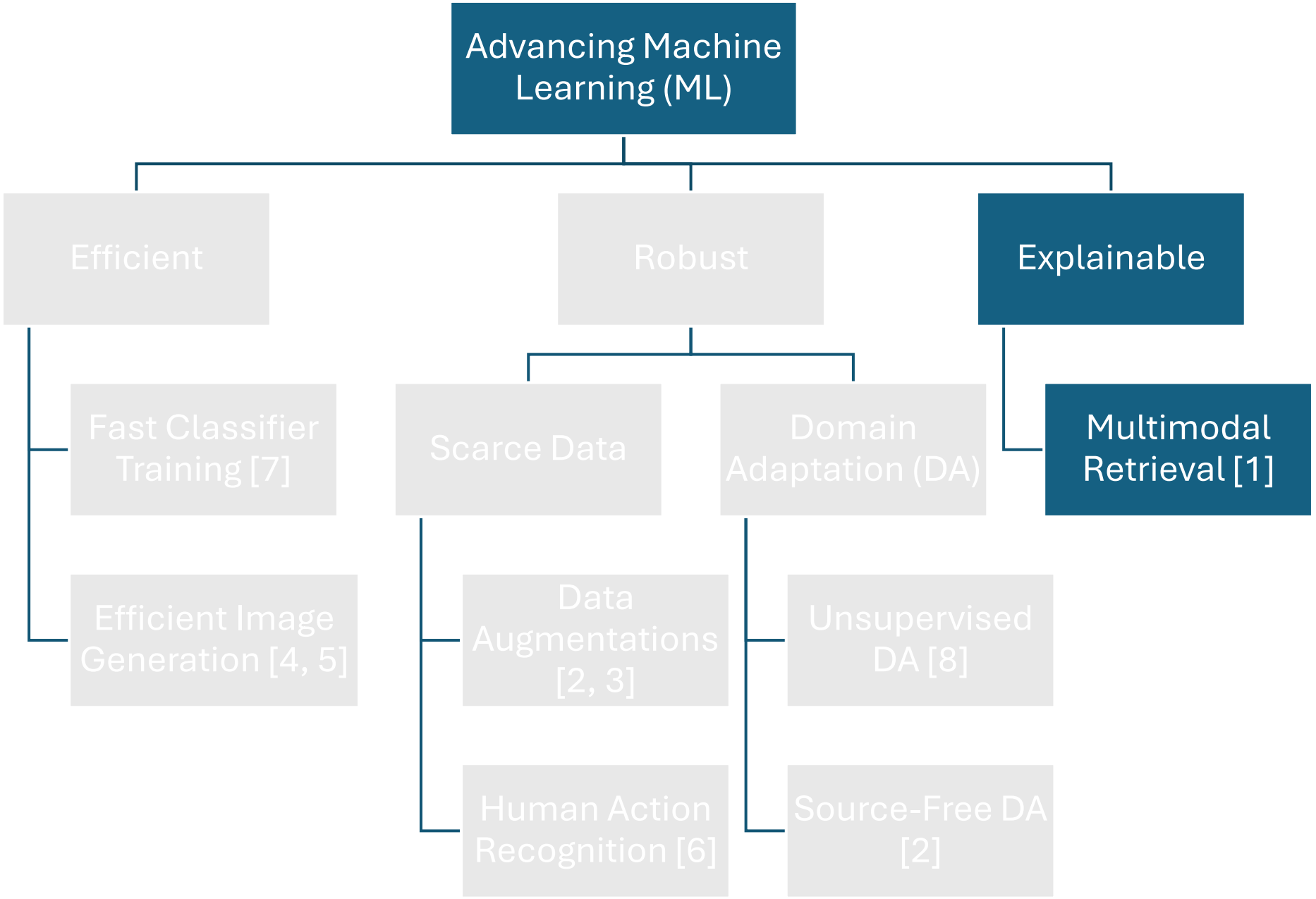
6. **Prasanna Reddy Pulakurthi**, Celso M. de Melo, Raghuveer M. Rao, Majid Rabbani. "Enhancing Human Action Recognition with GAN-Based Data Augmentation." *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*, Vol. 13035, pp. 194–204, SPIE, 2024.

7. **Prasanna Reddy Pulakurthi**, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "A Globally Optimal Fast Iterative Linear Maximum Likelihood Classifier." *Electronic Imaging*, Vol. 35, pp. 1–5, 2023.

8. **Prasanna Reddy Pulakurthi**, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "Unsupervised Domain Adaptation Using Feature-Aligned Maximum Classifier Discrepancy." *Applications of Machine Learning 2022*, Vol. 12227, pp. 37–45, SPIE, 2022.

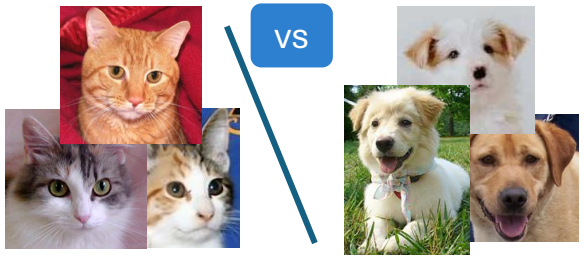

Advancing Machine Learning (ML)
Efficient
Robust
Explainable
Fast Classifier Training [7]
Efficient Image Generation [4, 5]
Scarce Data
Domain Adaptation (DA)
Multimodal Retrieval [1]
Data Augmentations [2, 3]
Human Action Recognition [6]
Unsupervised DA [8]
Source-Free DA [2]


Advancing Machine Learning (ML)
Efficient
Robust
Explainable
Fast Classifier Training [7]
Efficient Image Generation [4, 5]
Scarce Data
Domain Adaptation (DA)
Multimodal Retrieval [1]
Data Augmentations [2, 3]
Human Action Recognition [6]
Unsupervised DA [8]
Source-Free DA [2]

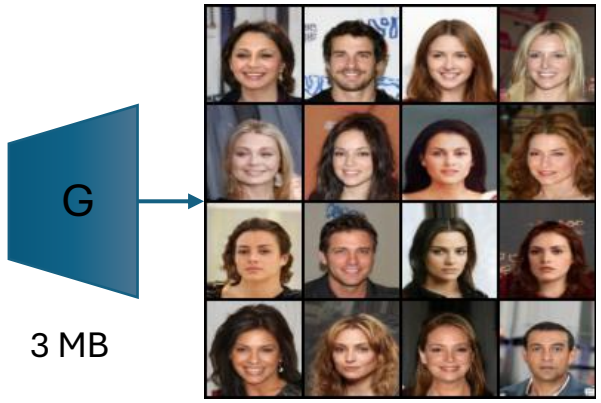

Advancing Machine Learning (ML)
Efficient
Robust
Explainable
Fast Classifier Training [7]
Efficient Image Generation [4, 5]
Scarce Data
Domain Adaptation (DA)
Multimodal Retrieval [1]
Data Augmentations [2, 3]
Human Action Recognition [6]
Unsupervised DA [8]
Source-Free DA [2]

Efficient

Fast Classifier Training



Efficient Image Generation



PMish + ARD → Efficient GANs
(Params ↓, FID ≈)

Robust

Data Augmentation

Shuffle PatchMix (SPM)

Random Patch Shuffle + Blend

Dual Region Augmentation

Foreground Noise + Back Patch Shuffle

Domain Adaptation

Synthetic Data

Real Data

Synthetic → Real (No Labels!)

Explainable

Text-to-Video Retrieval

Text Query

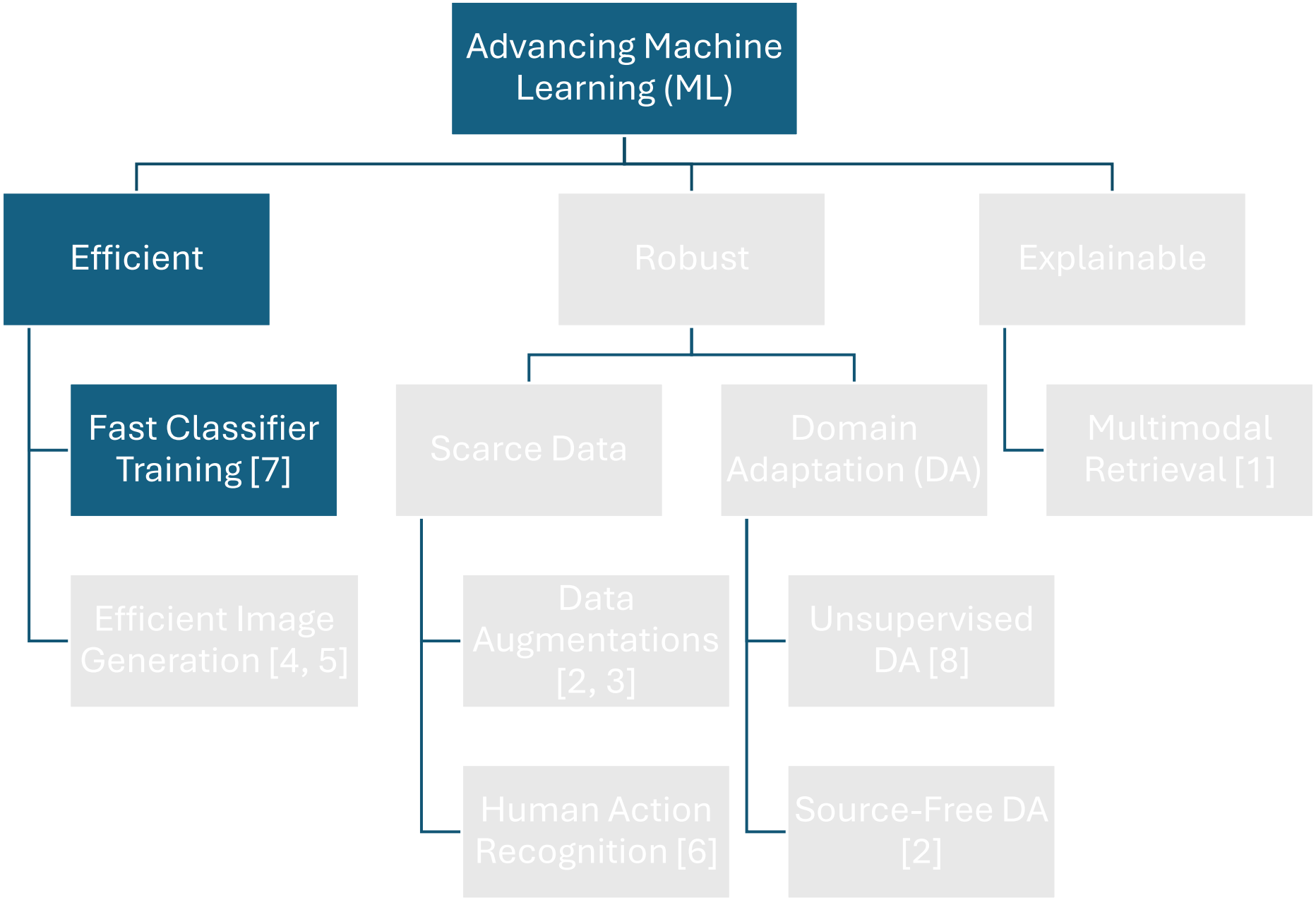
Video Pool

Which video matches the query and ***why***?

Video Ranks + Explanations

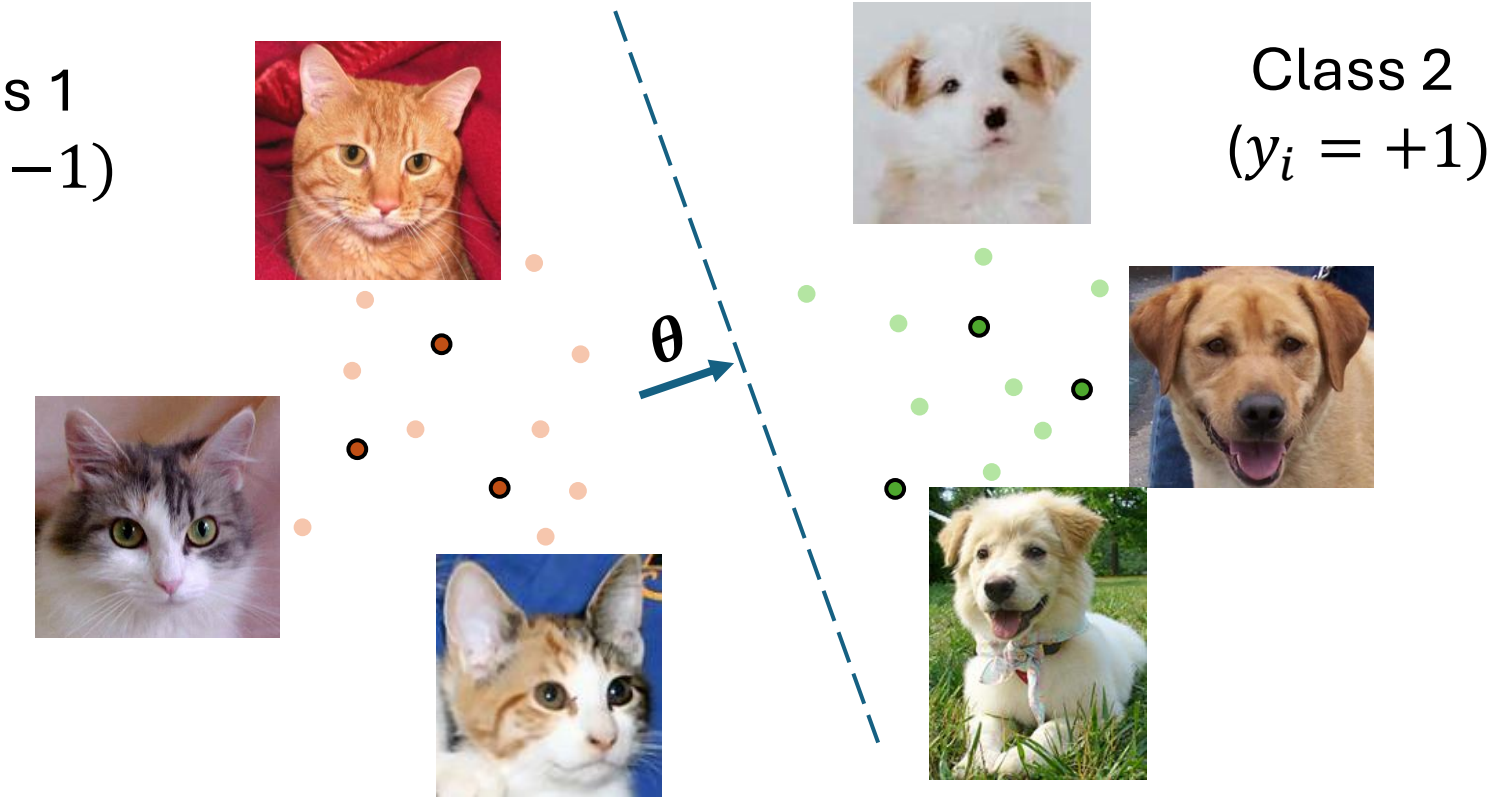
1. Video 12
2. Video 05
3. Video 07
...

Reason: query mentions 'red bus'; Video 12 shows a red double-decker.



7. Prasanna Reddy Pulakurthi, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "A Globally Optimal Fast Iterative Linear Maximum Likelihood Classifier." *Electronic Imaging*, Vol. 35, pp. 1–5, 2023.

Fast Classifier Training

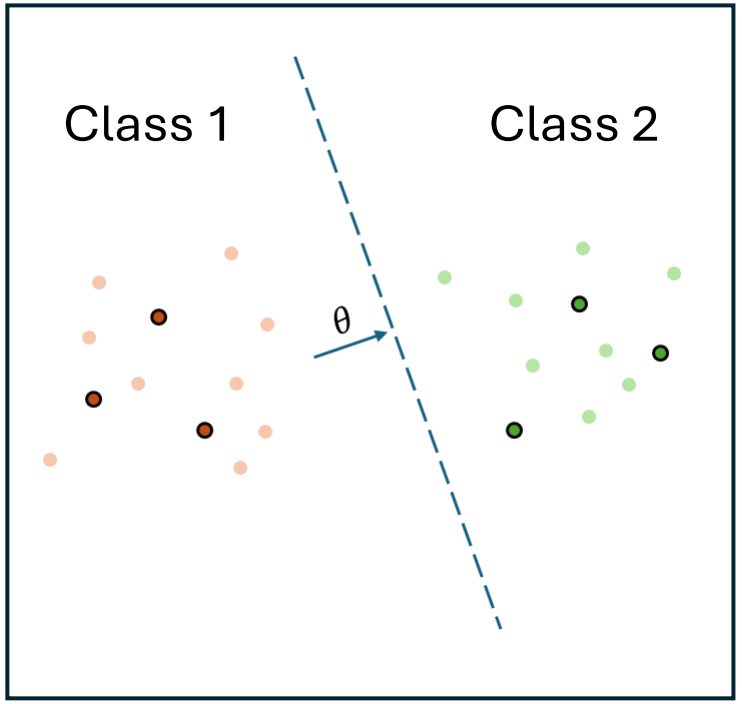


- A linear classifier aims at finding a separating hyperplane: $z_i = \boldsymbol{\theta}^T \boldsymbol{x}_i + b$
- A test sample is assigned to a class using: $\widehat{y_i} = \text{sign}(z_i)$

Solving for the Model Parameters

- The model parameters $\boldsymbol{\theta}$ are found by minimizing the negative log likelihood loss with an added regularization term ($\lambda \, ||\boldsymbol{\theta}||^2$) to prevent overfitting.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left[\sum_{i=1}^{N} \ln\left(1 + \exp\left(-y_i(\boldsymbol{\theta}^T \boldsymbol{x}_i + b)\right)\right) + \lambda \, ||\boldsymbol{\theta}||^2 \right]$$



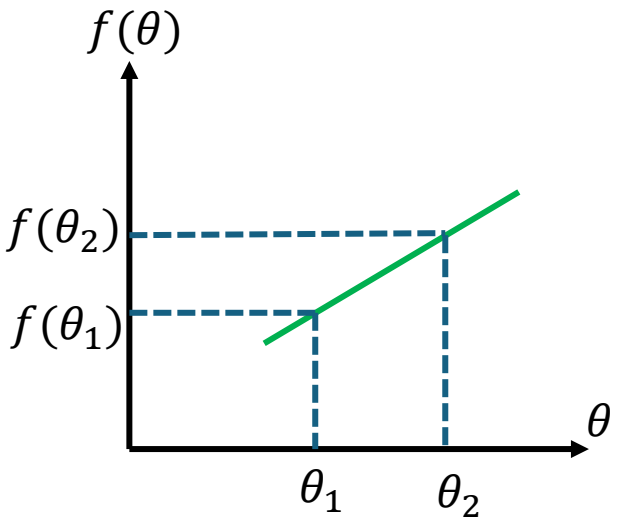
Contraction Mapping

- $f(\boldsymbol{\theta})$ is contraction mapping if it satisfies,

  $\|f(\boldsymbol{\theta}_2) - f(\boldsymbol{\theta}_1)\| < \rho \, \|\boldsymbol{\theta}_2 - \boldsymbol{\theta}_1\|$

  and $0 \leq \rho < 1$.

- If $f(\boldsymbol{\theta})$ is a contraction mapping, then the iterative solution $\boldsymbol{\theta}(n+1) = f(\boldsymbol{\theta}(n))$ always converges to the fixed-point solution.



Solving for the Model Parameters

- Setting the derivative of the loss function with respect to $\boldsymbol{\theta}$ equal to zero yields the following equation:.

$$\boldsymbol{\theta} = \frac{1}{2\lambda N}\sum_{i=1}^{N}\frac{y_i \boldsymbol{x}_i \exp(-y_i(\boldsymbol{\theta}^T \boldsymbol{x}_i + b))}{1+\exp(-y_i(\boldsymbol{\theta}^T \boldsymbol{x}_i + b))} = f(\boldsymbol{\theta})$$

- We prove that the above function would be a **contraction mapping** for certain values of λ.

$$\lambda > \frac{1}{8N}\sum_{i=1}^{N} ||\boldsymbol{x}_i||^2$$

Pseudo Code for Finding Model Parameters

Algorithm 1 Proposed Linear Iterative Maximum Likelihood Classifier Solution.

Input: N number of samples, N_E number of iterations.

Initialize $n = 0$, parameter vector $\boldsymbol{\theta}(0)$ by a small random value, and Set $\lambda = (1+\epsilon)\ \frac{1}{8N} \sum_{i=1}^{N} \|\boldsymbol{x}_i\|^2$.

1: **while** $n < N_E$ **do**

2: Sample the data $\{\boldsymbol{x}_i\}_{i=1}^{N}$ and its corresponding labels $\{y_i\}_{i=1}^{N}$.

3: $\boldsymbol{\theta}(n+1) \leftarrow \frac{1}{2\lambda N} \sum_{i=1}^{N} \frac{y_i \boldsymbol{x}_i \exp(-y_i(\boldsymbol{\theta}(n)^T \boldsymbol{x}_i + b))}{1+\exp(-y_i(\boldsymbol{\theta}(n)^T \boldsymbol{x}_i + b))}$

4: $b \leftarrow \frac{1}{N} \sum_{i=1}^{N} (y_i - \boldsymbol{\theta}(n)^T \boldsymbol{x}_i)$

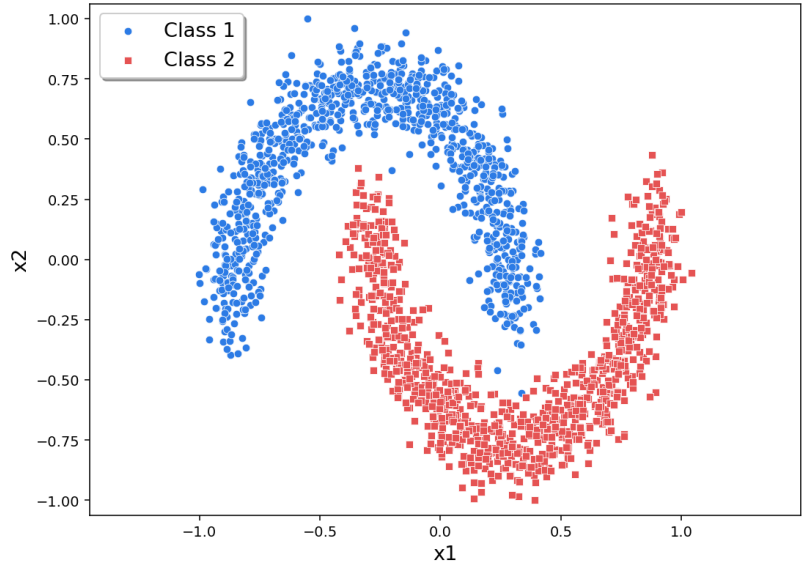
5: $n \leftarrow n + 1$

6: **end while**

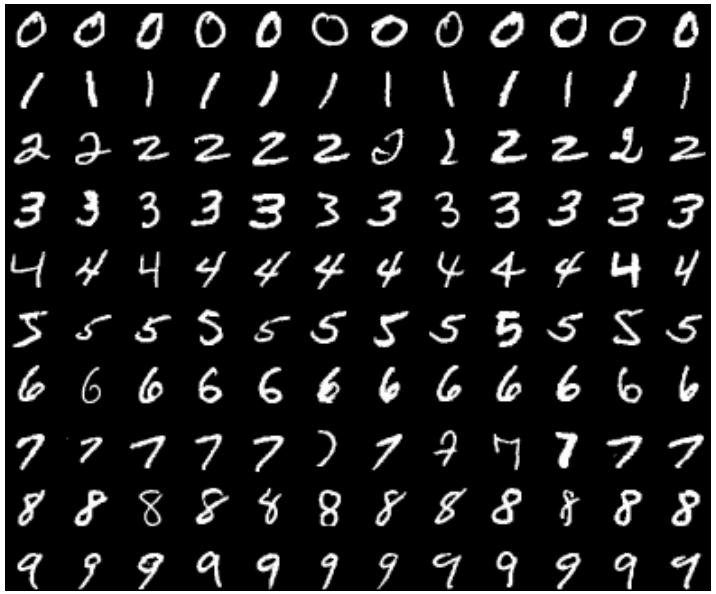
Experiments

1. Binary classification on synthetic two moons dataset.

2. Multiclass Image classification on the handwritten MNIST digits dataset.



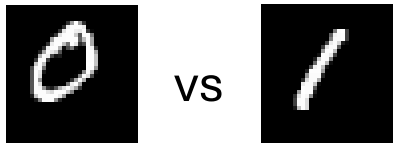
Synthetic Two Moons Dataset



MNIST Dataset

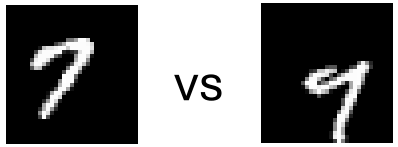
Image Classification Results

- **Binary Classification:**



Accuracy

99.3%



91.3%

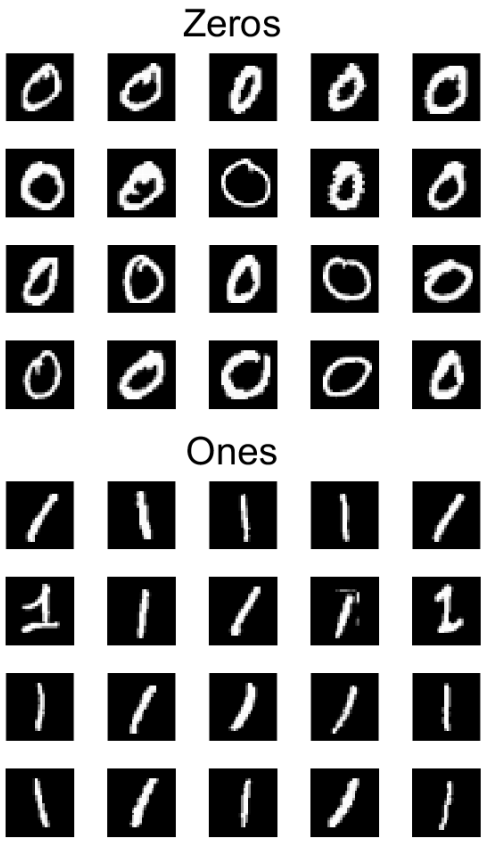
- **Multiclass Classification:** Using all-vs-all classifiers with majority voting, an accuracy of **84.0%** is achieved.

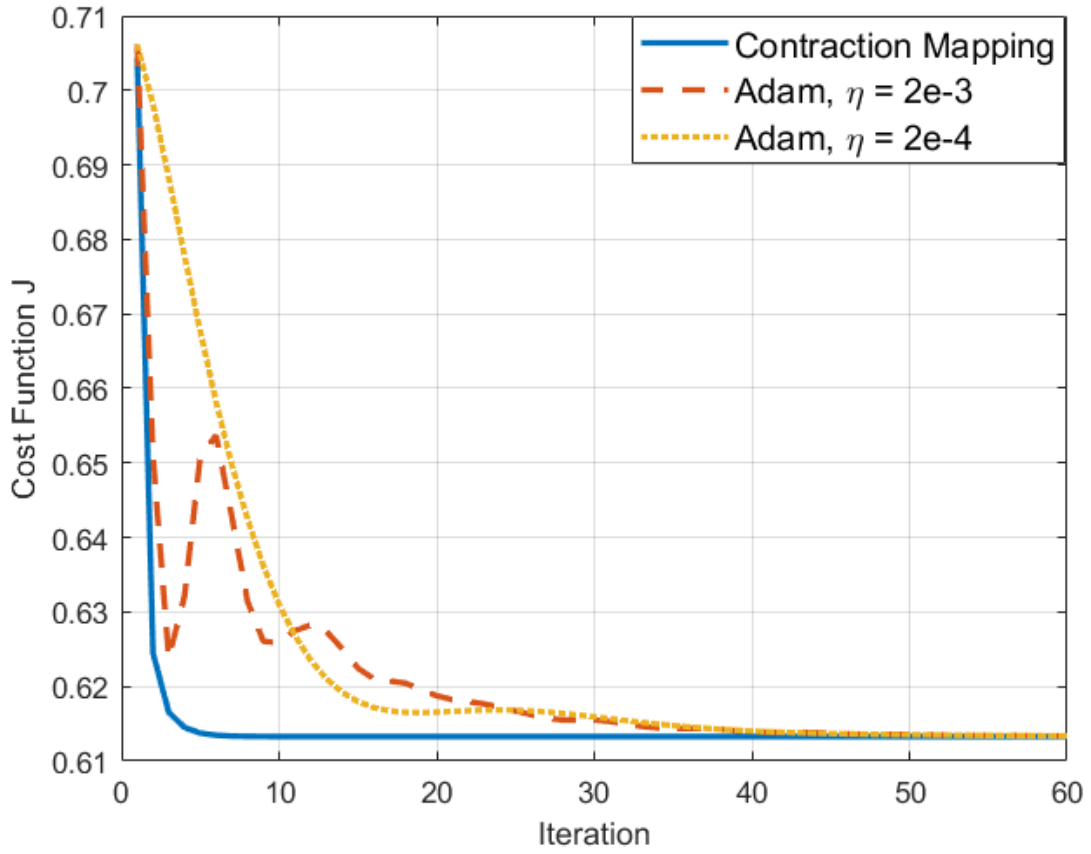
	0	1	2	3	4	5	6	7	8	9
0	–	99.3	96.8	97.1	97.9	93.8	95.1	98.5	95.5	96.8
1	99.3	–	96.3	97.4	98.4	97.7	98.0	96.2	96.2	98.6
2	96.8	96.3	–	93.4	96.8	95.9	95.9	96.6	94.2	97.0
3	97.1	97.4	93.4	–	98.8	91.3	98.7	96.4	92.7	96.7
4	97.9	98.4	96.8	98.8	–	97.5	97.6	96.7	97.6	91.5
5	93.8	97.7	95.9	91.3	97.5	–	96.2	97.7	94.3	96.1
6	95.1	98.0	95.9	98.7	97.6	96.2	–	99.2	97.4	99.2
7	98.5	96.2	96.6	96.4	96.7	97.7	99.2	–	96.2	91.3
8	95.5	96.2	94.2	92.7	97.6	94.3	97.4	96.2	–	95.8
9	96.8	98.6	97.0	96.7	91.5	96.1	99.2	91.3	95.8	–

Pairwise classification accuracies (%) using IMLC† for all-vs-all binary classifiers among 0 to 9 MNIST digits.

†**Prasanna Reddy Pulakurthi**, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "A Globally Optimal Fast Iterative Linear Maximum Likelihood Classifier." *Electronic Imaging*, 2023.

Fast Classifier vs Gradient Descent

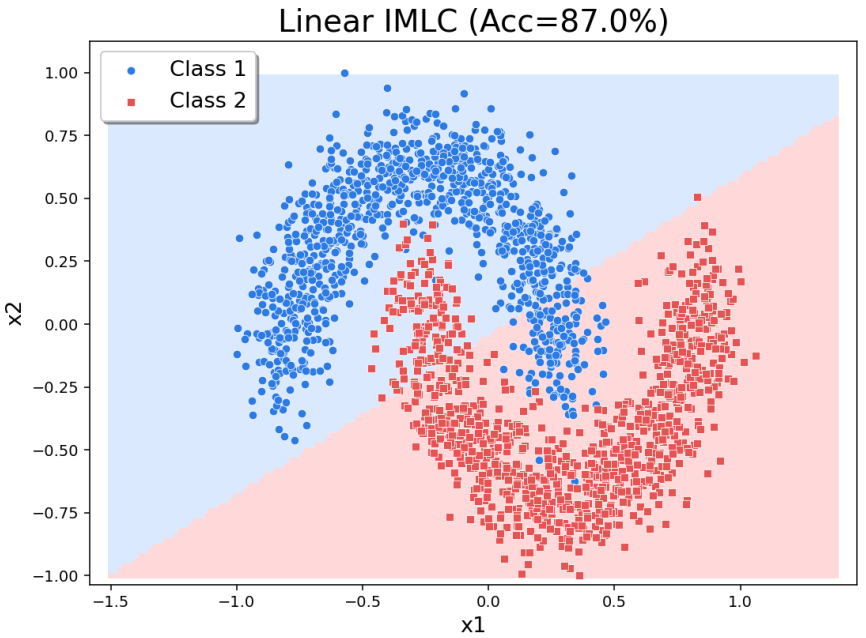


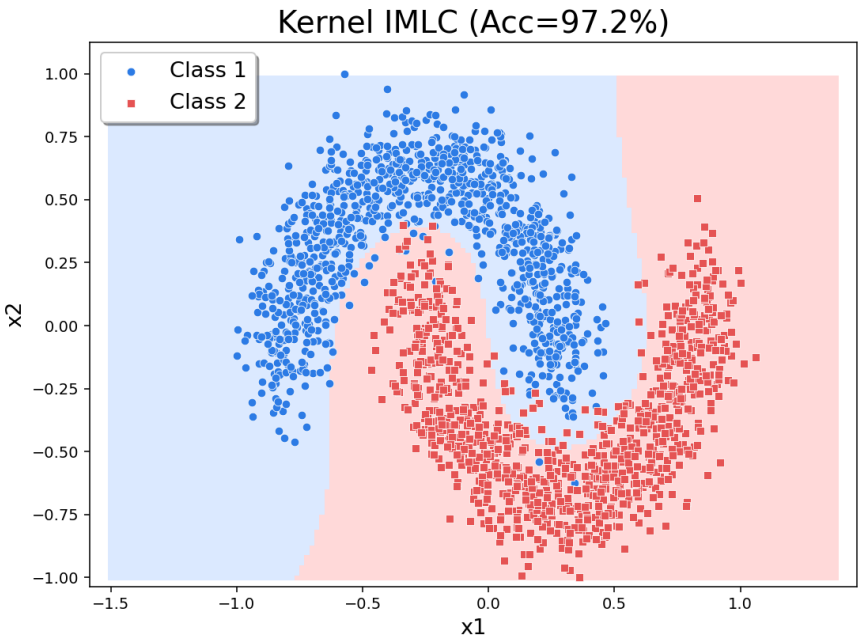


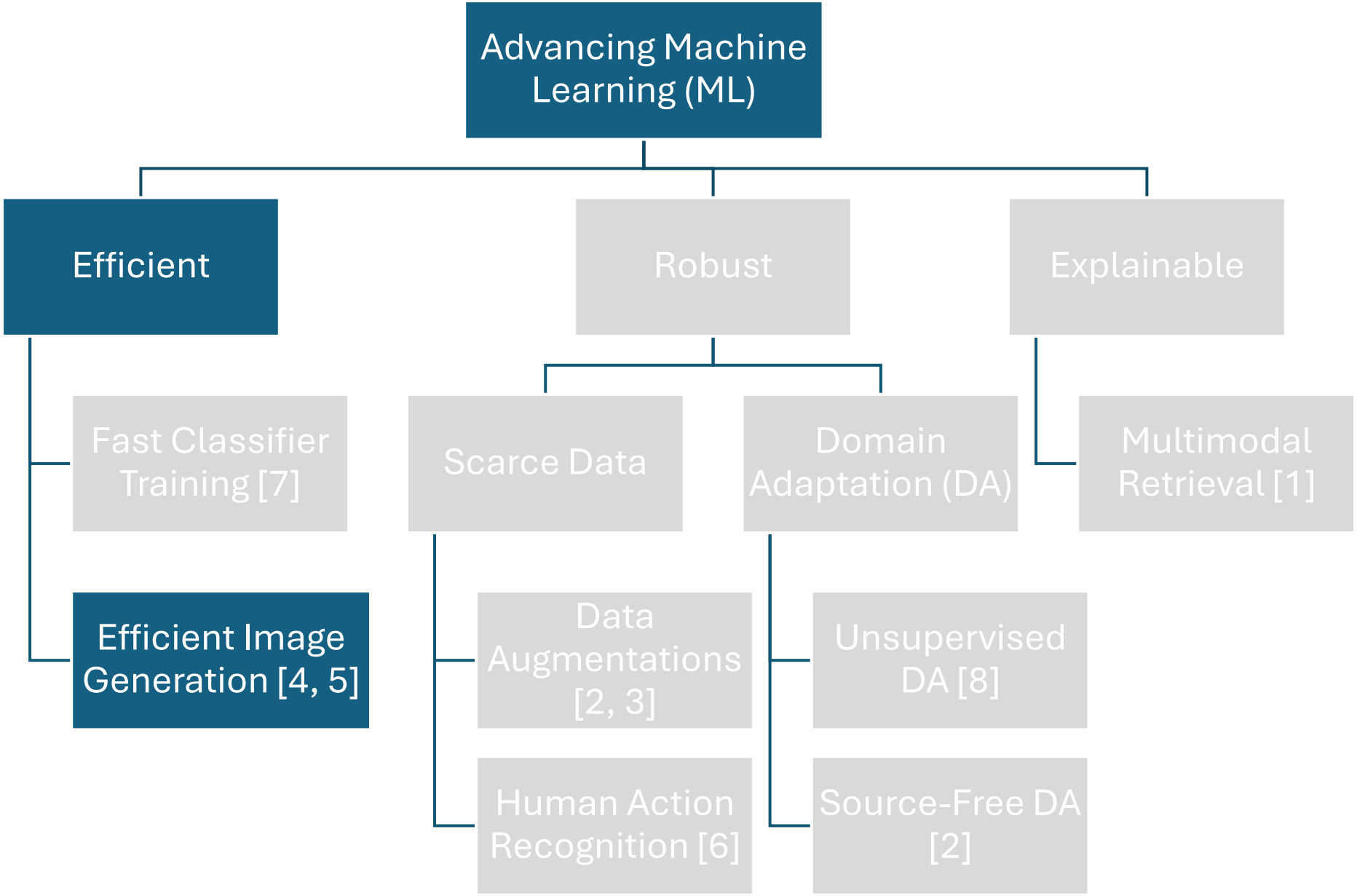
†**Prasanna Reddy Pulakurthi**, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "A Globally Optimal Fast Iterative Linear Maximum Likelihood Classifier." *Electronic Imaging*, 2023.

Extending to Non-Linear Classification

- Using the Kernel trick, we can extend this to non-linear settings.
- **Multiclass Classification:** Using this on the MNIST digits dataset, the classification accuracy increases from 84.0% to **88.4% (+4.4%)**.



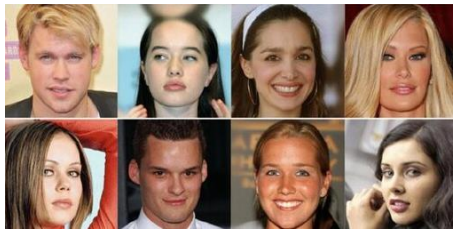




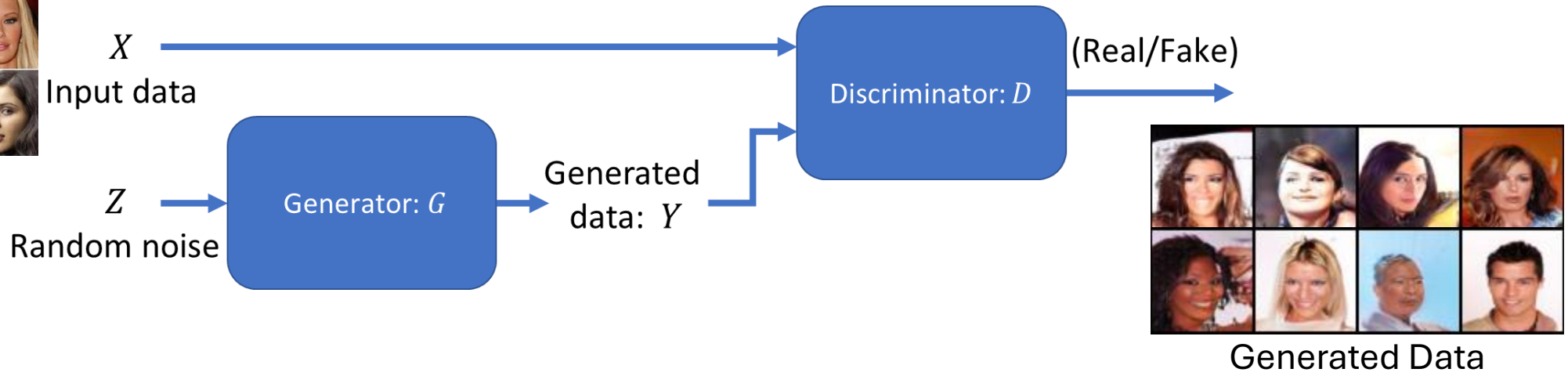
4. Prasanna Reddy Pulakurthi, Mahsa Mozaffari, Sohail A. Dianat, Jamison Heard, Raghuveer M. Rao, Majid Rabbani. "Enhancing GANs with MMD Neural Architecture Search, PMish Activation Function, and Adaptive Rank Decomposition." ***IEEE Access Journal***, Vol. 12, pp. 174222–174244, 2024.

5.Prasanna Reddy Pulakurthi, Mahsa Mozaffari, Sohail A. Dianat, Majid Rabbani, Jamison Heard, Raghuveer M. Rao. "Enhancing GAN Performance through Neural Architecture Search and Tensor Decomposition." *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (**ICASSP**)*, pp. 7280–7284, 2024.

GANs† for Image Generation



Real Data



Generated Data

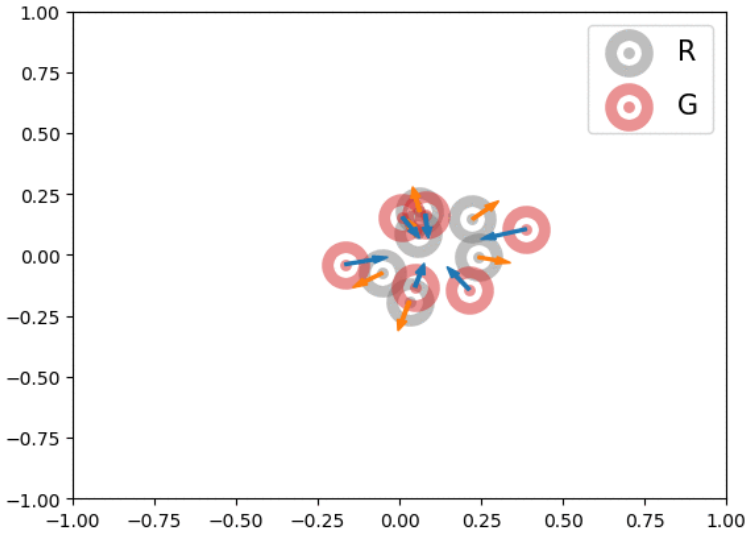
- Generative Adversarial Networks (GANs) generate adversarial samples from a desired data distribution.
- The learning is accomplished via a two-player minimax optimization game between the generator and the discriminator.
- The generator network G aims at generating adversarial samples that fool the discriminator into thinking they are real, and the discriminator network D tries to distinguish between the real and fake samples.

†Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.

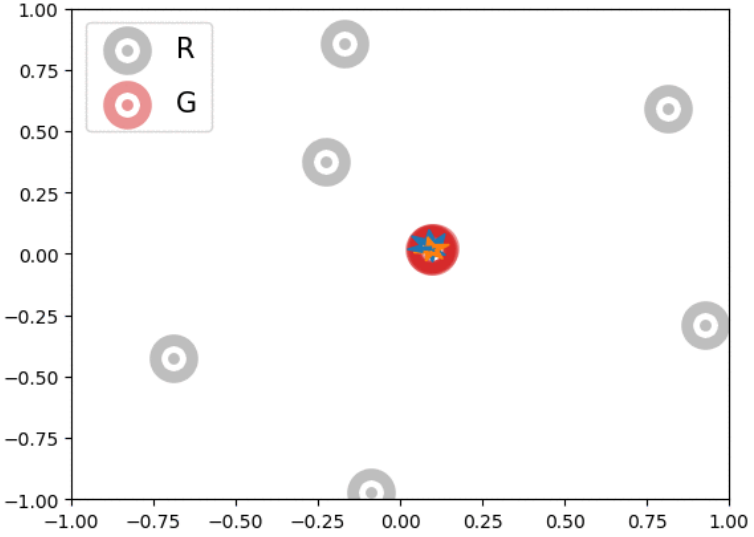
Background: Loss Function

- Training objective: Bounded MMD-GAN Repulsive loss†. Real P_X and Generated P_Y.
- Discriminator: $\min_D E_{P_X}[k_D(x,x')] - E_{P_Y}[k_D(y,y')]$, where the discriminator is subjected to Lipschitz constraint.
- Generator : $\min_G E_{P_X}[k_D(x,x')] + E_{P_Y}[k_D(y,y')] - 2E_{P_X,P_Y}[k_D(x,y)]$ (MMD Loss).

Discriminator Training



Generator Training

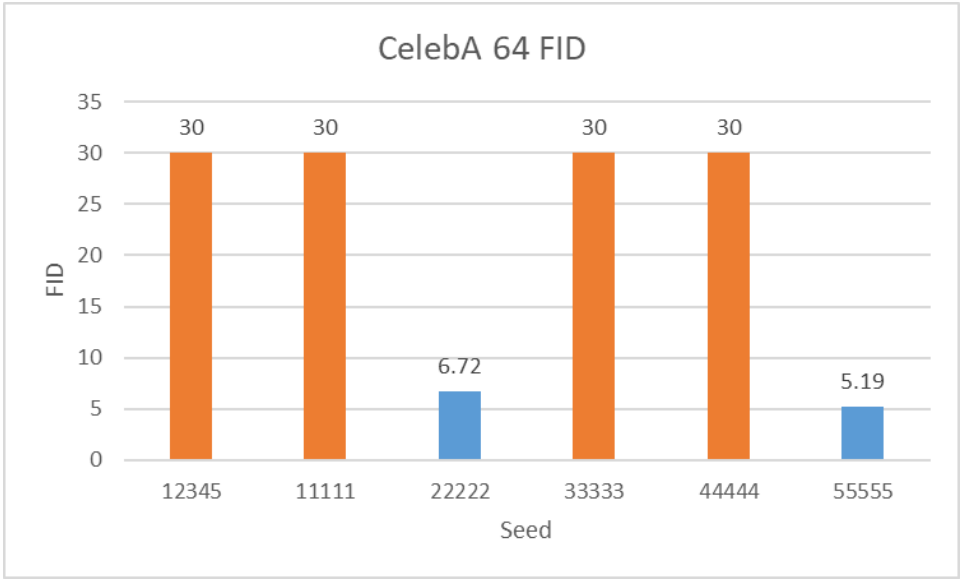


R: Real
G: Generated

†Wang, Wei, Yuan Sun, and Saman Halgamuge, "Improving MMD-GAN training with repulsive loss function." *arXiv preprint arXiv:1812.09916* (2018).

Drawbacks of Bounded MMD-GAN Loss†

1. In the original work, the upper bound is **fixed to 4**, which is **not optimal**.
2. Inconsistent convergence for large architectures.



Our Solution:

1. **Training Strategy:** Change the Upper bound while training to improve performance and achieve stable convergence.
2. **Modified Loss:** Propose a modification to the loss function to make the model consistently converge.

†Wang, Wei, Yuan Sun, and Saman Halgamuge, "Improving MMD-GAN training with repulsive loss function." *arXiv preprint arXiv:1812.09916* (2018).

Thesis Contribution to Improving GAN Efficiency

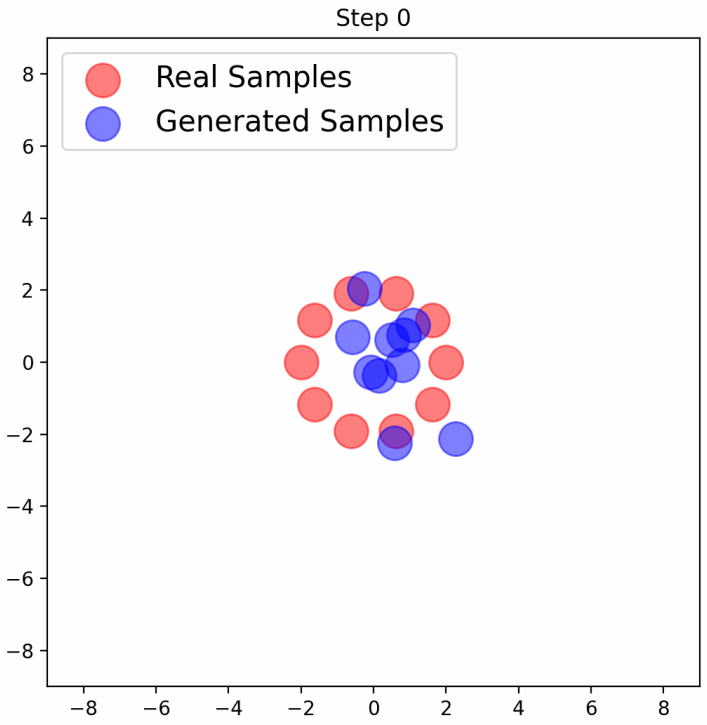
1. Loss Function: Training Strategy
2. Loss Function: Modified MMD-GAN rep Loss
3. Activation Function: Introduced PMish, an adaptive activation
4. GAN Compression: Adaptive Rank Selection for Tensor Decomposition

[†] Pulakurthi, Prasanna Reddy, et al. "Enhancing GANs with MMD Neural Architecture Search, PMish Activation Function and Adaptive Rank Decomposition." *IEEE Access* (2024).

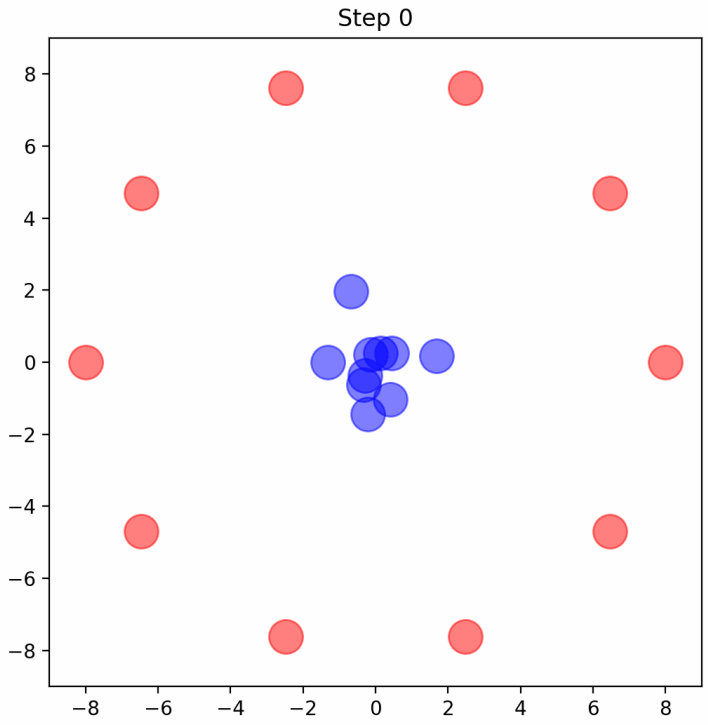
1. Training Strategy

- We argue that a **fixed upper bound** is **not** optimal for determining the optimal solution.
- A **small upper bound** constricts real samples close to each other, hindering the generator from learning fine-level details.
- Conversely, a very **large upper bound** results in a large gradient to the generator, making the training **unstable** and leading to suboptimal performance.
- Therefore, this thesis proposes that the upper bound progressively increases when performance saturation is observed.

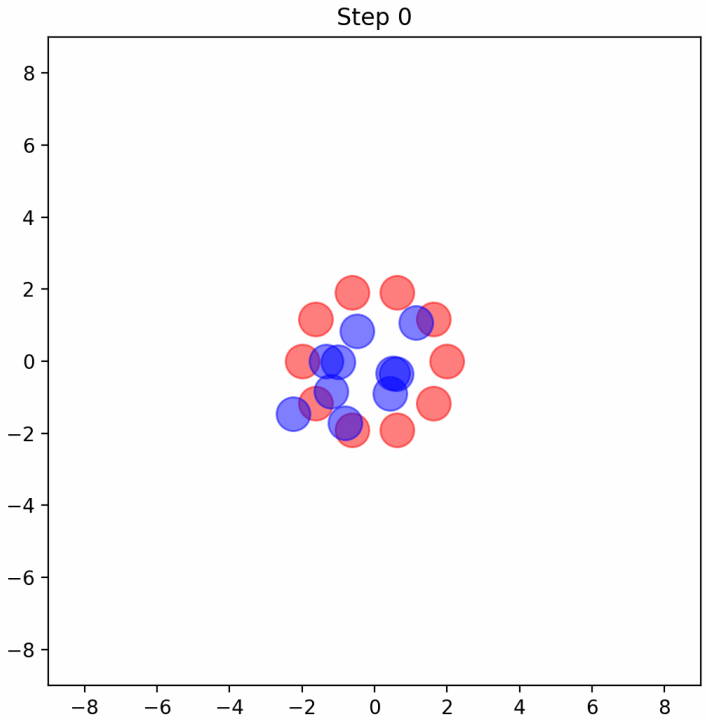
1. Training Strategy



Small Upper Bound



Large Upper Bound



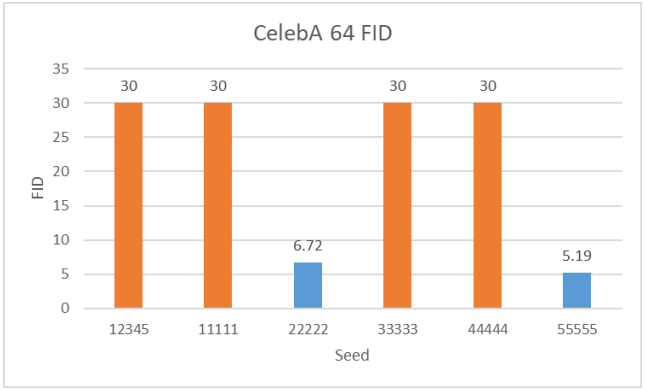
Progressively Increasing Upper Bound (Proposed)

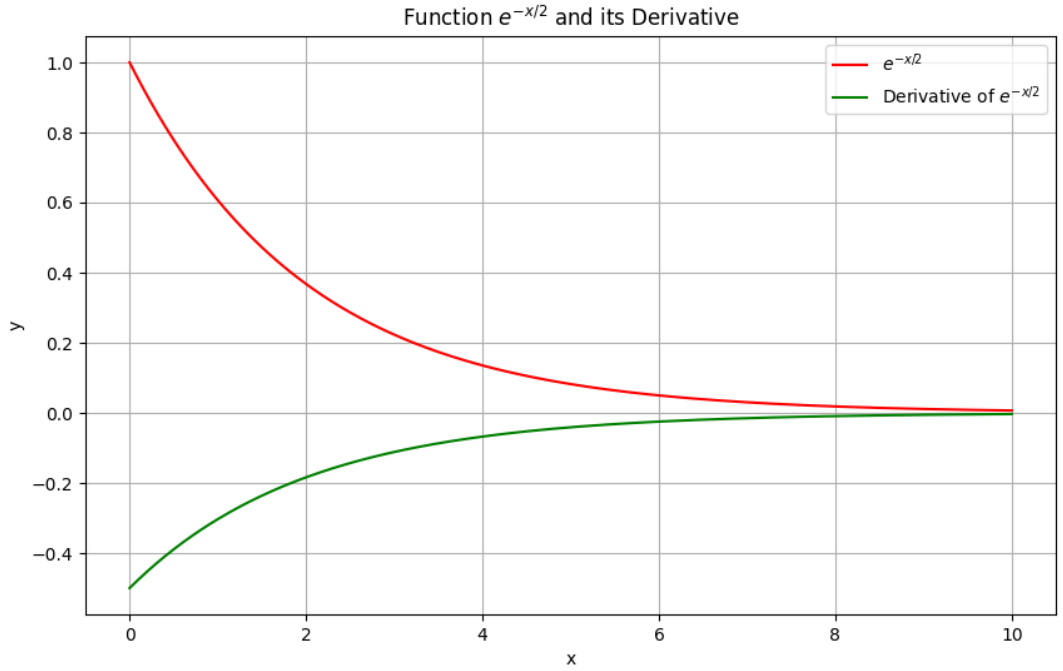
$$k_\sigma^{\text{rbf-b}}(\boldsymbol{u},\boldsymbol{v}) = \begin{cases} \exp\left(-\frac{1}{2\sigma^2}\max(\|\boldsymbol{u}-\boldsymbol{v}\|^2, b_l)\right) \boldsymbol{u},\boldsymbol{v} \in \boldsymbol{D}(\boldsymbol{Y}) \\ \exp\left(-\frac{1}{2\sigma^2}\min(\|\boldsymbol{u}-\boldsymbol{v}\|^2, b_u)\right) \boldsymbol{u},\boldsymbol{v} \in \boldsymbol{D}(\boldsymbol{X}), \end{cases}$$

2. Modified Loss

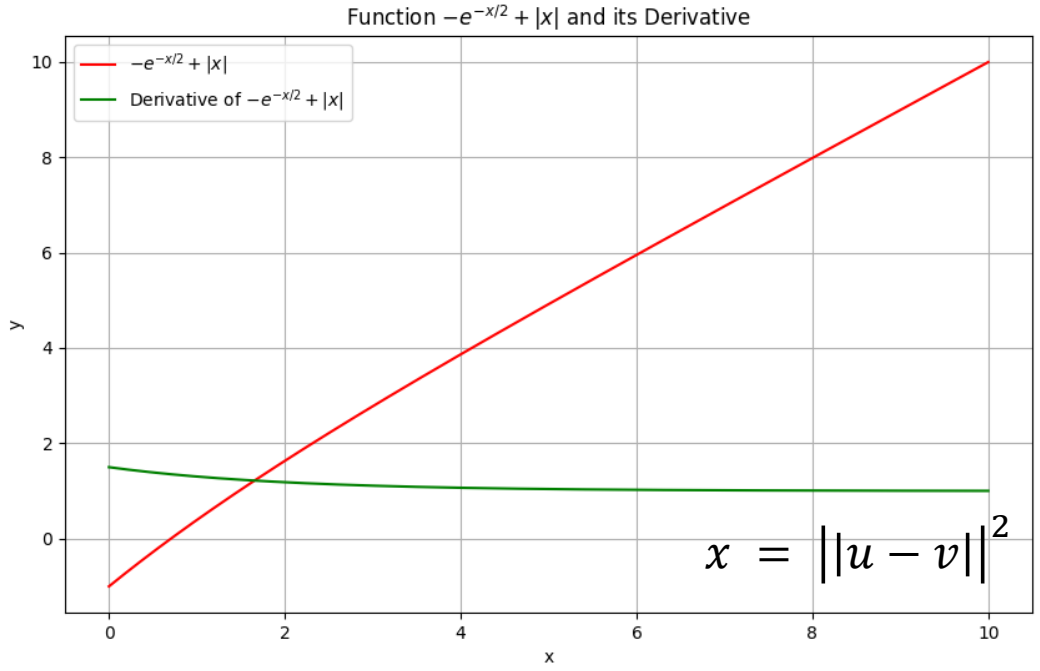
The modified Loss adds a linear distance kernel to improve gradients and helps the training converge reliably.

$$\min_{\boldsymbol{D}} \mathcal{L}_{\boldsymbol{D}_{\text{mod}}} = \mathcal{L}_{\boldsymbol{D}} + \lambda_{mod}\mathbb{E}_{P_{\boldsymbol{Y}}}[k_{\boldsymbol{D}}^{\text{lin}}(\boldsymbol{y},\boldsymbol{y}')],$$



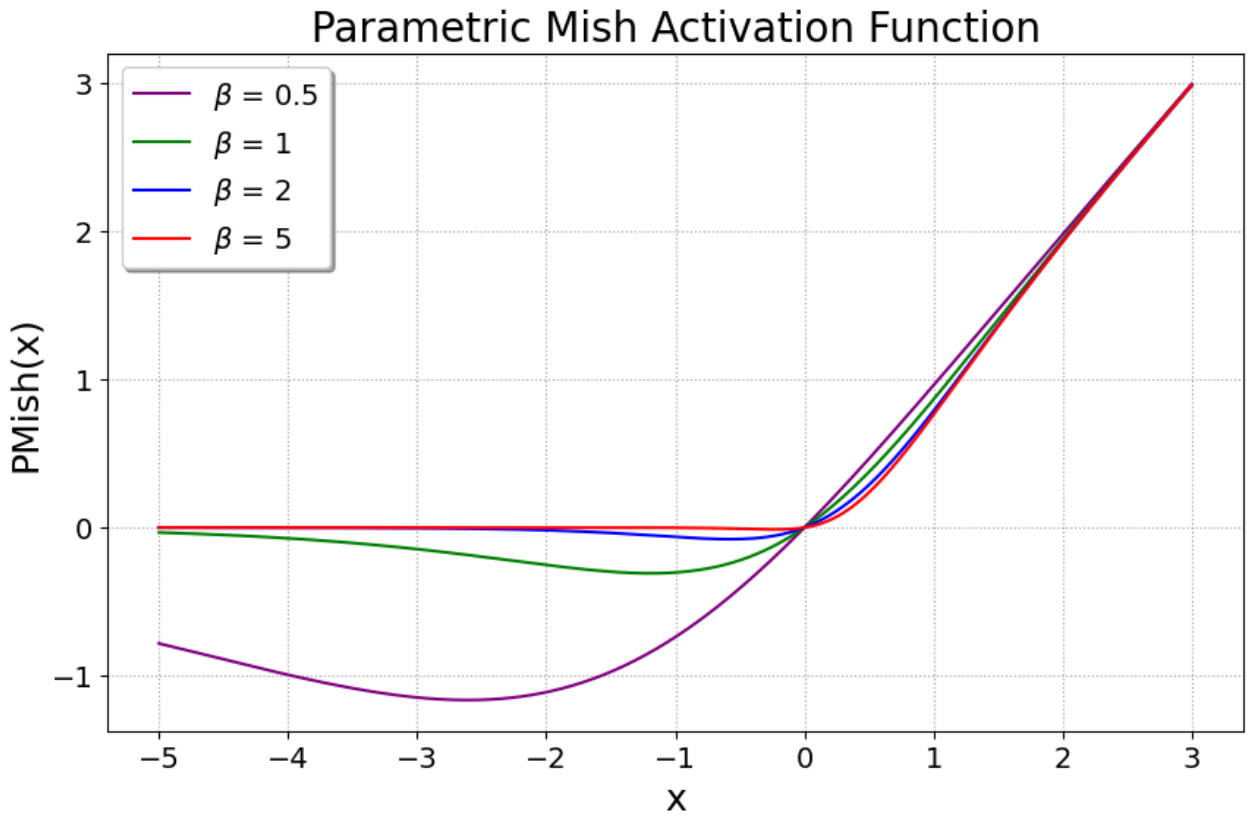


Original



Proposed

3. Activation Function: Parametric Mish

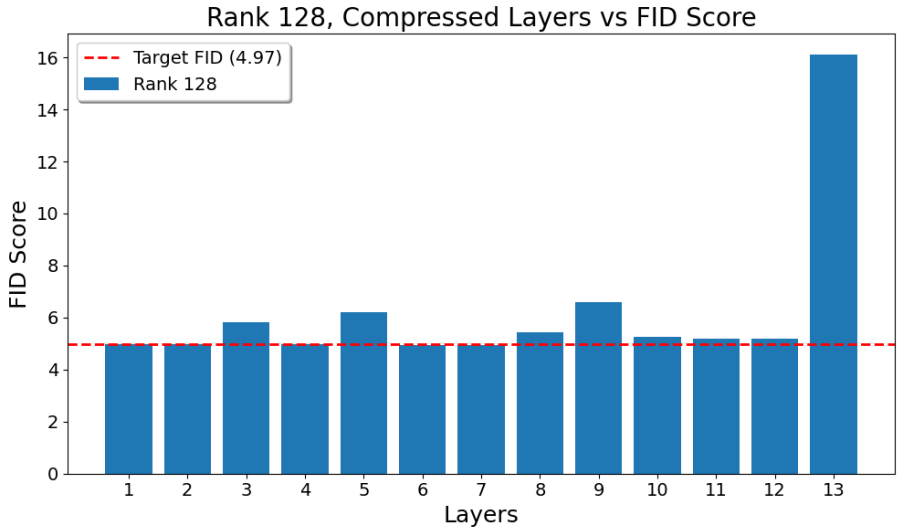


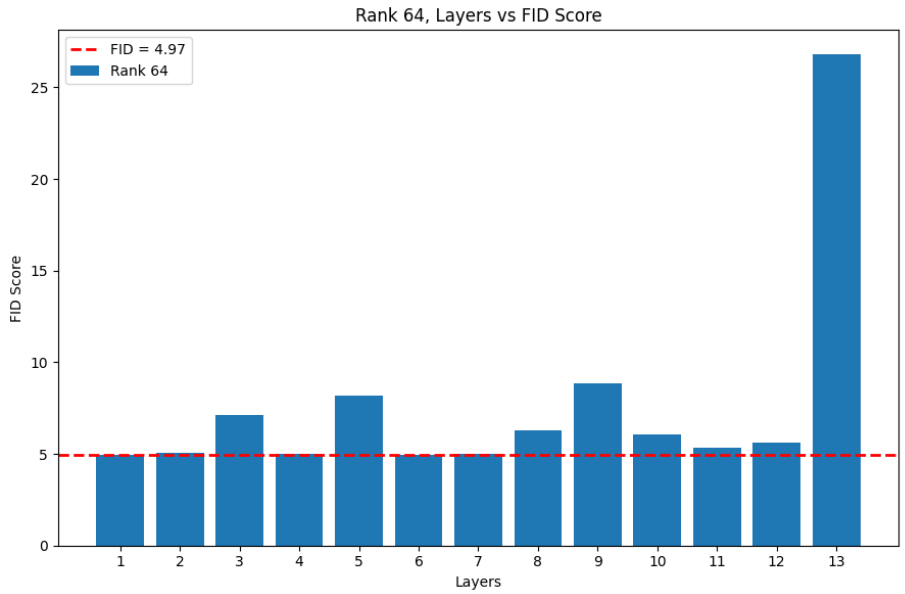
$$f(\boldsymbol{x}) = \boldsymbol{x} \cdot \tanh\left(\frac{\ln(1+e^{\beta \boldsymbol{x}})}{\beta}\right), \beta > 0$$

- **PMish**† has built-in parameters that can adjust its shape as the model trains.
- This enables each activation layer to find an optimal balance between **linearity** and **non-linearity**, which is crucial for capturing complex and nuanced data distributions in image generation.

†**Prasanna Reddy Pulakurthi**, Mahsa Mozaffari, Sohail A. Dianat, Jamison Heard, Raghuveer M. Rao, Majid Rabbani. “Enhancing GANs with MMD Neural Architecture Search, PMish Activation Function, and Adaptive Rank Decomposition.” ***IEEE Access Journal***, Vol. 12, pp. 174222–174244, 2024.

4. Network Compression by Adaptive Rank Decomposition (ARD)





Algorithm 2 Adaptive Rank Decomposition (ARD)

Finding the optimal Ranks with ARD:

Set candidate ranks for convolutional layers:

$R_{\text{conv}} = [128, 256, 512, 768, nc]$;

Set candidate ranks for fully connected layers:

$R_{\text{fc}} = [2, 4, 8, 16, 32, nc]$;

Set Penalty Factor value: example PF $= 1/10$;

foreach *layer* L *in* G **do**

  foreach *candidate rank* R *in* R_{conv} *or* R_{fc} **do**

    i. CPD Compress layer L with rank R;

    ii. Compute FID score $\text{FID}_L(R)$ for the generator network G with layer L compressed using rank R;

    iii. Compute Compression Ratio $\text{CR}_L(R)$ for layer L with rank R;

    iv. Compute penalty factor $\gamma_L(R) = 1 + \text{PF} \cdot (1 - \text{CR}_L(R))$;

    v. Store the score $\gamma_L(R) \times \text{FID}_L(R)$ for rank R;

  vi. For layer L select the optimal rank R_L^* that minimizes $\gamma_L(R) \times \text{FID}_L(R)$;

Output: Optimal ranks R_L^* for each layer L in G.

Fine-Tuning:

1. Compress the generator G using CPD with the optimal ranks R_L^*;

2. Fine-tune both generator G and discriminator D with compressed layers using a lower learning rate;

Result: A compressed generator network with minimal performance degradation.

Adaptive Rank Decomposition

TABLE 2. The optimal rank R_L^* corresponding to each Penalty Factor for all the layers in the generator network, determined on the CIFAR-10 dataset. These results pertain to the large generator network architecture. The last two columns correspond to network size in megabytes and FID score.

Penalty Factor	Layer													Network Size (MB)	FID ↓
	1	2	3	4	5	6	7	8	9	10	11	12	13		
1	128	128	256	128	256	128	128	128	512	128	128	128	768	2.55	13.22
1/5	128	128	768	128	768	128	128	256	512	256	128	128	768	3.21	9.52
1/10	128	128	768	128	768	128	128	256	768	256	768	768	nc	5.25	7.67
1/15	128	128	768	128	768	128	128	512	nc	256	768	768	nc	6.62	6.63
1/20	128	128	768	128	nc	128	128	nc	nc	512	768	768	nc	8.32	5.85
0	nc	nc	nc	nc	nc	nc	nc	nc	nc	nc	nc	nc	nc	20.18	4.97

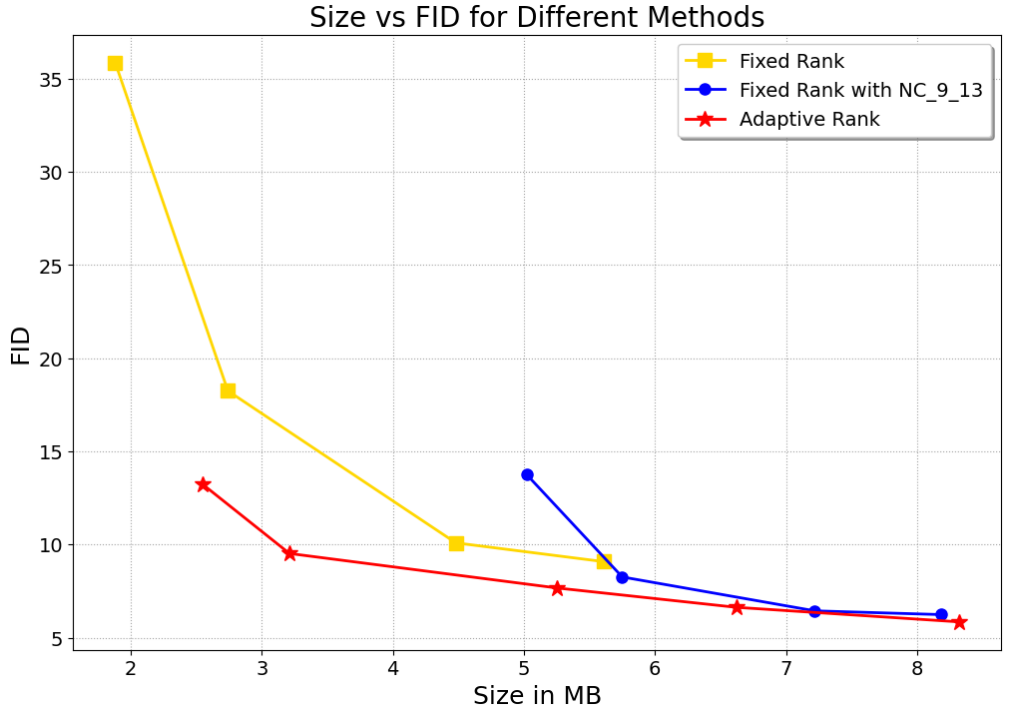


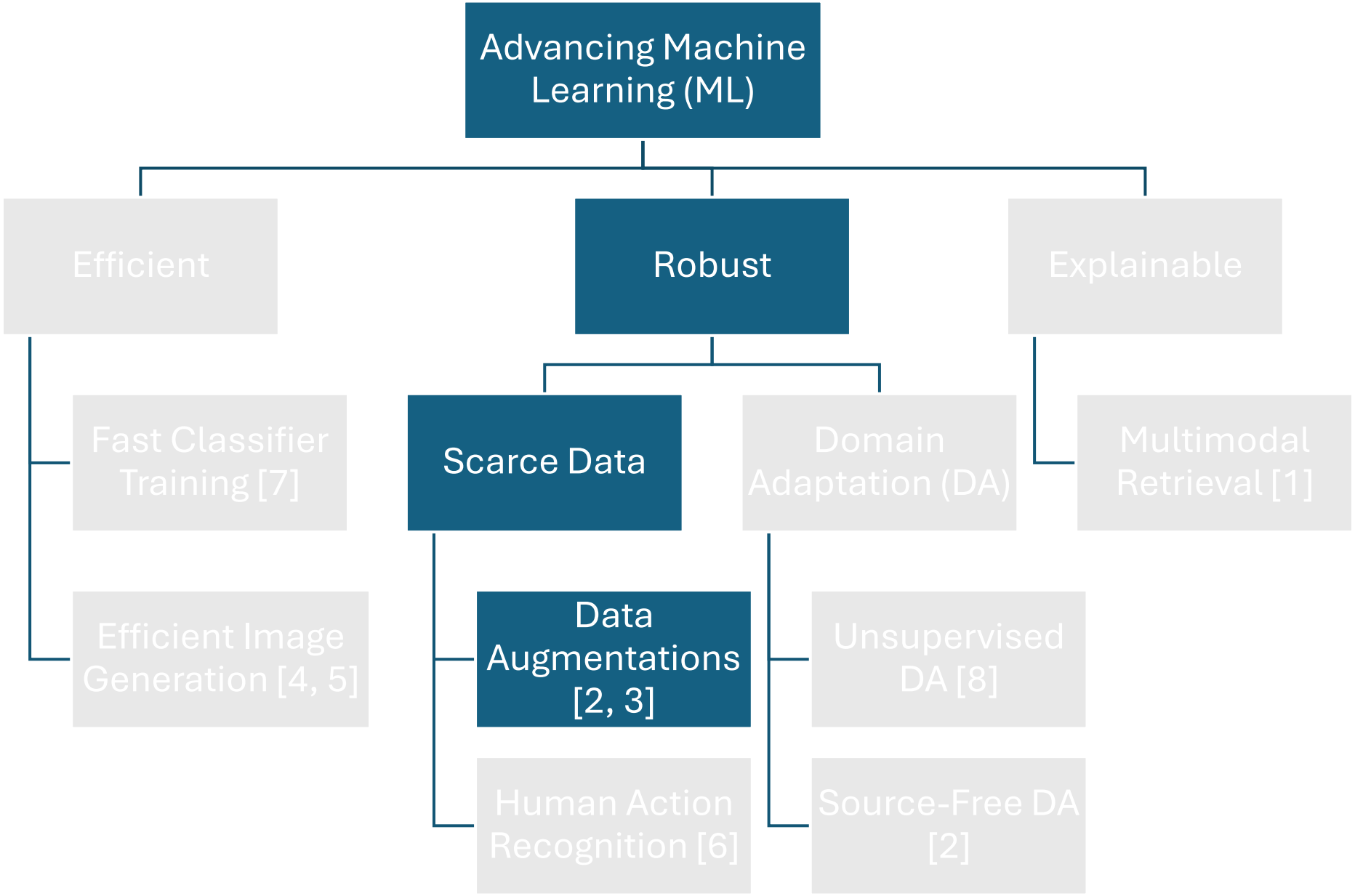
Image Generation on CelebA (64x64)

TABLE 5. The quantitative comparisons of various methods on the CelebA (64 × 64), using FID as the evaluation metric.

	Method	FID ↓
	FCE [84]	12.21
	NCP-VAE [85]	5.25
	TransGAN [74]	5.01
	BigGAN-OBRS [86]	3.74
6.67 MB	MMD-PMish-NAS (Ours)	**1.92**

3.04 MB	MMD-PMish-NAS (Compressed)	**FID ↓ 2.03**

Prasanna Reddy Pulakurthi, Mahsa Mozaffari, Sohail A. Dianat, Jamison Heard, Raghuveer M. Rao, Majid Rabbani. "Enhancing GANs with MMD Neural Architecture Search, PMish Activation Function, and Adaptive Rank Decomposition." ***IEEE Access Journal***, Vol. 12, pp. 174222–174244, 2024.

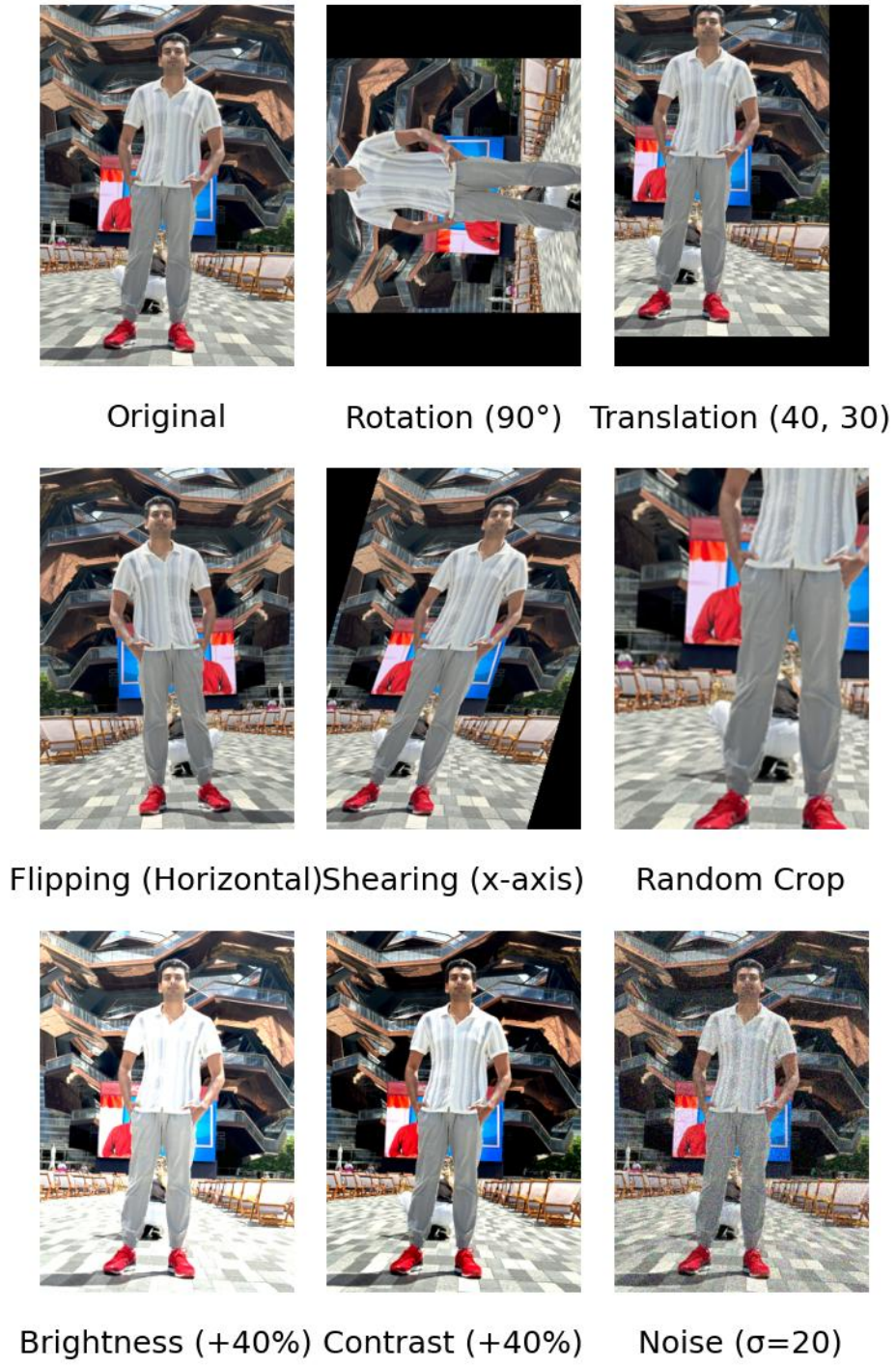


2. Prasanna Reddy Pulakurthi, Majid Rabbani, Jamison Heard, Sohail Dianat, Celso M. de Melo, Raghuveer M. Rao. "Shuffle PatchMix Augmentation with Confidence-Margin Weighted Pseudo-Labels for Enhanced Source-Free Domain Adaptation." *Proceedings of the IEEE International Conference on Image Processing (**ICIP**)*, IEEE, 2025.

3. Prasanna Reddy Pulakurthi, Majid Rabbani, Celso M. de Melo, Sohail A. Dianat, Raghuveer M. Rao. "Effective Dual-Region Augmentation for Reduced Reliance on Large Amounts of Labeled Data." *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications III*, Vol. 13459, pp. 210–218, SPIE, 2025.

Motivation

- Deep learning models have achieved high performance in many CV tasks **but require vast amounts of labeled data**.
- In data-scarce scenarios, **data augmentation** improves performance by transforming the training data while retaining class characteristics.



Contributions

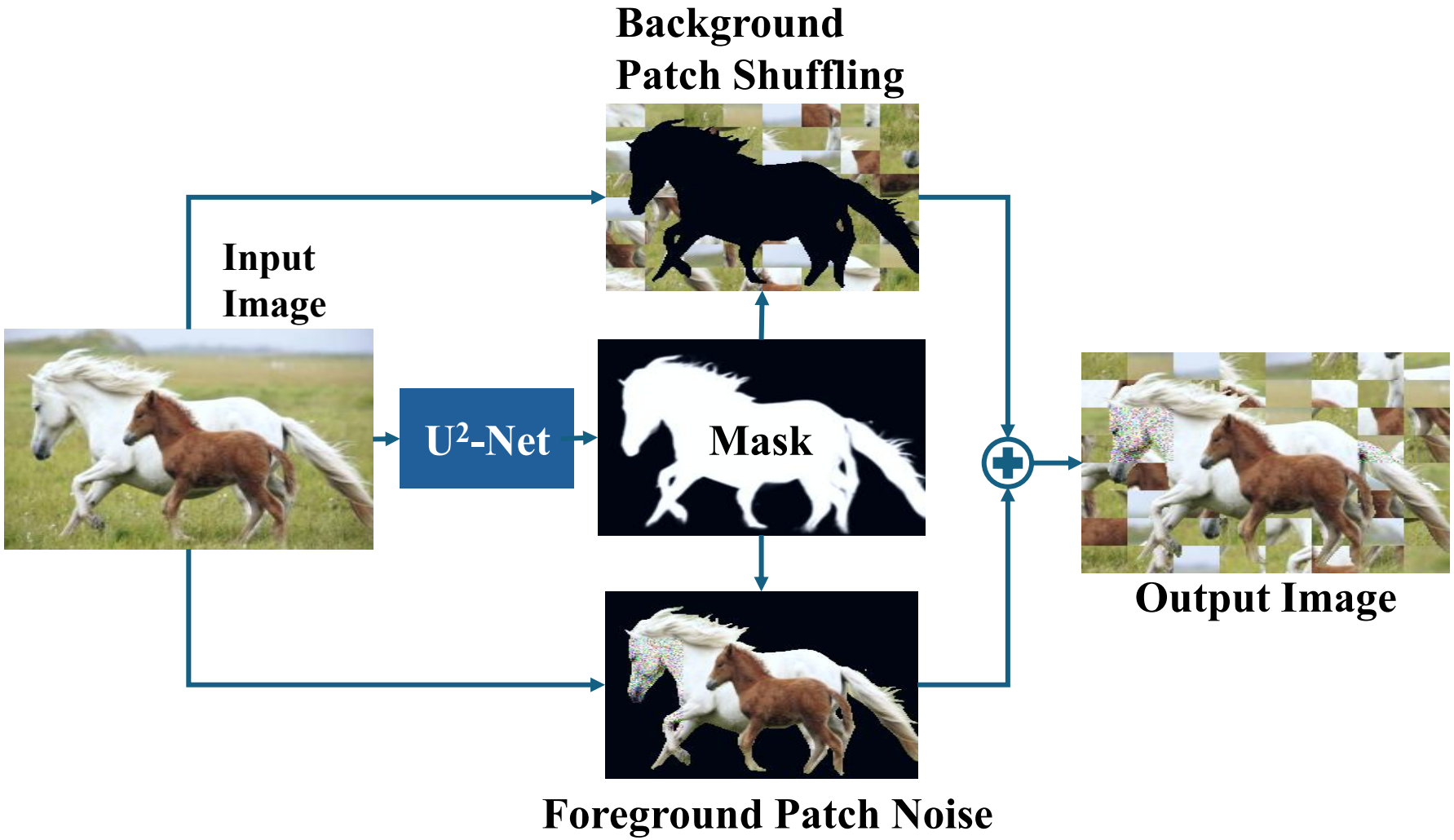
We propose two data augmentation methods.

1. **Dual-Region Augmentation (DRA):** Apply targeted perturbations:
   - Foreground Patch Noise (FPN): Patch-based Gaussian noise.
   - Background Patch Shuffle (BPS): Spatial patch shuffling
2. **Shuffle PatchMix (SPM):**
   - An intra-image patch mixing strategy that creates diverse and challenging transformations by shuffling and blending patches.

1. Dual-Region Augmentation (DRA)

Steps

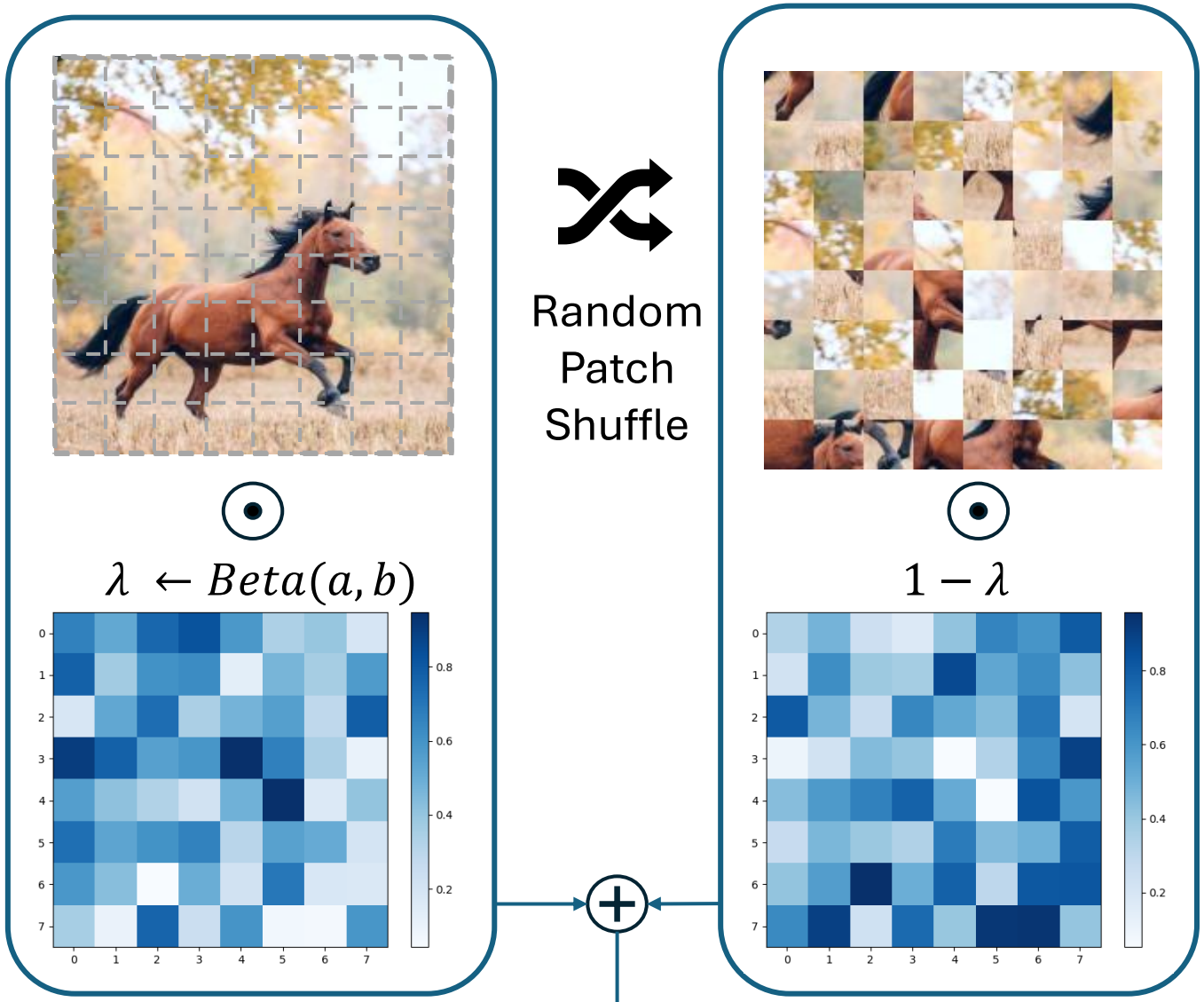
1. Input Image
2. U^2-Net Mask
3. Foreground Patch Noise
4. Background Patch Shuffle
5. Combined Output



Foreground Patch Noise – Encourages the model to utilize a broader set of discriminative features rather than relying on specific regions.

Background Patch Shuffle – Encourages the model to focus on the foreground.

2. Shuffle PatchMix (SPM)



Additional Augmentations:

- Random Resized Crop
- Random Horizontal Flip
- Color Jitter
- Random Grayscale
- Random Gaussian Filter

Adaptive Mixing Strength & Patch Overlap

Original

Beta(8,2)

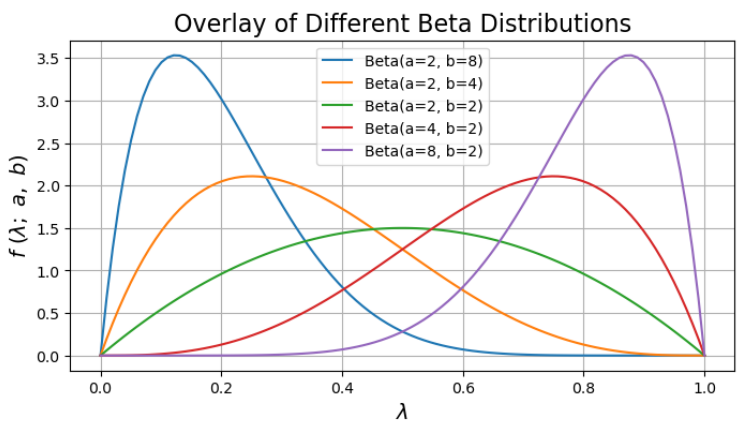
Beta(4,2)

Beta(2,2)

Epoch

0

100



16 Patches

Overlapping Patch Blending

Results on Source-Free Domain Adaptation

P

A

C

S

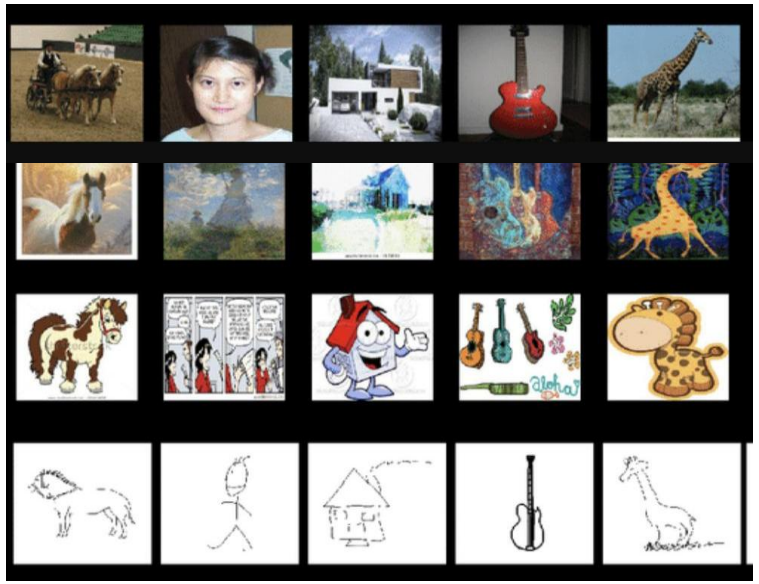
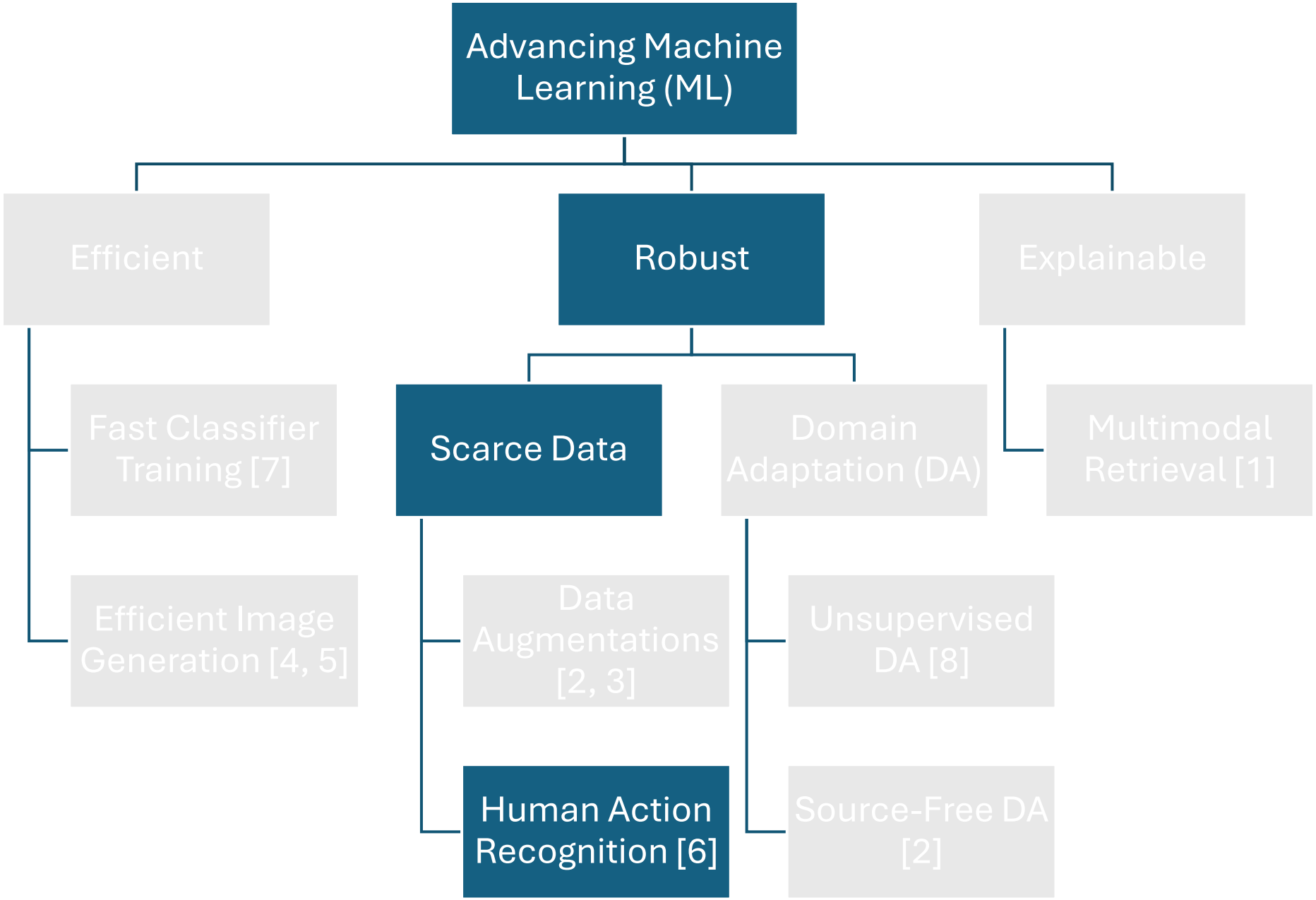


Table 3.1: Classification accuracy (%) on PACS [13] for the single-target setting with ResNet-18. Legend: **P**: Photo, **A**: Art-Painting, **C**: Cartoon, and **S**: Sketch. The highest accuracies are in **bold**. * indicates our reproduced results using [14]. SF stands for Source-Free adaptation.

Method	SF	P→A	P→C	P→S	A→P	A→C	A→S	Average
Source only	-	60.6	22.6	22.4	96.0	49.9	37.3	48.1
NEL [149]	✓	82.6	**80.5**	32.3	98.4	**84.3**	56.1	72.4
AdaContrast [14]*	✓	81.3	72.2	66.7	98.7	79.7	77.9	79.4
DRA [3] (Ours)	✓	83.7	76.4	**77.0**	98.4	82.9	**85.3**	**84.0** +4.6%
SPM [2] (Ours)	✓	**86.6**	77.7	74.1	**99.1**	84.1	84.6	**84.4** +5.0%

- Our augmentations outperform the baseline AdaContrast† by **+4.6%** (DRA) and **+5.0%** (SPM).
- Notably, highest gains are achieved in the challenging **Sketch domain**.

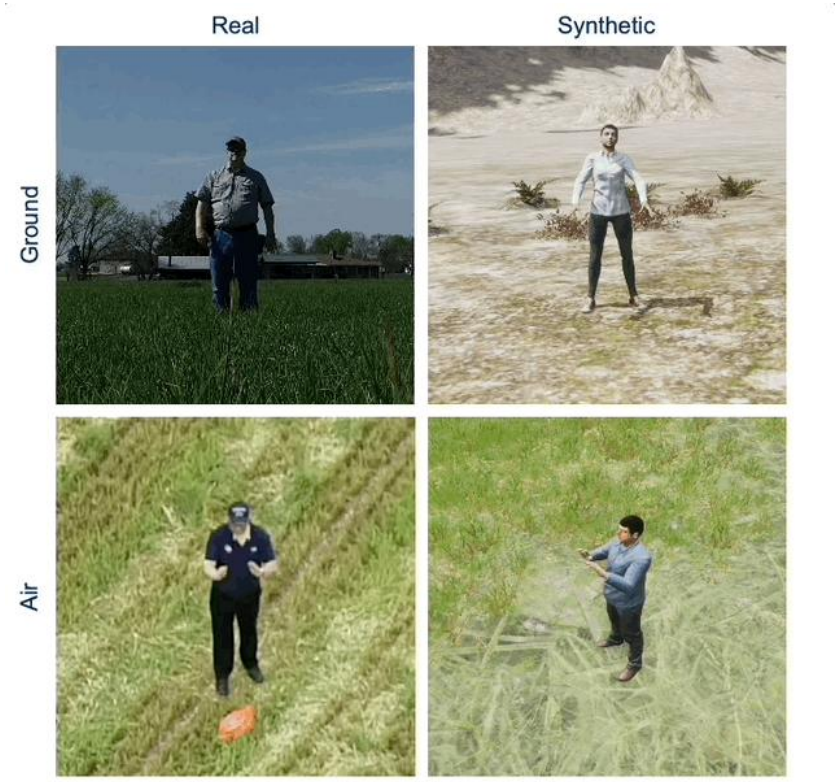
†Chen, Dian, et al. "Contrastive test-time adaptation." Proceedings of the IEEE/CVF CVPR 2022.



6. Prasanna Reddy Pulakurthi, Celso M. de Melo, Raghuveer M. Rao, Majid Rabbani. "Enhancing Human Action Recognition with GAN-Based Data Augmentation." *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*, Vol. 13035, pp. 194–204, SPIE, 2024.

Background: Human Action Recognition (HAR)

- In many real-world scenarios, HAR datasets are limited in size or diversity†.
- To address this, we use GANs to generate synthetic data, expanding the original dataset and improving performance.



†Reddy, Arun V., et al. "Synthetic-to-Real Domain Adaptation for Action Recognition: A Dataset and Baseline Performances." 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023.

Synthetic Video Generation








Liu, Wen, et al. "Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis." Proceedings of the IEEE/CVF ICCV. 2019.

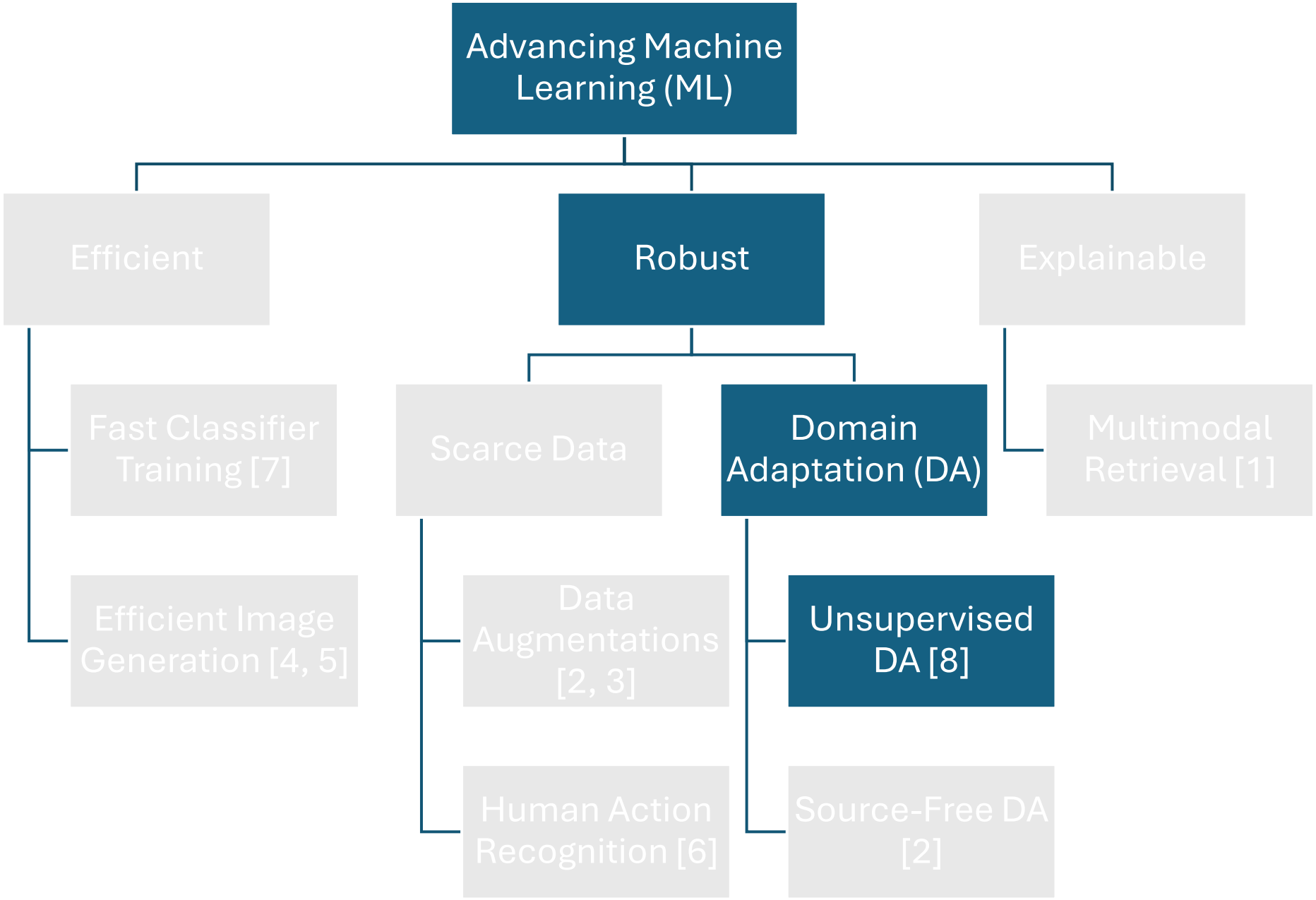
Results

Table 2. The results (top-1 accuracy %) of all four experiments.

Experiment No.	Experiment Name	Ground %	Aerial %
1	Original Data	$84.00_{\pm 2.74}$	$65.93_{\pm 3.56}$
2	One Motion All Appearances (1MAA)	$87.04_{\pm 4.09}$	$68.79_{\pm 5.86}$
3	All Motions One Appearance (AM1A)	$85.64_{\pm 3.96}$	$67.99_{\pm 6.45}$
4	All Motions All Appearances (AMAA)	$\mathbf{88.40}_{\pm \mathbf{0.55}}$	$\mathbf{73.63}_{\pm \mathbf{5.04}}$

- Augmenting the original dataset with synthetic data improves classification performance by **+4.4%** for ground and **+7.7%** on aerial datasets.

Prasanna Reddy Pulakurthi, Celso M. de Melo, Raghuveer M. Rao, Majid Rabbani. “Enhancing Human Action Recognition with GAN-Based Data Augmentation.” *Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*, Vol. 13035, pp. 194–204, SPIE, 2024.



8. Prasanna Reddy Pulakurthi, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "Unsupervised Domain Adaptation Using Feature-Aligned Maximum Classifier Discrepancy." *Applications of Machine Learning 2022*, Vol. 12227, pp. 37–45, SPIE, 2022.

Motivation

Synthetic Data†

Real Data†

Train → Model → High Performance ↑
Accuracy: 99.0%

Test → Trained Model → Low Performance↓
Accuracy: 43.8%

†Peng, Xingchao, et al. "Syn2real: A new benchmark for synthetic-to-real visual domain adaptation." arXiv:1806.09755 (2018).

Domain Adaptation

- **Unsupervised Domain Adaptation (UDA):** Transferring domain knowledge from labeled source data to unlabeled target data.

Source Domain Data

Labeled data†

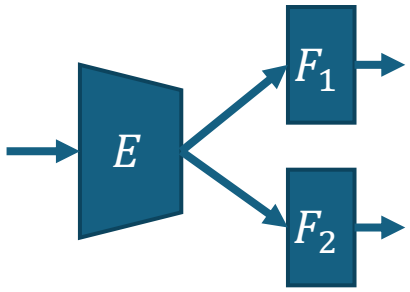
Target Domain Data

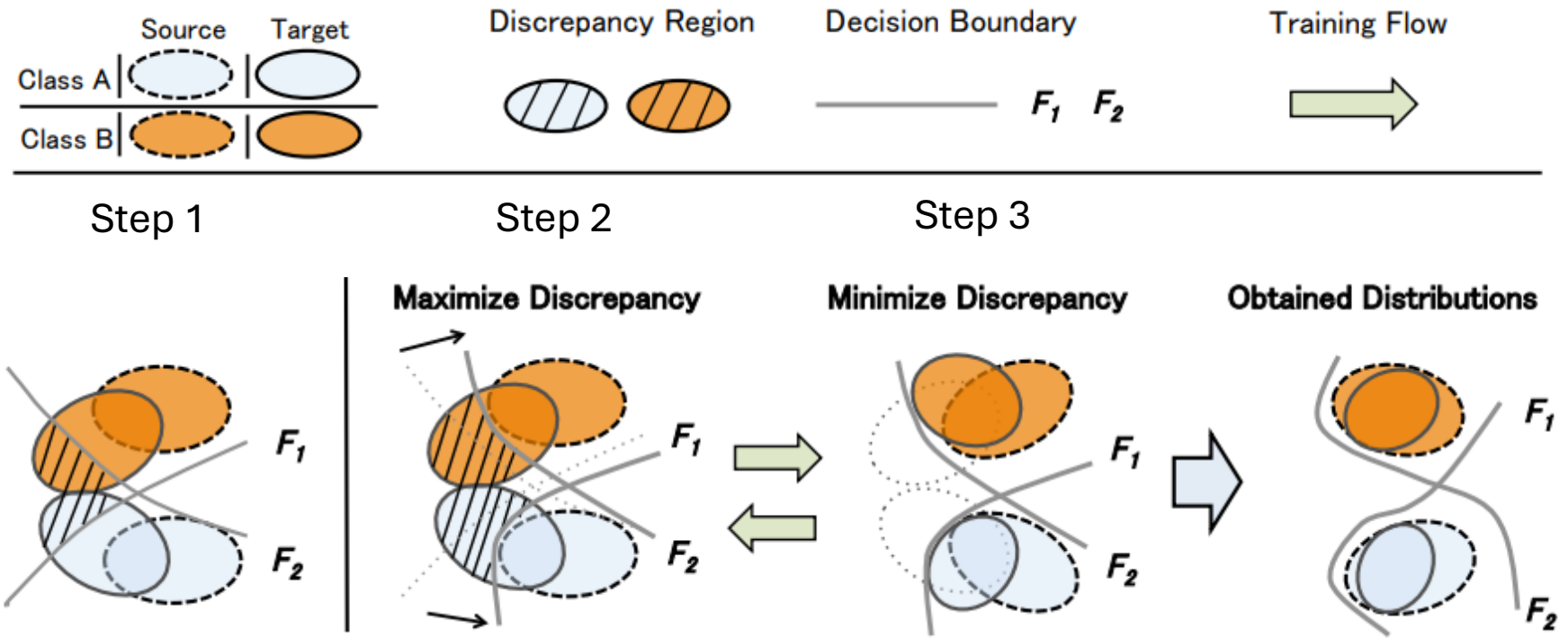
Unlabeled data†

- **Goal**: Learn a classifier f_θ that reliably classifies the target samples using labeled source data and unlabeled target data.

†Peng, Xingchao, et al. "Syn2real: A new benchmark for synthetic-to-real visual domain adaptation." *arXiv preprint arXiv:1806.09755* (2018).

Background: Maximum Classifier Discrepancy (MCD)†



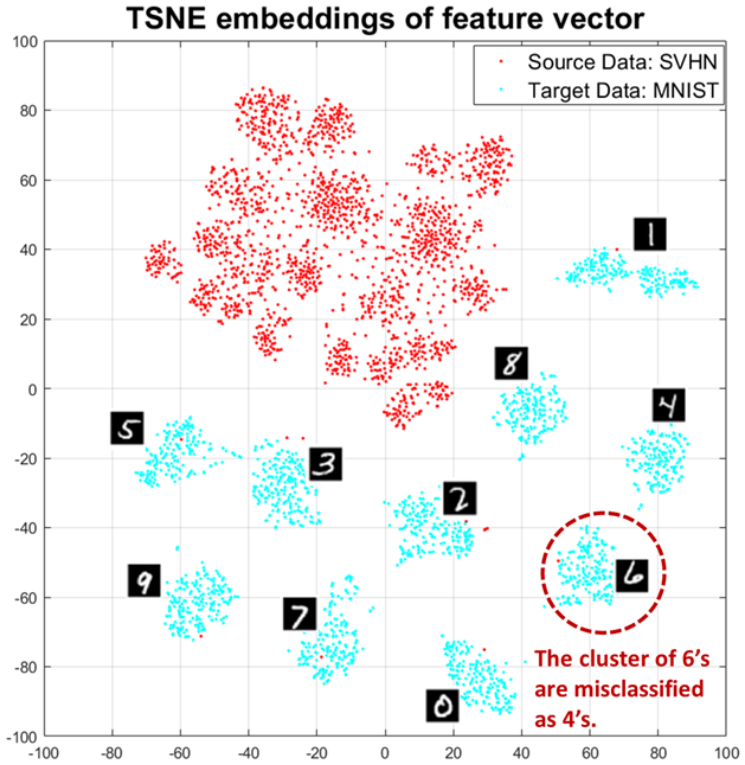


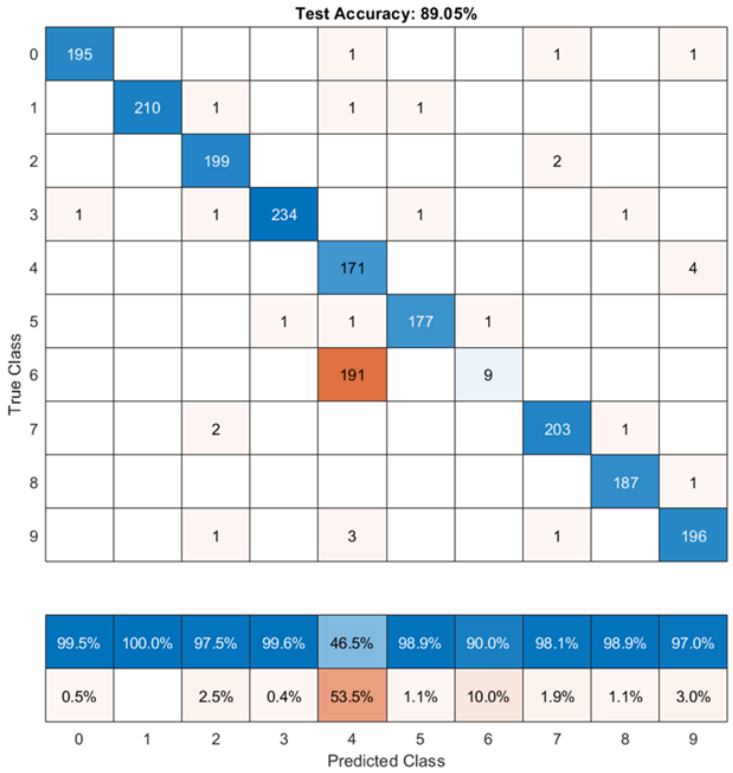
- In this thesis, we propose **two improvements** to the Maximum Classifier Discrepancy (MCD) method.

†Saito, Kuniaki, et al. "Maximum classifier discrepancy for unsupervised domain adaptation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Drawbacks of MCD

- For a large domain shift (SVHN to MNIST), two problems are observed.
  1. The source and target features distributions are not aligned.
  2. Most of the target domain 6's are misclassified as 4's.

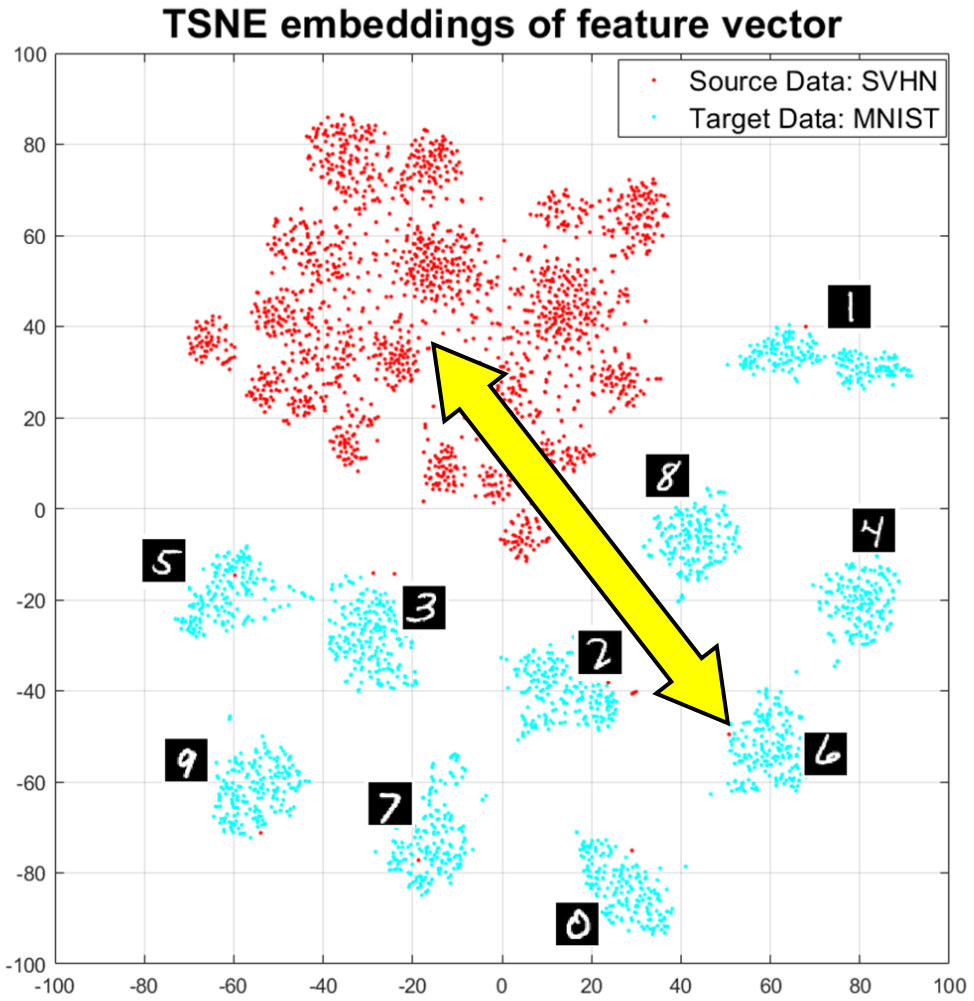




Contributions

- To address these problems, two new training objectives are introduced.
- The first training objective is named the **Feature Alignment Loss** and aims at minimizing the distance between the source and target features and is given by,

$$Loss_{fa} = \|G(X_s) - G(X_t)\|_1$$



Contributions

- The second objective is the **Maximum Entropy Loss** and aims at generating a uniform distribution of target predictions in a minibatch. This is achieved by the loss function:

$$Loss_h = -H\left(E_{X_t}\left(P_{t_1}\right)\right) - H\left(E_{X_t}\left(P_{t_2}\right)\right)$$

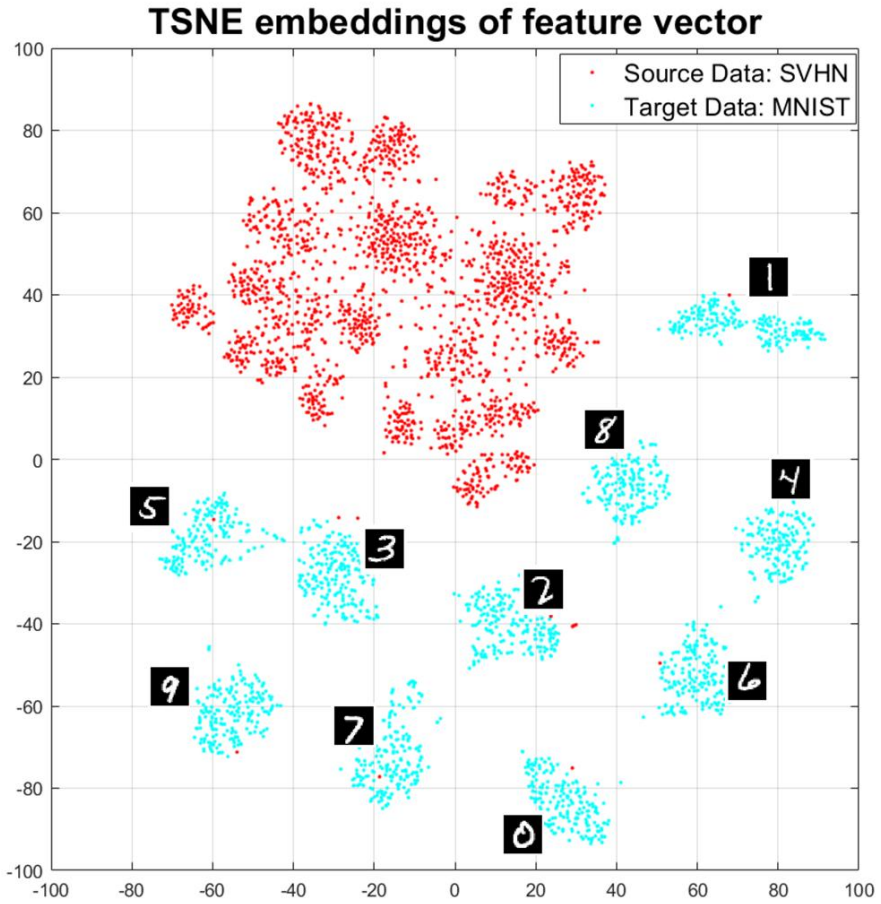
where $H(\,.\,)$ is the entropy and is defined as,

$$H(p) = -\sum_{k=1}^{K} p_k \log_K(p_k)$$

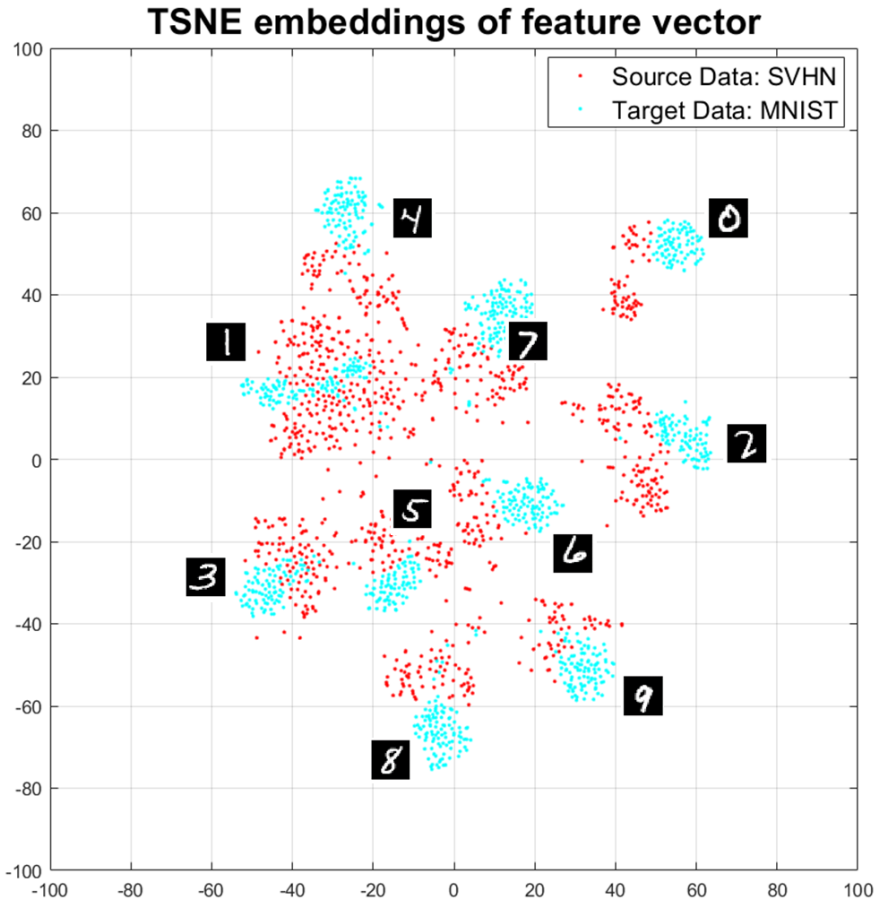
- The two losses are added to the loss function in Step 1 as follows,

$$Loss_1 = \min_{G,F_1,F_2}\left(\mathrm{L}_{CE}(P_{s1}, Y_s) + \mathrm{L}_{CE}(P_{s2}, Y_s) + \lambda_{fa}\, Loss_{fa} + \lambda_h\, Loss_h\right)$$

Proposed Method on SVHN to MNIST Domain Shift



a) Results **without** the two loss functions



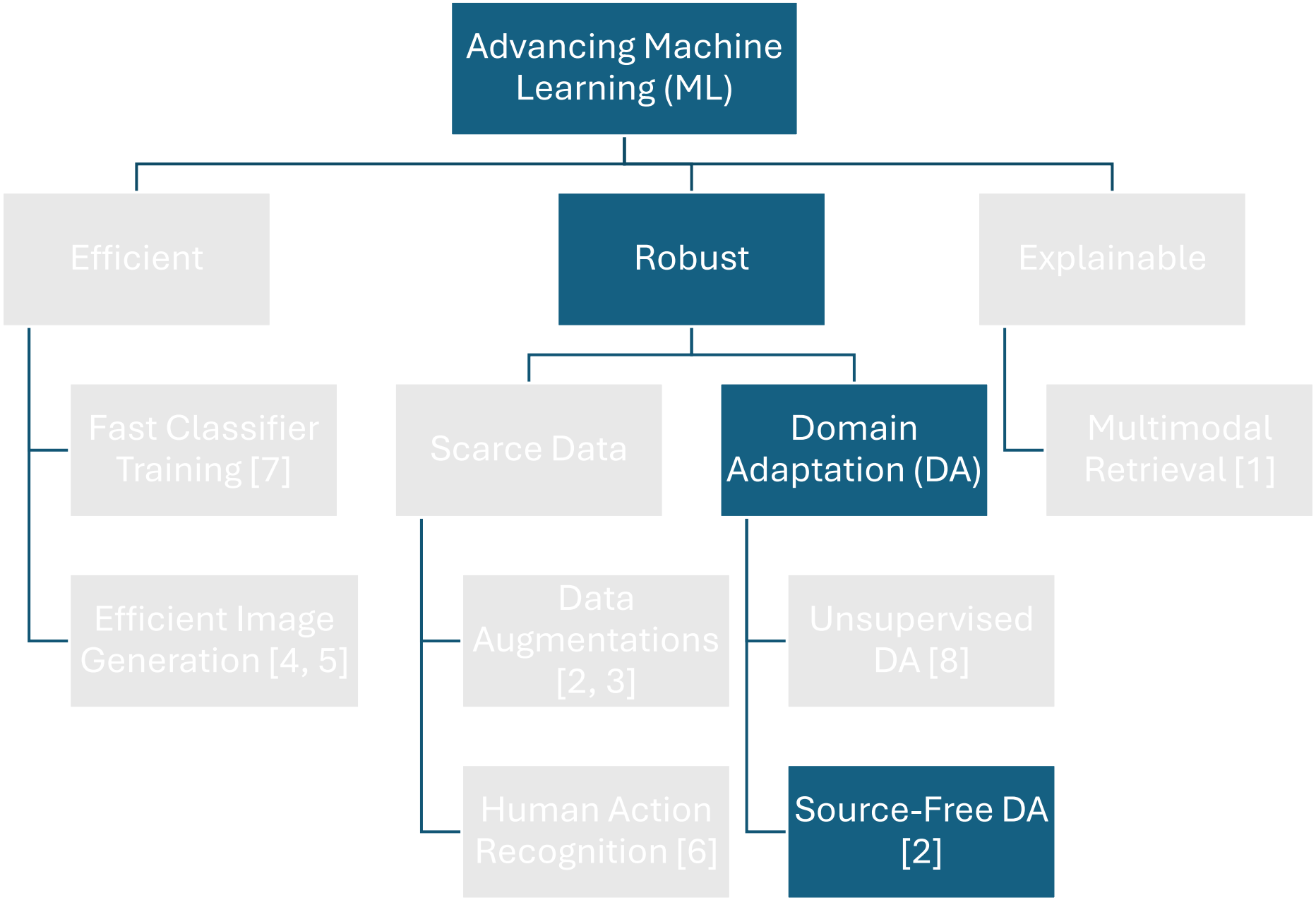
b) Results **with** the two loss functions

Results

Table 1. Experimental results of domain adaptation on digit classification domain shifts. Each experiment is repeated five times, and the mean and standard deviation is reported.

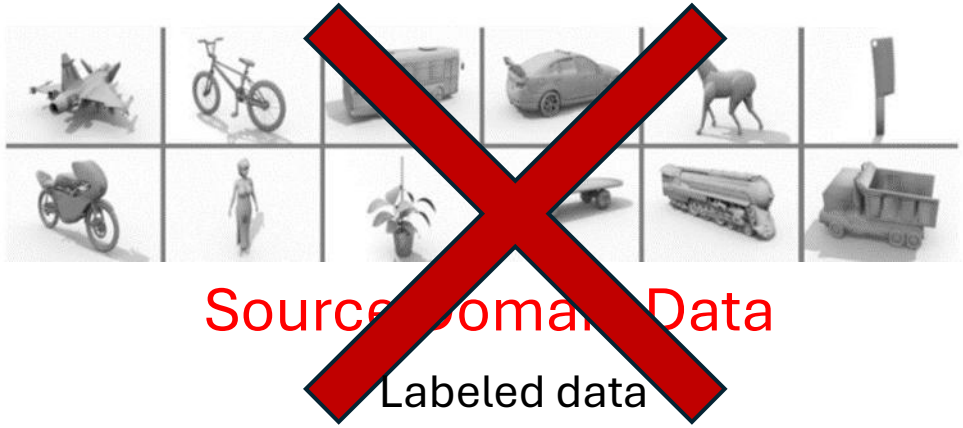
No	Model Name	MNIST → USPS	USPS → MNIST	SVHN → MNIST
1	DANN[15]	85.1	73.0 ± 0.2	71.1
2	ADDA[16]	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8
3	CoGAN[17]	91.2 ± 0.8	89.1 ± 0.8	-
4	CyCADA[19]	95.6 ± 0.2	96.5 ± 0.1	90.4± 0.4
5	MCD[20]	96.5 ± 0.3	-	96.2 ± 0.4
6	MMCD[21]	98.5 ± 0.2	97.0 ± 0.1	98.2 ± 0.1
7	**Proposed Method**	**98.72 ± 0.33**	**98.75 ± 0.12**	**98.76 ± 0.10**

Prasanna Reddy Pulakurthi, Sohail A. Dianat, Majid Rabbani, Suya You, Raghuveer M. Rao. "Unsupervised Domain Adaptation Using Feature-Aligned Maximum Classifier Discrepancy." *Applications of Machine Learning 2022*, Vol. 12227, pp. 37–45, SPIE, 2022.



2. Prasanna Reddy Pulakurthi, Majid Rabbani, Jamison Heard, Sohail Dianat, Celso M. de Melo, Raghuveer M. Rao. "Shuffle PatchMix Augmentation with Confidence-Margin Weighted Pseudo-Labels for Enhanced Source-Free Domain Adaptation." *Proceedings of the IEEE International Conference on Image Processing (**ICIP**)*, IEEE, 2025.

Source-Free Domain Adaptation (SFDA)

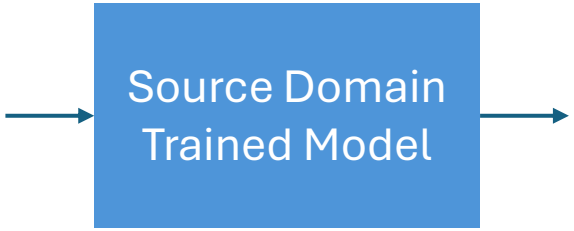


Source Domain Data

Labeled data

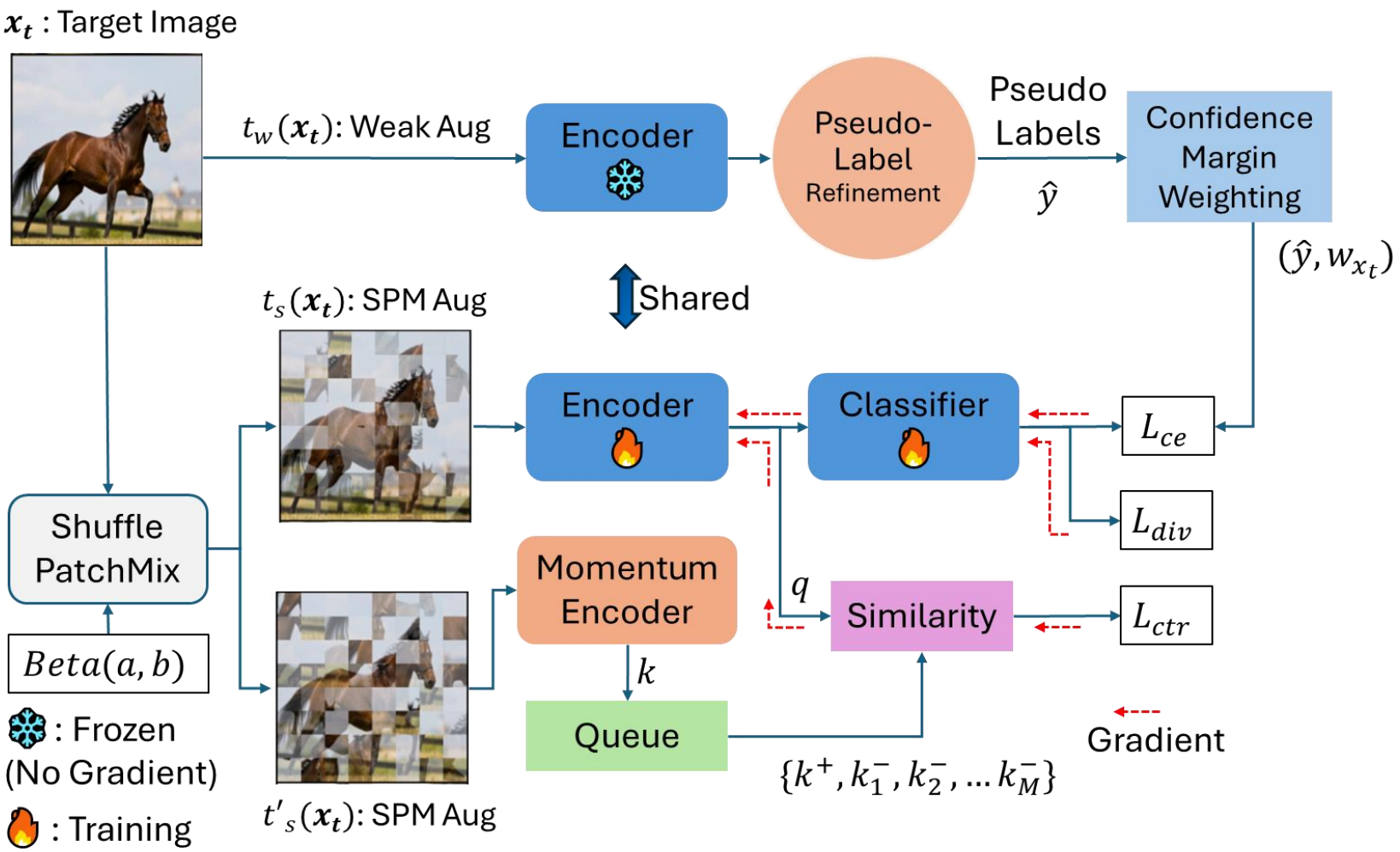
Target Domain Data

Unlabeled data



- However, source data is not always available due to data privacy, limited transmission bandwidth, and security concerns.
- SFDA adapts a **source-trained model** to **unlabeled** target data.

Method Overview



Steps

1. Pseudo-labels & refinement
2. **Confidence-margin weights**
3. **SPM augmentations**
4. **Weighted classification loss L_{ce}**
5. Diversity loss L_{div} & contrastive loss L_{ctr}
6. Train the model to minimize the combined losses

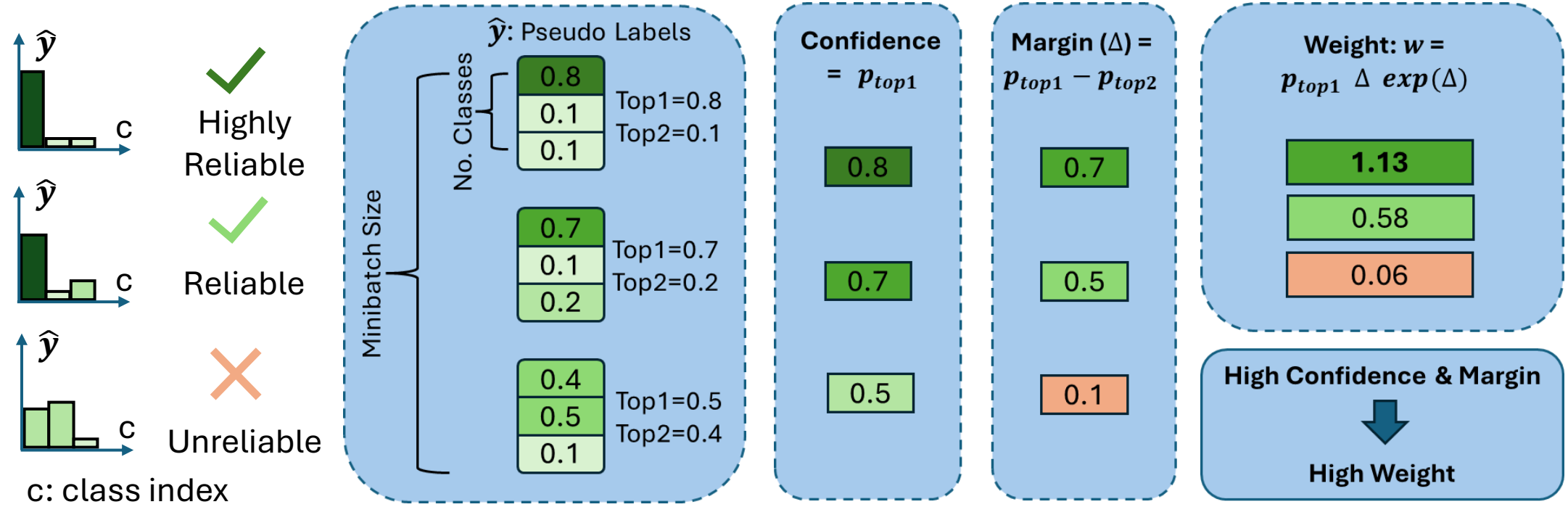
$$L_{ce} + L_{div} + L_{ctr}$$

Contributions

1. **Confidence-Margin Reweighting**: Because pseudo-labels can be noisy, we compute a reliability weight for each pseudo-label and use it for reweighting the classification loss.

2. **Shuffle PatchMix (SPM):** We introduce a novel augmentation technique that enriches target-domain data with diverse and challenging transformations.

1. Confidence-Margin Reweighting



- The equation to compute the weights is given by:

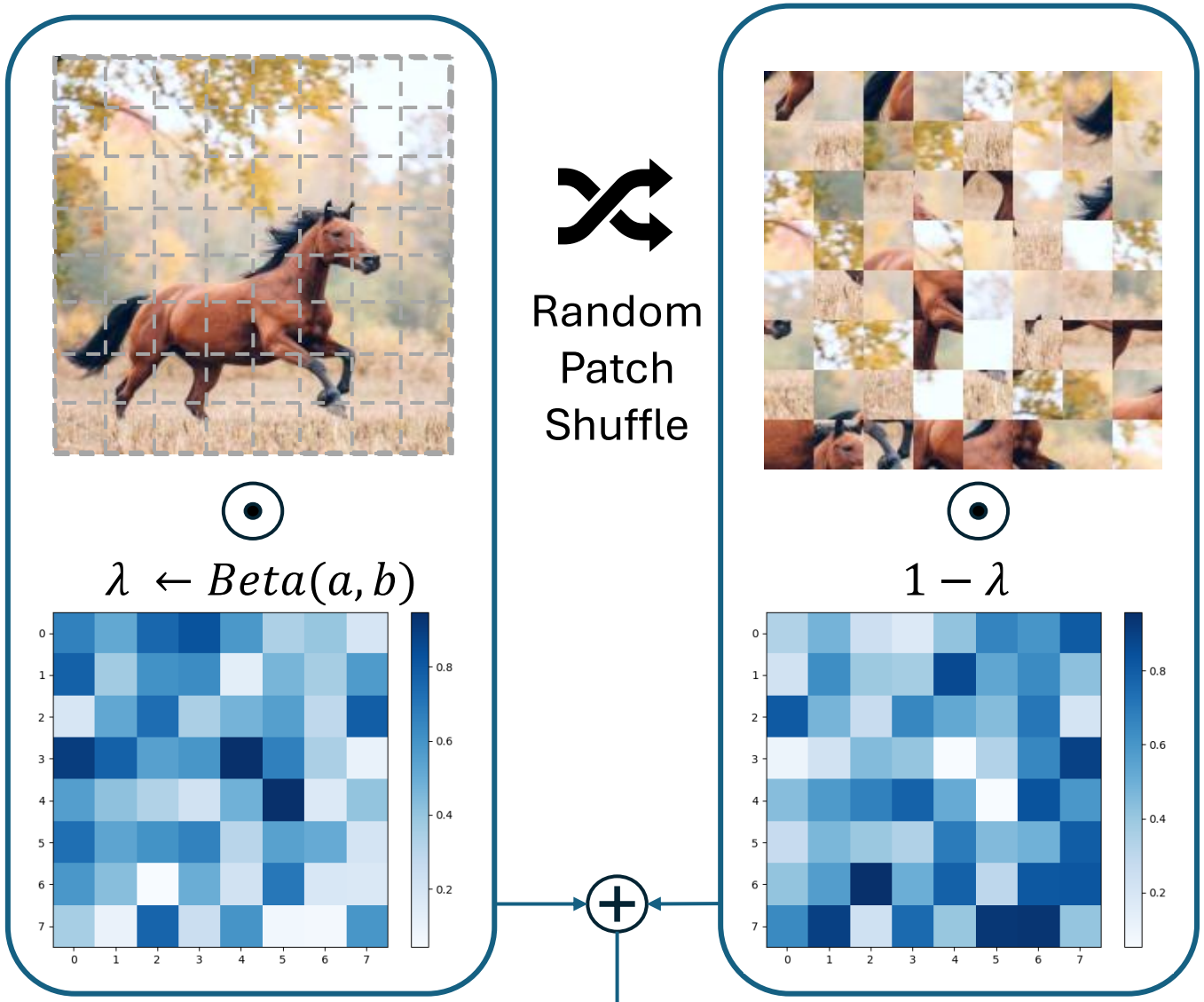
$$w_{x_t} = p_{top1}\,\Delta\,\exp(\Delta),\ \text{where Margin}\ \Delta = p_{top1} - p_{top2}$$

- These weights are incorporated into the classification loss as:

$$L_{ce} = -E_{x_t \in X_t}[w_{x_t} \sum_{c=1}^{C} \hat{y}_t^c \log(p_q^c)]$$

Weights Cross-entropy loss

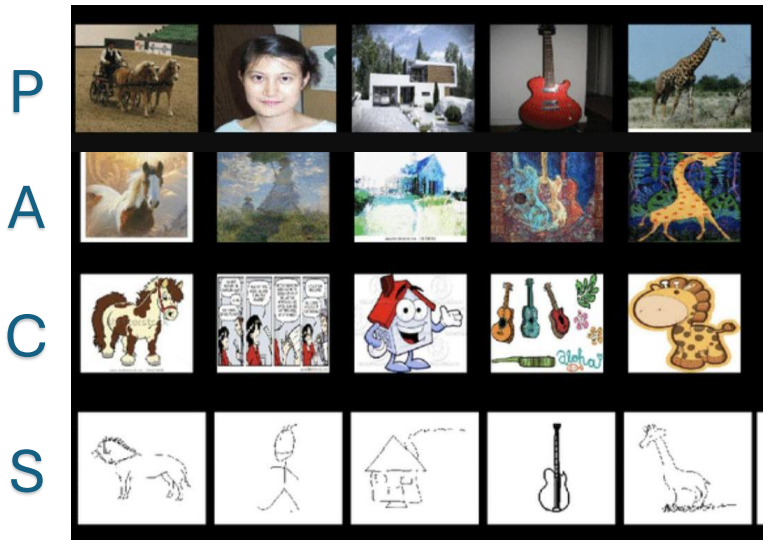
2. Shuffle PatchMix (SPM)

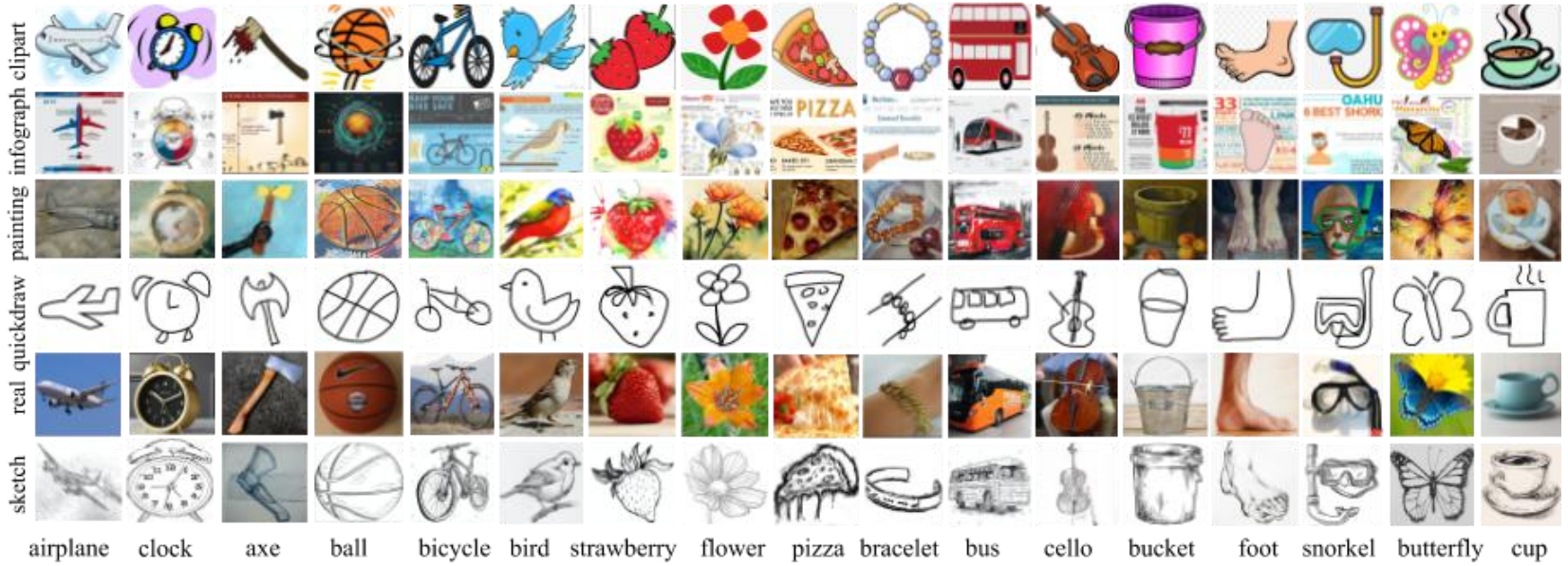


Additional Augmentations:

- Random Resized Crop
- Random Horizontal Flip
- Color Jitter
- Random Grayscale
- Random Gaussian Filter

Benchmarks: PACS & DomainNet





PACS†: 7 categories,
4 domains, ≈9.9k images

DomainNet‡: 345 categories,
6 domains, ≈586k images

†Li, Da, et al. "Deeper, broader and artier domain generalization." ICCV 2017.
‡Peng, Xingchao, et al. "Moment matching for multi-source domain adaptation." ICCV 2019.

Results: PACS & DomainNet-126

PACS single-target (+7.3%)

Method	P→A	P→C	P→S	A→P	A→C	A→S	Avg.
NEL [19]	82.6	80.5	32.3	98.4	84.3	56.1	72.4
AdaContrast [6]*	81.3	72.2	66.7	98.7	79.7	77.9	79.4
SPM (Ours)	**89.7**	**82.3**	**74.5**	**99.1**	**87.9**	**86.4**	**86.7**

PACS multi-target (+7.2%)

Multi-Target UDA		P → A, C, S			A → P, C, S			
Method	SF	A	C	S	P	C	S	Avg.
1-NN	×	15.2	18.1	25.6	22.7	19.7	22.7	20.7
ADDA [2]	×	24.3	20.1	22.4	32.5	17.6	18.9	22.6
DSN [36]	×	28.4	21.1	25.6	29.5	25.8	26.8	25.8
ITA [37]	×	31.4	23.0	28.2	35.7	27.0	28.9	29.0
KD [38]	×	24.6	32.2	33.8	35.6	46.6	57.5	46.6
NEL [19]	✓	80.1	76.1	25.9	96.0	**82.8**	49.8	68.4
AdaContrast [6]*	✓	70.1	77.9	62.9	95.9	72.7	72.9	75.4
SPM (Ours)	✓	**85.2**	**89.2**	**66.4**	**97.7**	76.4	**81.0**	**82.6**

DomainNet-126 (+2.8%)

Method	SF	R→C	R→P	P→C	C→S	S→P	R→S	P→R	Avg.
MCC [8]	×	44.8	65.7	41.9	34.9	47.3	35.3	72.4	48.9
Source only	-	55.5	62.7	53.0	46.9	50.1	46.3	75.0	55.6
TENT [12]	✓	58.5	65.7	57.9	48.5	52.4	54.0	67.0	57.7
SHOT [15]	✓	67.7	68.4	66.9	60.1	66.1	59.9	80.8	67.1
AdaContrast [6]	✓	70.2	69.8	68.6	58.0	65.9	61.5	80.5	67.8
UPA [20]	✓	68.6	69.5	67.6	60.9	66.8	61.5	80.9	68.0
SF(DA)^2 [31]	✓	67.7	59.6	67.8	**83.5**	60.2	**68.8**	70.5	68.3
SPM (Ours)	✓	**74.2**	**71.9**	**72.5**	62.4	**68.1**	66.4	**81.8**	**71.1**

- PACS single-target (+7.3%) over baseline AdaContrast† .
- PACS multi-target (+7.2%) over baseline.
- DomainNet-126 (+3.3%) over baseline.

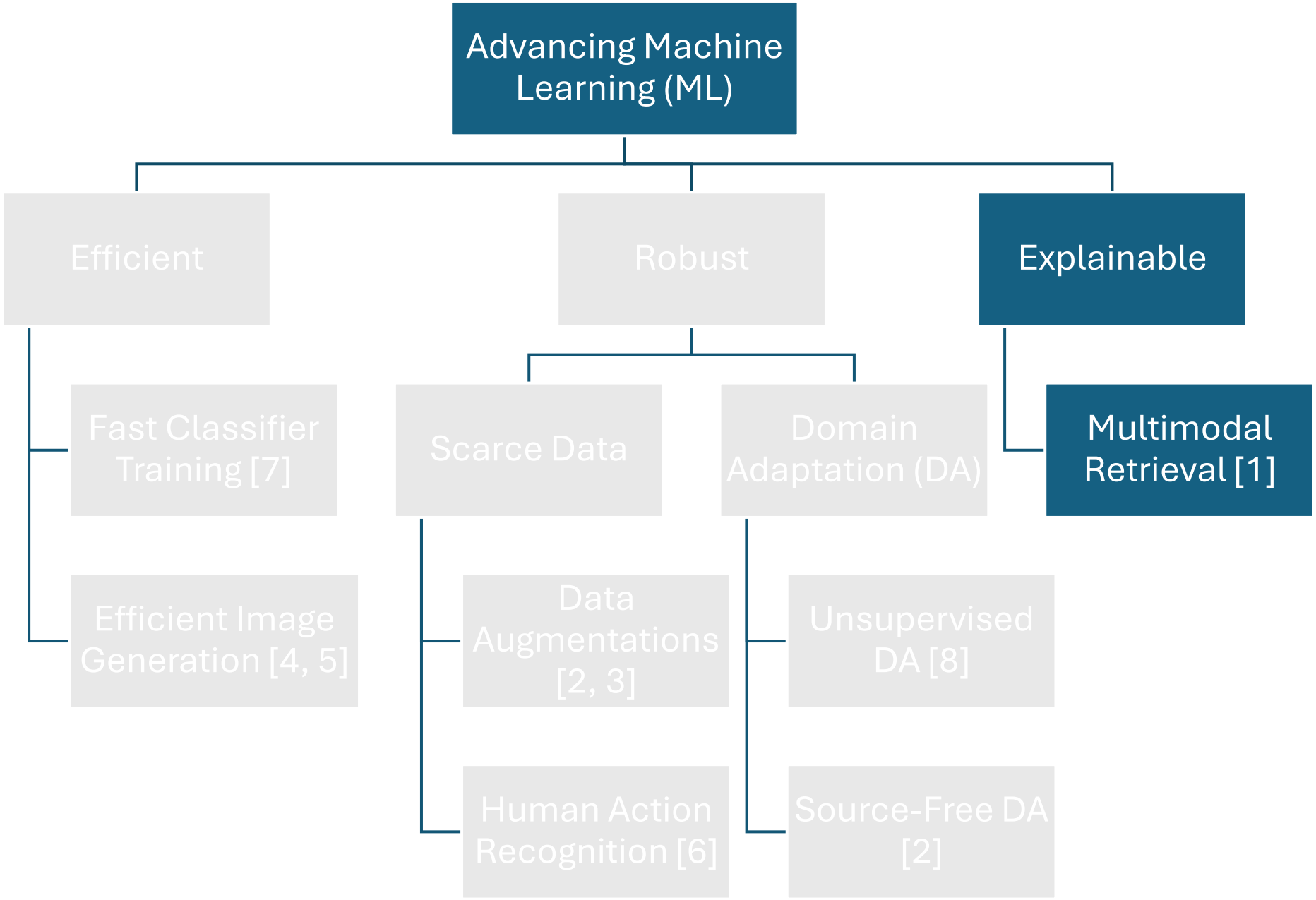
†Chen, Dian, et al. "Contrastive test-time adaptation." Proceedings of the IEEE/CVF CVPR 2022.

Results: VisDA-C

Method	SF	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
DANN [3]	×	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN [4]	×	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
SWD [39]	×	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
MCC [8]	×	88.7	80.3	80.5	71.5	90.1	93.2	85.0	71.6	89.4	73.8	85.0	36.9	78.8
CAN [40]	×	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
FixBi [26]	×	96.1	87.8	90.5	90.3	96.8	95.3	92.8	88.7	97.2	94.2	90.9	25.7	87.2
Source only	-	57.2	11.1	42.4	66.9	55.0	4.4	81.1	27.3	57.9	29.4	86.7	5.8	43.8
MA [41]	✓	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
BAIT [13]	✓	93.7	83.2	84.5	65.0	92.9	95.4	88.1	80.8	90.0	89.0	84.0	45.3	82.7
SHOT [15]	✓	95.3	87.5	78.7	55.6	94.1	94.2	81.4	80.0	91.8	90.7	86.5	59.8	83.0
AdaContrast [6]	✓	97.0	84.7	84.0	77.3	96.7	93.8	91.9	84.8	94.3	93.1	94.1	49.7	86.8
SF(DA)^2 [31]	✓	96.8	89.3	82.9	81.4	96.8	95.7	90.4	81.3	95.5	93.7	88.5	64.7	88.1
Improved SFDA [30]	✓	97.5	**91.4**	**87.9**	79.4	97.2	**97.2**	92.2	83.0	96.4	94.2	91.1	53.0	88.4
UPA [20]	✓	97.0	90.4	82.6	65.0	96.7	96.7	91.0	**87.0**	**96.8**	**96.5**	89.2	**75.0**	88.7
SPM (Ours)	✓	**98.1**	87.9	86.7	**86.2**	**97.7**	94.8	**93.3**	85.6	95.9	95.6	**95.5**	55.3	**89.4**

- VisDA-C avg: 89.4% (+0.7%). Improvement of +2.6% over baseline.
- Best or second-best in 8 of 12 categories.

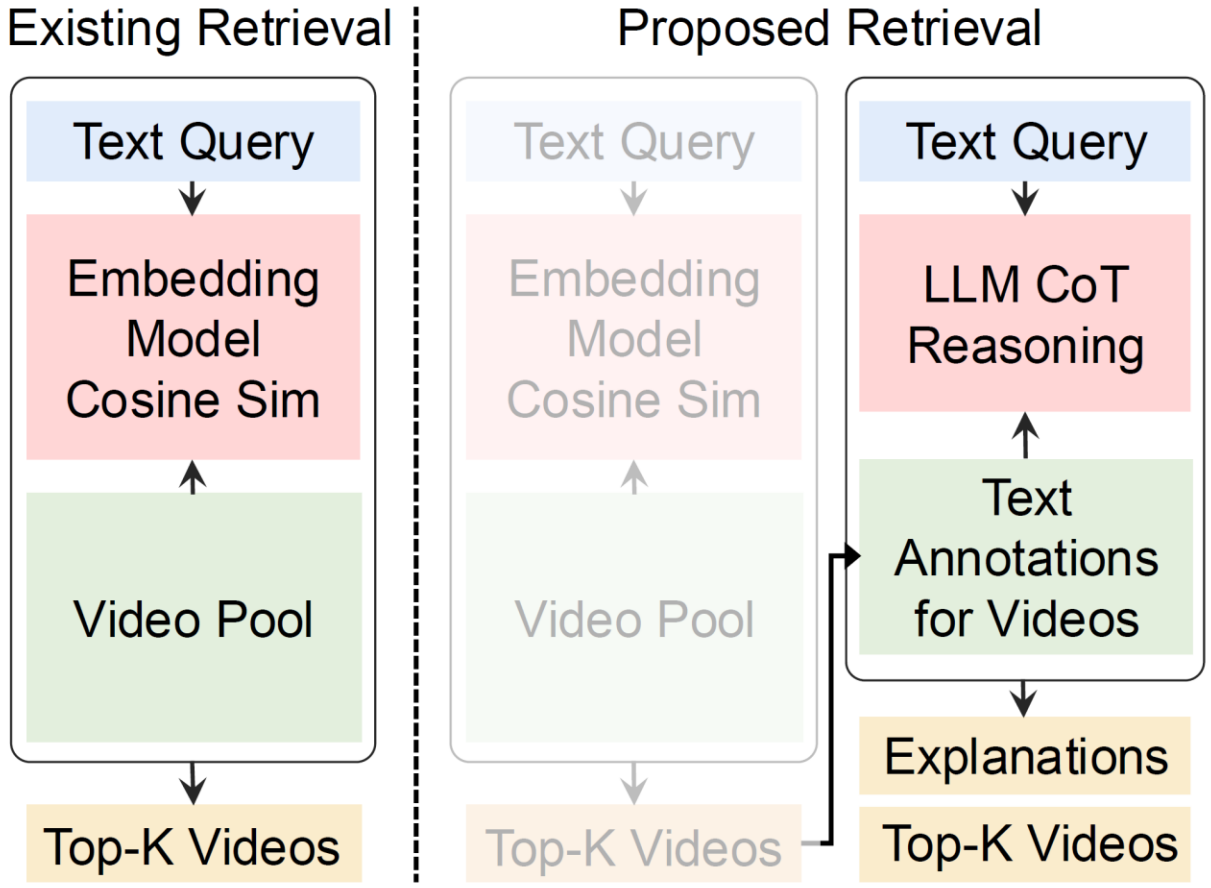
Prasanna Reddy Pulakurthi, Majid Rabbani, Jamison Heard, Sohail Dianat, Celso M. de Melo, Raghuveer M. Rao. "Shuffle PatchMix Augmentation with Confidence-Margin Weighted Pseudo-Labels for Enhanced Source-Free Domain Adaptation." *Proceedings of the IEEE International Conference on Image Processing (**ICIP**)*, IEEE, 2025.



1. **Prasanna Reddy Pulakurthi**, Jiamian Wang, Majid Rabbani, Sohail Dianat, Raghuveer Rao, Zhiqiang Tao. "X-CoT: Explainable Text-to-Video Retrieval via Large Language Models (LLM) Based Chain-of-Thought Reasoning." *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (**EMNLP**)*, 2025.

Motivation & Background

- Text-to-video retrieval systems rely on embedding models using cosine similarity.
- These methods lack interpretability and are sensitive to low-quality text-video pairs.
- **Goal:** Develop a transparent retrieval system that explains *why* a video is retrieved.

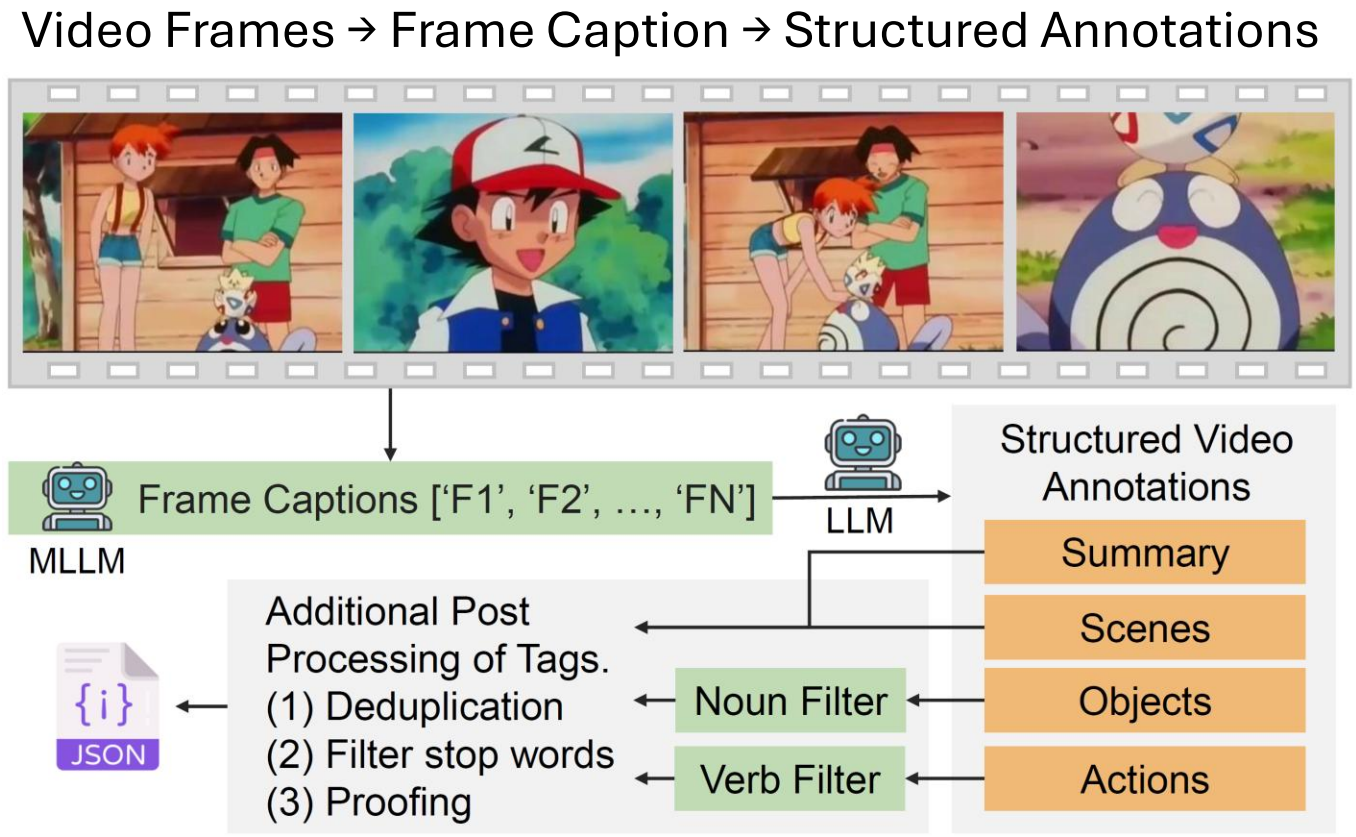


Contributions

- **Introduce** X-CoT, an explainable retrieval framework using LLM Chain-of-Thought reasoning, advancing transparent and trustworthy retrieval.
- **Expand** benchmarks with **structured video annotations** (objects, actions, scenes, ...) for richer semantics.
- **Employ** pairwise LLM comparisons with Bradley-Terry aggregation and produce both **rankings and explanations**.
- **Achieve** consistent performance gains across four major benchmarks (e.g., **+5.6 % R@1** on MSVD) while enabling model and data quality analysis.

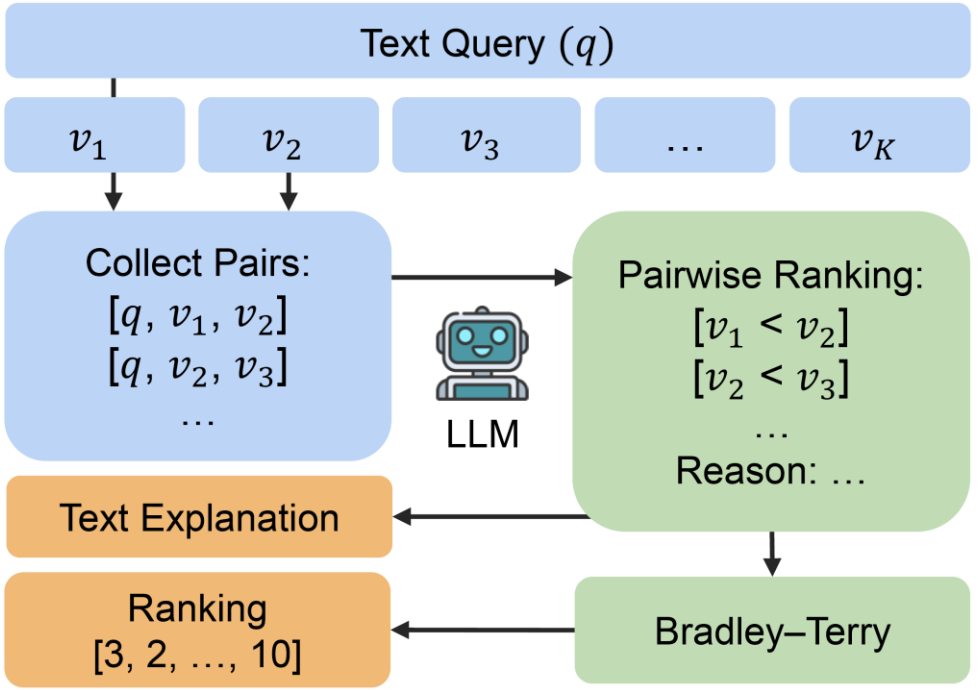
Method: Structured Video Annotations

- Generates frame-level captions using an MLLM to describe each sampled frame.
- Uses an LLM to produce structured annotations (objects, actions, scenes, summaries).
- Additional post-processing to output a JSON file.



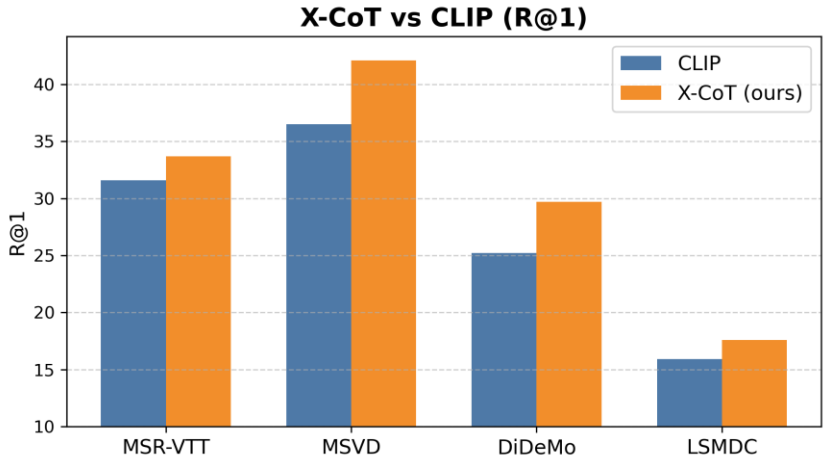
Method: X-CoT Framework

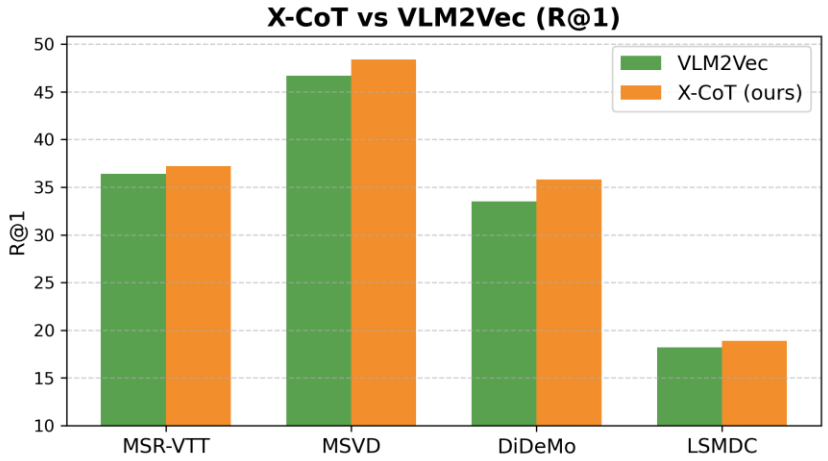
- Step 1: Embedding model → top-K videos ($\mathcal{V} = \{v_1, v_2 \ldots, v_k\}$).
- Step 2: Pairwise LLM comparison → pairwise ranking + explanations.
- Step 3: Pairwise explanations + Bradley-Terry aggregation → refined ranking + text explanation.

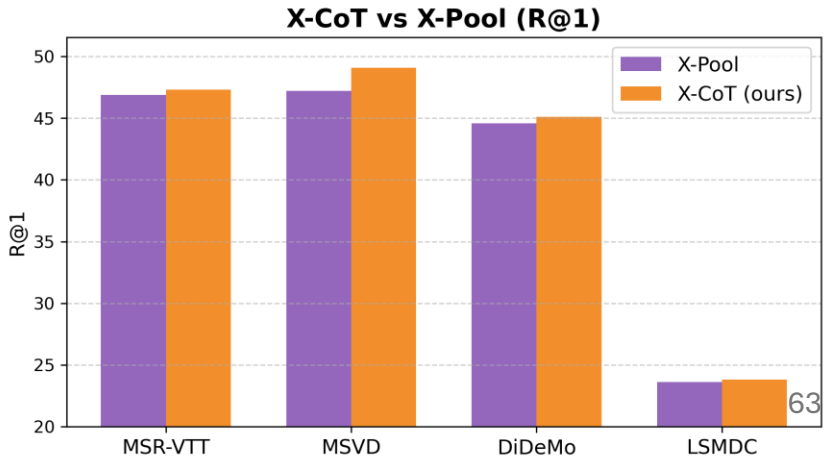


Results & Explainability

- Consistent improvements across four major benchmarks (e.g., **+5.6% R@1** on MSVD with CLIP).
- Provides human-readable rationales explaining ranking decisions.
- Facilitates model and data quality analysis, revealing caption noise and semantic focus.

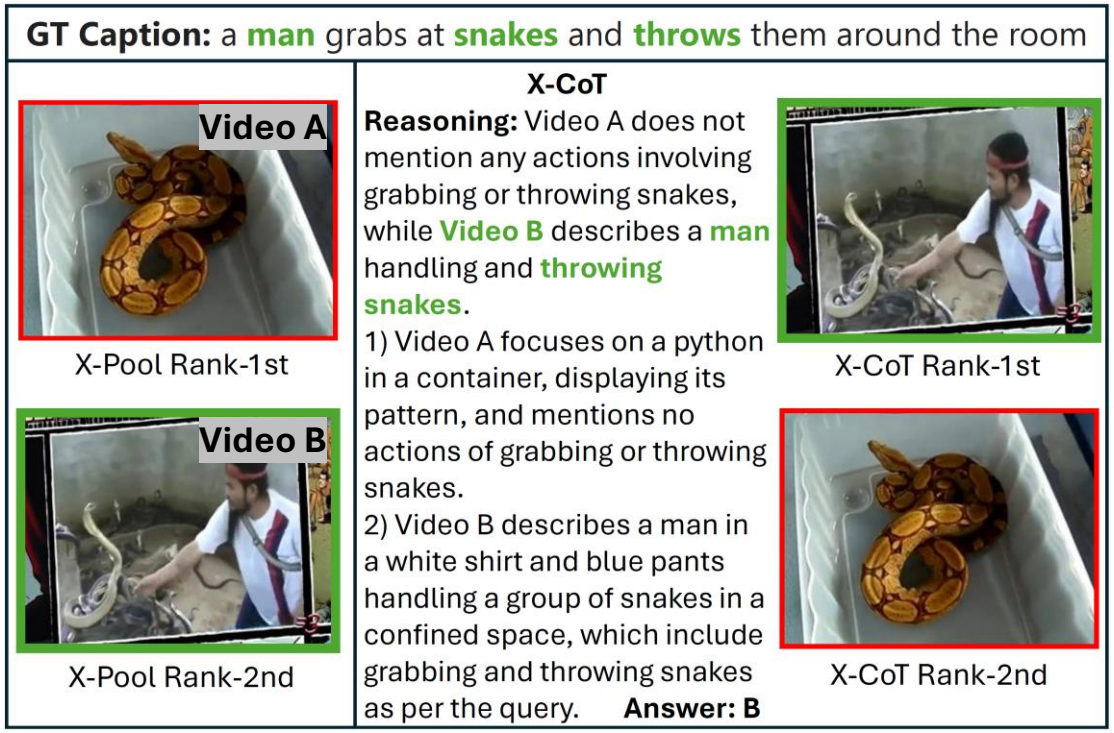






Explanation & Insights

- Reveals **semantic focus and missed concepts** in embedding models.
- Example: X-CoT correctly detects *a "man throwing snakes"* missed by the embedding model.
- Identifies **noisy or ambiguous captions**.

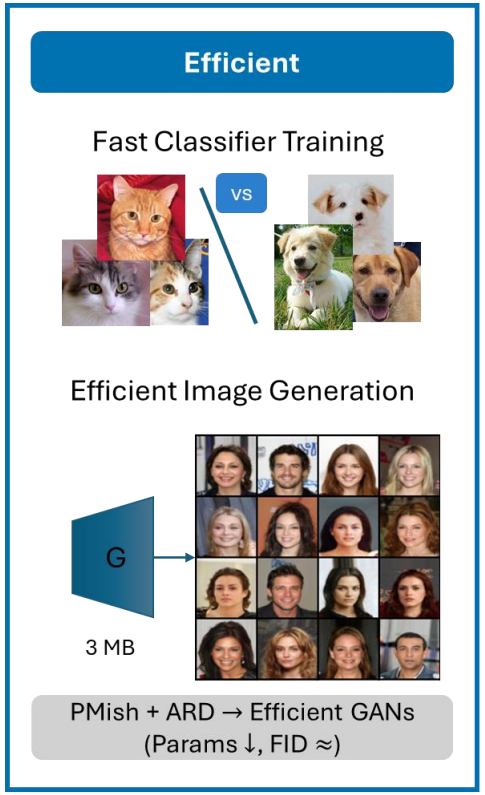


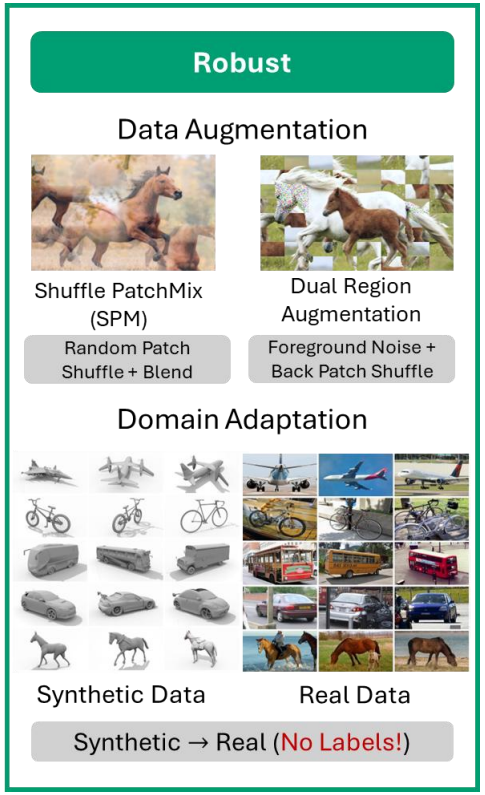
X-CoT provides human-readable explanations for ranking decisions.

Recap & Conclusion

This thesis dissertation aims at improving **efficiency**, **robustness**, and **explainability** of **Machine Learning (ML)** models.

1. Fast iterative classifier training
2. Efficient GAN for image generation
3. Robust ML models with data augmentation
4. Synthetic video generation using GANs for human action recognition
5. Advancements in domain adaptation
6. Explainable multimodal retrieval





Thank You!

Questions?