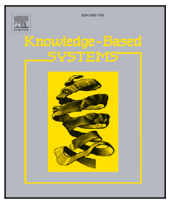
Knowledge-Based Systems 352 (2026) 117080



Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys





PatchMix: Local and contextual modeling with patch-wise mixed image for domain adaptive semantic segmentation ☆

Meiqin Liu [a,b], Linfeng He [a,b], Zilin Wang [a,b], Chao Yao [c,*], Yunchao Wei [a,b]

[a] Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China
[b] Visual Intelligence + X International Cooperation Joint Laboratory of MOE, Beijing Jiaotong University, Beijing 100044, China
[c] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Keywords:
Domain adaptation
Semantic segmentation
Self-training
Consistency training
Masked-and-mixed image modeling

ABSTRACT

Unsupervised domain adaptive semantic segmentation (UDASS) aims to transfer a model trained on an annotated source domain to an unlabeled target domain, thereby avoiding expensive pixel-level labeling when the two domains exhibit substantial distribution shifts. In practice, UDASS is hindered by two persistent challenges: achieving fine-grained local discrimination and effectively exploiting scene context. To address these issues, we propose PatchMix, a training-time regularization formulation based on a mix–encode–replace–semantic-reconstruct procedure. Two target images are combined on a random patch grid into one mixed input, which is processed by the student encoder. Before decoding, feature positions corresponding to patches from the second image are replaced with scale-wise learnable $[MASK]$ tokens. The decoder is then supervised to reproduce the semantic prediction of the complete first image generated by an exponential-moving-average teacher. Consequently, the encoder must recognize local content under mismatched context, while the decoder must recover missing semantics from retained evidence. The proposed strategy can be integrated into diverse UDASS pipelines and backbone architectures, ranging from CNNs to Transformers, and the PatchMix branch is removed at inference. Experiments on the standard synthetic-to-real benchmarks GTA5 → Cityscapes and SYNTHIA → Cityscapes demonstrate consistent improvements across the evaluated baselines.

1. Introduction

Semantic segmentation [1–3] is a fundamental problem in computer vision, where the goal is to assign a semantic label to every pixel in an image. With the rapid progress of deep convolutional and Transformer-based architectures [4–7], the performance of semantic segmentation models has been substantially improved on standard benchmarks. Beyond generic architectures, recent studies in Knowledge-Based Systems have also examined semantic segmentation under distribution shift, including knowledge-based domain adaptation (KDN) [8], progressive domain adaptation (PDA) [9], and domain-generalized token linking with vision foundation models [10].

Despite this progress, state-of-the-art segmentation models typically rely on large-scale, densely annotated datasets to generalize well. Acquiring pixel-level labels is highly labor-intensive; for example, annotating a single Cityscapes image takes more than 1.5 h on average [11]. A common strategy to reduce annotation cost is to leverage synthetic datasets such as GTA5 [12] and SYNTHIA [13], where pixel-wise labels can be generated automatically. However, models trained solely on synthetic data often transfer poorly to real-world images due to substantial domain gaps in layout, texture, and appearance. Unsupervised domain adaptive semantic segmentation (UDASS) has therefore emerged as a promising approach to bridge this gap without requiring additional manual labels on the target domain.

Existing UDASS methods can be roughly grouped into adversarial-learning-based and self-training-based methods. Adversarial methods align source and target distributions in feature space [14–17], prediction space [14–18], or image space [15–17,19], but typically introduce additional components such as discriminators or image translation networks, which increase the optimization complexity. Self-training methods [20–23] instead supervise the target domain using pseudo labels produced by a teacher or an earlier model, and refine pseudo labels via confidence estimation [24–26], class-wise reweighting [27–29], or prototype-based aggregation [30].

Masked-image modeling [31–34] and mixed-image strategies [35–37] have recently been shown to be effective for visual representation

☆ This work is supported by the National Natural Science Foundation of China under Grant 62372036, Grant 62120106009, and Grant 62332017.
* Corresponding author.
E-mail addresses: mqliu@bjtu.edu.cn (M. Liu), linfenghe@bjtu.edu.cn (L. He), 21120306@bjtu.edu.cn (Z. Wang), yaochao@ustb.edu.cn (C. Yao), yunchao.wei@bjtu.edu.cn (Y. Wei).

https://doi.org/10.1016/j.knosys.2026.117080
Received 6 December 2025; Received in revised form 17 August 2026; Accepted 15 September 2026
Available online 21 September 2026
0950-7051/© 2026 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

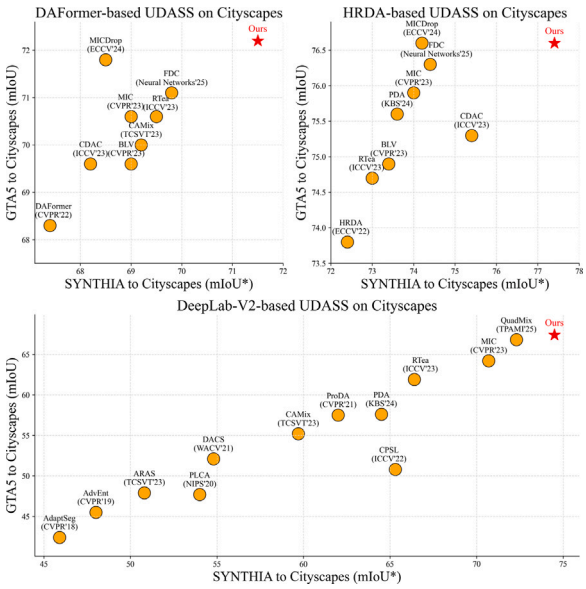


Fig. 1. Published performance overview on two synthetic-to-real benchmarks, grouped by segmentation architecture. The horizontal coordinate reports the standard 13-class mIoU* on SYNTHIA → Cityscapes, and the vertical coordinate reports mIoU on GTA5 → Cityscapes. QuadMix is included in the DeepLab-V2 panel, and PatchMix is marked by ★.

learning and training regularization in various vision tasks [37–39]. These ideas have also been adopted in UDASS. For example, MIC [40] performs masked-image consistency learning on the target domain to encourage long-range context modeling, while DACS [37] mixes source and target regions within an image to sharpen decision boundaries around mixed areas. Complementary to these, feature-dropout consistency methods such as FDC [41] stabilize pseudo labels by enforcing prediction consistency. However, most existing masked, mixed, and consistency-based methods still focus predominantly on either local details or contextual cues, and comparatively little work has explicitly targeted both aspects at the same time in UDASS.

In this work, we present PatchMix, a training-time regularization module designed to strengthen both local discrimination and context utilization. Instead of merely masking randomly selected patches, PatchMix replaces them with spatially corresponding visible patches sampled from another target-domain image, producing a patch-wise mixed image that contains content from both images. Feeding this mixed image into the student encoder creates a structured perturbation and requires the encoder to operate in a mixed feature space. Before decoding, the feature positions associated with the second image are replaced, at every encoder scale, by learnable $[MASK]$ tokens. The decoder receives these replaced features and reconstructs only the semantic prediction of the complete first image, using pseudo labels from an exponential-moving-average (EMA) teacher. This design requires the encoder to recognize retained local content under mismatched context and the decoder to infer missing semantics from the remaining evidence. Thus, the contribution lies in the complete mix–encode–replace–semantic-reconstruct training formulation, rather than in image mixing or masking alone.

We validate PatchMix on two representative UDASS benchmarks, *i.e.*, GTA5 → Cityscapes and SYNTHIA → Cityscapes. With DeepLab-V2 and a ResNet-101 backbone [2,4], it attains 67.4% mIoU on GTA5 → Cityscapes and 66.1% mIoU on SYNTHIA → Cityscapes. When integrated into DAFormer [21], PatchMix improves the listed baseline by 3.9 points on GTA5 → Cityscapes and 3.1 points on SYNTHIA → Cityscapes. Moreover, PatchMix can be combined with adversarial and self-training pipelines and applied to multiple backbone architectures, including ResNet-101, MiT-B5 [42], and Swin Transformer [43]. We summarize the published benchmark results by segmentation architecture in Fig. 1. Detailed benchmark comparisons and controlled paired gains are provided in Section 4.2.

Our main contributions are summarized as follows:

- We formulate a target-domain mix–encode–replace–semantic-reconstruct branch: real patches from a second target image perturb the encoder, whereas scale-wise feature replacement before the decoder preserves a single semantic reconstruction target.
- We use an EMA teacher on the complete first target image to impose semantic consistency on the perturbed student view, connecting local-content discrimination with context-based recovery in one training objective.
- Extensive experiments on standard synthetic-to-real benchmarks and mechanism-aligned diagnostics show consistent improvements across several UDASS pipelines and backbone architectures, from CNNs to Transformers, while the PatchMix branch is removed at inference.

2. Related work

2.1. Semantic segmentation

Semantic segmentation assigns a semantic category to every pixel in an image and thus decomposes the scene into regions with distinct semantic meanings. With the rise of deep learning, fully convolutional networks [1] have substantially advanced this task by enabling end-to-end dense prediction. Subsequent work has focused on enlarging the effective receptive field and aggregating contextual cues to better handle complex scenes. For instance, Chen et al. [2] employ atrous convolutions to increase the receptive field while controlling the model size and computational cost, and Lin et al. [7] further exploit interactions between learnable structure tokens and image features to refine segmentation results. Recently, Transformer architectures have been widely adopted in various vision tasks, including image classification [5,43], object detection [6], and semantic segmentation [18,42]. Transformer-based models often surpass convolutional backbones such as ResNet [4] in terms of accuracy, but typically require large quantities of high-quality pixel-level annotations. Datasets like Cityscapes [11] and PASCAL VOC [44] contain thousands of finely annotated images, making data collection expensive and time-consuming.

2.2. Unsupervised domain adaptive semantic segmentation

Existing methods for unsupervised domain adaptive semantic segmentation (UDASS) can be broadly categorized into adversarial-learning-based and self-training-based approaches. Adversarial methods typically adopt a generator–discriminator framework. The generator, usually a segmentation network with a ResNet backbone, produces dense predictions or serves as a feature extractor, while the discriminator distinguishes whether the feature representations [27–29], logits [14,19,45], or even images [15–17] originate from the source or target domain. For example, Tsai et al. [14] align the prediction space of source and target domains via adversarial learning. Luo et al. [45] incorporate class-wise information and propose a category-level adversarial scheme, while Tian et al. [18] estimate the degree of category-level asymmetry between domains and filter out unreliable categories accordingly.

Self-training approaches [20–23,30] generally rely on a teacher–student paradigm. A teacher produces high-confidence pseudo labels for unlabeled target images, and these labels supervise a student with the same prediction task. In mean-teacher variants, the teacher shares the student's architecture but is updated as an EMA of the student rather than optimized by back-propagation. To mitigate pseudo-label

noise, existing methods use confidence-based filtering [24–26], class-wise thresholding [46], or prototype-based weighting [20,30]. Mei et al. [46] adapt the pseudo-labeling threshold for each category to reduce the influence of hard classes, whereas Zhang et al. [30] compute distances between features and class prototypes to reweight pseudo logits.

Motivated by the success of Transformers, recent work [21,22,40,47] has introduced Transformer-based backbones and masked semantic objectives into UDASS. Hoyer et al. [21] systematically compare architectural choices and propose DAFormer and HRDA [22], which use a SegFormer-like encoder [42] and context-aware feature fusion. These architectures achieve consistently strong performance across benchmarks. PatchMix is compatible with both CNN- and Transformer-based pipelines because it augments the training procedure without replacing their inference architectures.

2.3. Masked and mixed image modeling

Inspired by the success of BERT [48], masked image modeling (MaskIM) has become an effective pretext task for visual representation learning. Methods such as MAE [31] and SimMIM [32] randomly mask out a subset of image patches and train the network to reconstruct the missing content, forcing the model to capture meaningful visual structures. In UDA, MIC [40] applies a related idea by enforcing consistency between masked target images and their corresponding complete pseudo-label maps.

Another line of work exploits mixed-image augmentations. Mixing strategies have proven effective in pretraining [49], classification [35,36], and semantic segmentation [37–39]. The central idea is to combine pixels from two training samples to create a strongly perturbed training image. CutMix [36] removes a rectangular region from one image and fills it with pixels from another. ClassMix and DACS [37,39] construct mixed images together with mixed semantic supervision. Dong et al. [50] combine cross-domain incremental image mixing with category contrast for domain-adaptive semantic segmentation. MixMAE [49] replaces masked tokens with visible tokens from another image for image pretraining and pixel reconstruction. More recently, QuadMix [51] performs class-aware intra- and inter-domain mixing along four directions for a unified treatment of image and video UDA-SS.

In UDASS, reliable predictions benefit from both local patch details and global contextual cues, yet most masked or mixed image modeling methods emphasize one of these aspects in isolation. The joint exploitation of local and contextual information under domain shift has been less studied. PatchMix differs from existing methods in the sequence and target of its operations. It uses target–target patch mixing to perturb the encoder, replaces feature positions associated with the second image before decoding, and reconstructs only the semantic prediction of the complete first image. MIC masks content before encoding, whereas PatchMix exposes the encoder to real but mismatched target patches. ClassMix and DACS retain both constituent images in the mixed supervision, whereas PatchMix replaces the direct features at positions supplied by the second image before decoding. Unlike the RGB objective used by MAE and MixMAE, PatchMix uses target-domain semantic pseudo labels. The contribution is therefore the combined training formulation rather than any individual mixing or masking operation.

2.4. Consistency regularization

Consistency regularization is a key technique in semi-supervised learning and UDASS, improving robustness by enforcing prediction invariance under input or feature perturbations. In the domain-adaptive setting, it is commonly used to mitigate the effect of pseudo-label noise and stabilize training. SAC [52] applies diverse data augmentations to an input image and enforces consistency across augmented views, while Wang et al. [53] further align predictions from cross-domain and self-attention, thus regularizing both attention and output spaces.

From this perspective, FDC [41] injects channel-wise feature-dropout noise and enforces consistent outputs under the perturbation. PatchMix instead defines a structured, spatial perturbation. The EMA teacher predicts the complete first target image, while the student predicts the same semantic target after target–target mixing and scale-wise feature replacement. The resulting loss is a semantic consistency constraint between an unperturbed teacher view and a mix–replace student view. Related robust-representation studies address interference suppression, adaptive decision boundaries, and structural correspondence in RF fingerprinting and point-cloud registration [54–56]. Although their tasks and protocols differ from UDASS, they provide cross-task examples of robustness objectives under structured perturbations; we cite them for this limited conceptual connection rather than as baselines or direct precursors of PatchMix.

3. Methodology

In this section, we first introduce the overall self-training pipeline for unsupervised domain-adaptive semantic segmentation (UDASS) in Section 3.1, and then describe the proposed PatchMix module in Section 3.2.

3.1. Pipeline

We consider the standard setting of unsupervised domain adaptation for semantic segmentation. Let $\mathcal{D}_S = \{(\mathbf{x}_s^n, \mathbf{y}_s^n)\}_{n=1}^{N_S}$ denote a labeled synthetic source-domain dataset with pixel-level annotations and $\mathcal{D}_T = \{\mathbf{x}_t^n\}_{n=1}^{N_T}$ an unlabeled real target-domain dataset that shares the same C semantic classes. During training, mini-batches contain labeled source images and unlabeled target images. The goal is to transfer knowledge from the labeled source data while exploiting abundant unlabeled target images through self-training. We adopt a student–teacher framework. The student $\Theta = (\Theta_e, \Theta_d)$ and teacher $\Theta' = (\Theta'_e, \Theta'_d)$ have the same encoder–decoder architecture but different update rules. At initialization, the teacher parameters are copied from the student, $\Theta' \leftarrow \Theta$.

For a labeled source image $\mathbf{x}_s$, we first apply the training crop and feed it to the student to obtain $\mathbf{p}_s = \Theta(\mathbf{x}_s)$. Given the corresponding cropped ground-truth map $\mathbf{y}_s$, the source-domain pixel-wise cross-entropy loss is formulated as:

$$\mathcal{L}_S = -\frac{1}{HW}\sum_{j=1}^{HW}\sum_{c=1}^{C}\mathbf{y}_s^{(j,c)}\log\mathbf{p}_s^{(j,c)}, \tag{1}$$

where $\mathcal{L}_S$ denotes the source-domain segmentation loss, $j \in \{1,\dots,HW\}$ indexes crop pixels, $c \in \{1,\dots,C\}$ indexes semantic classes, H and W denote the crop height and width, $\mathbf{y}_s^{(j,c)}$ is the one-hot ground-truth value, and $\mathbf{p}_s^{(j,c)}$ is the corresponding student probability.

For an unlabeled target image $\mathbf{x}_t$, the teacher generates the hard pseudo-label map $\hat{\mathbf{y}}_t$ by pixel-wise argmax without gradient computation. We retain the DACS strong-augmentation pipeline used by the underlying baselines, including color jitter, Gaussian blur, and ClassMix; geometric and mixing transformations are applied consistently to the image and its supervision, yielding $\mathbf{x}_t^A$ and $\hat{\mathbf{y}}_t^A$. Given the augmented student prediction $\mathbf{p}_t^A = \Theta(\mathbf{x}_t^A)$ and pseudo labels $\hat{\mathbf{y}}_t^A$, the target self-training loss is formulated as:

$$\mathcal{L}_T = -\frac{q_t}{HW}\sum_{j=1}^{HW}\sum_{c=1}^{C}(\hat{\mathbf{y}}_t^A)^{(j,c)}\log(\mathbf{p}_t^A)^{(j,c)}, \tag{2}$$

where $\mathcal{L}_T$ denotes the target self-training loss, $(\hat{\mathbf{y}}_t^A)^{(j,c)}$ is the augmented one-hot pseudo-label value, $(\mathbf{p}_t^A)^{(j,c)}$ is the corresponding student probability, and $q_t = \frac{1}{HW}\sum_{j=1}^{HW}\mathbb{I}[\max_c \mathbf{p}_{\Theta'}(\mathbf{x}_t)^{(j,c)} \geq \tau]$ denotes the crop-level confidence weight. Here, $\mathbf{p}_{\Theta'}(\mathbf{x}_t)$ is the teacher softmax output, τ is the confidence threshold, and $\mathbb{I}[\cdot]$ is the indicator function; the pixel

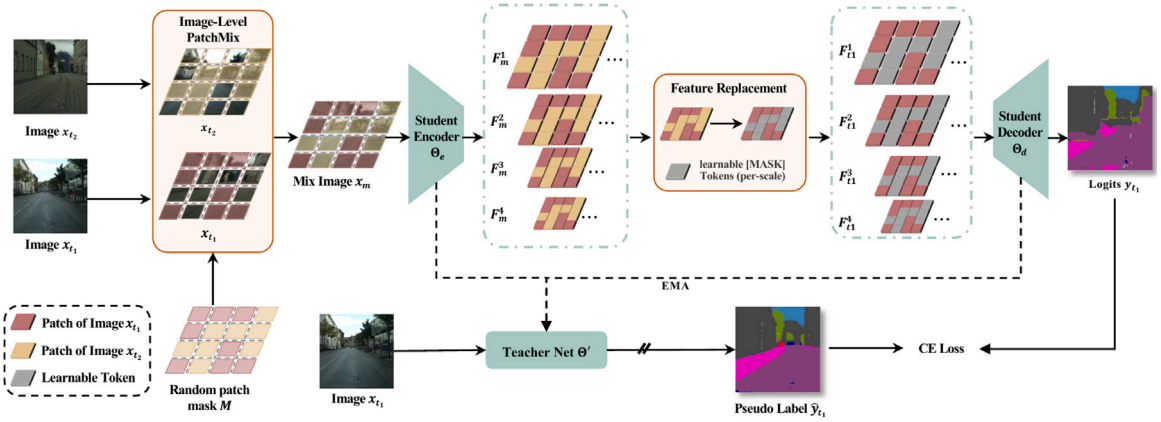


Fig. 2. PatchMix training branch. Two target images $\mathbf{x}_{t_1}$ and $\mathbf{x}_{t_2}$ are combined by a patch-source mask to form $\mathbf{x}_m$. The student encoder Θ_e processes the mixed image and produces multi-scale features $\{\mathbf{F}_m^i\}_{i=1}^K$. Before decoding, positions spatially corresponding to $\mathbf{x}_{t_2}$ patches are replaced, at each scale, by the learnable token $\mathbf{z}_{\text{mask}}^i$. The student decoder Θ_d therefore reconstructs only the semantic prediction of $\mathbf{x}_{t_1}$, supervised by pseudo labels from the EMA teacher on the complete $\mathbf{x}_{t_1}$. This branch is used only during training.

and class indices follow Eq. (1). Because q_t is the fraction of teacher predictions above τ, it scales the whole pseudo-labeled crop loss rather than deleting individual low-confidence pixels from $\hat{\mathbf{y}}_t$. The PatchMix loss introduced in Section 3.2 supplies an additional target-domain consistency constraint alongside the conventional augmented self-training branch.

Following the mean-teacher strategy [57], the gradient-free teacher update after each student step is formulated as:

$$\Theta' \leftarrow \alpha\Theta' + (1-\alpha)\Theta, \tag{3}$$

where Θ' and Θ denote the teacher and student parameters, and $\alpha \in [0,1)$ denotes the EMA momentum controlling the teacher's tracking rate.

3.2. PatchMix

Accurate target-domain segmentation requires leveraging both fine-grained local evidence and broader scene context. To strengthen these two aspects in one training branch, PatchMix combines patch-level target–target mixing with semantic reconstruction consistency, as illustrated in Fig. 2. Given two target crops $\mathbf{x}_{t_1}, \mathbf{x}_{t_2} \in \mathbb{R}^{H\times W\times 3}$ and patch size P, the crops are partitioned into an $H/P \times W/P$ grid. Each mask entry is sampled independently as $M_{uv} \sim \text{Bernoulli}(r)$, without labels, pseudo labels, uncertainty, or hard-region mining. Thus, r is the expected fraction of patches retained from $\mathbf{x}_{t_1}$ rather than an exact per-image quota. The binary patch-source mask $\mathbf{M} \in \{0,1\}^{H/P\times W/P}$ uses 1 for a patch from $\mathbf{x}_{t_1}$ and 0 for the spatially corresponding patch from $\mathbf{x}_{t_2}$. Nearest-neighbor expansion yields the pixel mask $\widetilde{\mathbf{M}} \in \{0,1\}^{H\times W}$. Given this expanded mask, the mixed target image is formulated as:

$$\mathbf{x}_m = \widetilde{\mathbf{M}} \odot \mathbf{x}_{t_1} + (1-\widetilde{\mathbf{M}}) \odot \mathbf{x}_{t_2}, \tag{4}$$

where $\mathbf{x}_m \in \mathbb{R}^{H\times W\times 3}$ denotes the mixed target crop, $\odot$ denotes element-wise multiplication, and $1-\widetilde{\mathbf{M}}$ selects pixels supplied by $\mathbf{x}_{t_2}$. For example, $\mathbf{M} = \left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right]$ selects the diagonal patches from $\mathbf{x}_{t_1}$ and the other two patches from $\mathbf{x}_{t_2}$. The resulting $\mathbf{x}_m$ contains visible target-domain patches from both images. Unlike blank or random input fillers, these patches preserve realistic target-domain texture and place each retained region of $\mathbf{x}_{t_1}$ in a deliberately mismatched but visually meaningful context. We use $P = 32$ and $r = 0.5$ by default; Section 4.3 reports their sensitivity.

The student encoder processes $\mathbf{x}_m$ and produces K mixed feature maps $\mathbf{F}_m^i \in \mathbb{R}^{H_i\times W_i\times C_i}$, which aggregate content originating from both target crops. Directly decoding these mixed features would produce a prediction influenced by two different scenes. Supervising both constituent images as simultaneous semantic targets can introduce ambiguity about which scene should be reconstructed, and the component study in Section 4.3 shows that mixed-label consistency alone provides only a limited gain. PatchMix therefore retains the mixed image as an encoder perturbation but replaces the direct features at positions supplied by the second image before decoding. For each scale i, the same patch-source mask is resized by nearest-neighbor interpolation to $\mathbf{M}_i \in \{0,1\}^{H_i\times W_i\times 1}$. Given $\mathbf{F}_m^i$ and $\mathbf{M}_i$, the decoder input at scale i is formulated as:

$$\mathbf{F}_{t_1}^i = \mathbf{M}_i \odot \mathbf{F}_m^i + (1-\mathbf{M}_i) \odot \mathbf{z}_{\text{mask}}^i, \tag{5}$$

where $\mathbf{F}_{t_1}^i \in \mathbb{R}^{H_i\times W_i\times C_i}$ denotes the scale-i feature passed to the decoder, $\mathbf{M}_i$ selects positions retained from $\mathbf{x}_{t_1}$, $\mathbf{z}_{\text{mask}}^i \in \mathbb{R}^{C_i}$ denotes the learnable token broadcast to all replaced positions, and $i \in \{1,\dots,K\}$ indexes feature scales. Replacing rather than simply deleting these features preserves an explicit placeholder through which the decoder can reason about missing content. Because feature scales have different channel dimensions, one token is used per scale; within a scale it is shared across spatial positions, images, and batches. Each token is initialized once from a truncated Gaussian distribution with standard deviation 0.02. It belongs only to the student PatchMix branch and is optimized jointly with the student by AdamW through $\mathcal{L}_M$; it is not included in the EMA teacher and has no class- or position-specific semantic meaning.

The decoder receives $\{\mathbf{F}_{t_1}^i\}_{i=1}^K$ and produces $\mathbf{p}_{t_1}^M = \Theta_d(\{\mathbf{F}_{t_1}^i\}_{i=1}^K)$. Independently, the EMA teacher predicts the complete, unmasked $\mathbf{x}_{t_1}$ and supplies the pseudo-label map $\hat{\mathbf{y}}_{t_1}$. We compute q_{t_1} from the teacher confidence on $\mathbf{x}_{t_1}$ by the same crop-level rule used for q_t in Eq. (2). Given these teacher pseudo labels and the PatchMix student prediction, the semantic reconstruction loss is formulated as:

$$\mathcal{L}_M = -\frac{q_{t_1}}{HW}\sum_{j=1}^{HW}\sum_{c=1}^{C}\hat{\mathbf{y}}_{t_1}^{(j,c)}\log(\mathbf{p}_{t_1}^M)^{(j,c)}, \tag{6}$$

where $\mathcal{L}_M$ denotes the semantic reconstruction loss, q_{t_1} denotes the crop-level confidence weight for $\mathbf{x}_{t_1}$, $\hat{\mathbf{y}}_{t_1}^{(j,c)}$ is the one-hot teacher pseudo-label value, and $(\mathbf{p}_{t_1}^M)^{(j,c)}$ is the corresponding student probability; j and c index pixels and classes as in Eq. (1). Thus, $\mathbf{x}_{t_2}$ supplies a structured perturbation to the encoder but is not a second semantic target for the decoder. The student is asked to reproduce the first image's semantic prediction from partially retained evidence rather than to recover low-level pixels or jointly decode two scenes. Combining the segmentation losses and optional feature-distance regularizer, the complete objective is formulated as:

$$\mathcal{L}_{\text{final}} = \mathcal{L}_S + \mathcal{L}_T + \mathcal{L}_M + \lambda_{FD}\mathcal{L}_{FD}, \tag{7}$$

Algorithm 1 PatchMix training procedure.

Input: $\mathcal{D}_S$, $\mathcal{D}_T$, student Θ, EMA teacher Θ', patch size P, ratio r.

Initialize the teacher from the student: $\Theta' \leftarrow \Theta$.
for *iter* $\leftarrow 1$ to N **do**
  Sample $(\mathbf{x}_s, \mathbf{y}_s) \sim \mathcal{D}_S$, $\mathbf{x}_t \sim \mathcal{D}_T$, and $(\mathbf{x}_{t_1}, \mathbf{x}_{t_2}) \sim \mathcal{D}_T^2$.
  Without gradients: obtain pixel-wise pseudo labels $\hat{\mathbf{y}}_t \leftarrow \operatorname{argmax}_c \mathbf{p}_{\Theta'}(\mathbf{x}_t)$ and $\hat{\mathbf{y}}_{t_1} \leftarrow \operatorname{argmax}_c \mathbf{p}_{\Theta'}(\mathbf{x}_{t_1})$; compute the crop-level weights q_t and q_{t_1}.
  Augment $(\mathbf{x}_t, \hat{\mathbf{y}}_t)$ consistently and compute $\mathcal{L}_S$ and $\mathcal{L}_T$ by Eqs. (1)–(2).
  Sample each entry of $\mathbf{M} \in \{0,1\}^{H/P \times W/P}$ independently from Bernoulli(r).
  Construct $\mathbf{x}_m$ from $\mathbf{x}_{t_1}$ and $\mathbf{x}_{t_2}$ by Eq. (4); compute $\{\mathbf{F}_m^i\} \leftarrow \Theta_e(\mathbf{x}_m)$.
  Resize $\mathbf{M}$ to each feature scale and replace the $\mathbf{x}_{t_2}$ positions by Eq. (5).
  Decode $\{\mathbf{F}_{t_1}^i\}$ and compute $\mathcal{L}_M$ against the pseudo labels of the complete $\mathbf{x}_{t_1}$ by Eq. (6).
  Update the student and $\{\mathbf{z}_{\text{mask}}^i\}$ with AdamW using $\mathcal{L}_{\text{final}}$ in Eq. (7).
  Update Θ' from Θ by EMA using Eq. (3).
end for
return Student segmentation network Θ.

where $\mathcal{L}_{\text{final}}$ denotes the complete objective, $\mathcal{L}_{FD}$ denotes DAFormer's thing-class ImageNet feature-distance loss [21], and λ_{FD} denotes its weight, set to zero when this regularizer is absent. The weights of $\mathcal{L}_T$ and $\mathcal{L}_M$ are one. Algorithm 1 summarizes the resulting sequence of mask sampling, feature replacement, loss computation, and student and EMA updates.

PatchMix can be interpreted as structured semantic consistency rather than formal feature disentanglement. The second image and $\mathbf{M}$ change local appearance and surrounding context, while the teacher target for $\mathbf{x}_{t_1}$ remains fixed. Minimizing $\mathcal{L}_M$ penalizes prediction changes across these views. During reconstruction, the encoder must distinguish retained local content from the perturbing target patches, which strengthens robustness to patch-level appearance changes. At the same time, the decoder cannot treat replaced positions as a second labeled image; it must use the retained local and surrounding evidence to recover missing semantics. This joint requirement connects local discrimination with context utilization while retaining a single, unambiguous semantic target. When teacher pseudo labels are reliable, the constraint reduces sensitivity to target-domain appearance perturbations. Conversely, erroneous pseudo labels or implausible image pairs can impose incorrect invariance, which defines an important limitation.

4. Experiment

4.1. Experimental setup

4.1.1. Dataset

We adopt the standard synthetic-to-real setting for unsupervised domain adaptive semantic segmentation (UDASS). GTA5 [12] and SYNTHIA [13] serve as labeled synthetic source domains, while Cityscapes [11] is used as the unlabeled real target domain. GTA5 contains 24,966 images rendered from "Grand Theft Auto V", depicting urban driving scenes at 1914 × 1052 resolution. For SYNTHIA, we follow [14] and use the SYNTHIA-RAND-CITYSCAPES subset consisting of 9400 images at 1280 × 760 with 16 classes aligned to Cityscapes. Cityscapes is collected from 50 European cities and provides 2975 training and 500 validation images at 2048 × 1024. In our UDASS setting, we treat the Cityscapes training split as unlabeled target data and evaluate all methods on the validation set.

4.1.2. Network architecture

We build our approach on two strong Transformer-based UDASS frameworks, DAFormer [21] and HRDA [22], which serve as our main baselines. DAFormer [21] uses a MiT-B5 Transformer encoder with a carefully designed decoder for domain adaptation, while HRDA [22] further incorporates multi-resolution branches and a context-aware fusion strategy.

To verify that PatchMix is not restricted to Transformer architectures, we additionally use DeepLab-V2 [2] with a ResNet-101 backbone [4]. DeepLab-V2 uses an Atrous Spatial Pyramid Pooling head with dilation rates $\{6, 12, 18, 24\}$, whereas DAFormer and HRDA retain their original decoders and fusion design. Thus, PatchMix changes only the training branch and does not replace the segmentation head of a baseline. All backbone encoders are initialized from ImageNet [58] pretraining.

4.1.3. Implementation detail

All accuracy experiments are implemented in PyTorch and run on two Tesla V100 GPUs. We employ AdamW [59] as the optimizer with decoupled weight decay. The learning rate is initialized to 6×10^{-5} for the encoder and 6×10^{-4} for the decoder. We set the weight decay to 0.01 and the AdamW momentum parameters to $(\beta_1, \beta_2) = (0.9, 0.999)$. A linear warm-up schedule is applied during the first 1500 iterations, followed by a linear decay for the remaining training.

Input images are resized to 1280 × 720 (GTA5), 1280 × 760 (SYNTHIA), and 1280 × 640 (Cityscapes). We train with random crops of size 640 × 640, a batch size of 4, and a total of 40,000 iterations. Unless otherwise specified, PatchMix uses fixed 32 × 32 patches, a mixing ratio $r = 0.5$, and scale-wise tokens initialized once with a truncated Gaussian distribution of standard deviation 0.02. We set the feature-distance weight [21] to $\lambda_{FD} = 0.005$, the EMA momentum to $\alpha = 0.999$, and the pseudo-label confidence threshold to $\tau = 0.95$. The patch-size and ratio choices are examined in Section 4.3.

4.1.4. Testing

At test time, image pairing, mask generation, the EMA teacher, scale-wise tokens, feature replacement, and $\mathcal{L}_M$ are removed. Only the original student encoder and decoder remain, and a single forward pass is used. PatchMix therefore does not change the inference architecture or its parameterized computation. Matched training and inference measurements are reported in Section 4.3.10.

4.1.5. Evaluation

We evaluate all methods on Cityscapes validation images resized to 1280 × 640. We adopt Intersection-over-Union (IoU), the standard metric for semantic segmentation, and report both the IoU of each semantic class and the mean IoU (mIoU) over all evaluated categories. Following prior work [14,21,22,30,60], GTA5 → Cityscapes uses 19 categories, while SYNTHIA → Cityscapes uses both the 16-category mIoU and the standard 13-category mIoU*.

4.2. Comparison with state-of-the-art methods

To assess the effectiveness of PatchMix, we compare it with representative UDASS approaches based on the ResNet [4] backbone, including:

- Adversarial-learning methods: AdaptSeg [14] and ADVENT [61].
- Self-training methods: PLCA [62], DACS [37], ProDA [30], CPSL [20], ARAS [63], and PDA [9].

We further compare PatchMix with recent methods evaluated with Transformer-based architectures, including DAFormer [21], HRDA [22], BLV [64], MIC [40], CDAC [53], RTea [65], CAMix [66], MICDrop [67], and FDC [41]. Tables 1 and 2 also include the published QuadMix [51] results in the DeepLab-V2 group, allowing a direct comparison between the two mixing methods on both benchmarks.

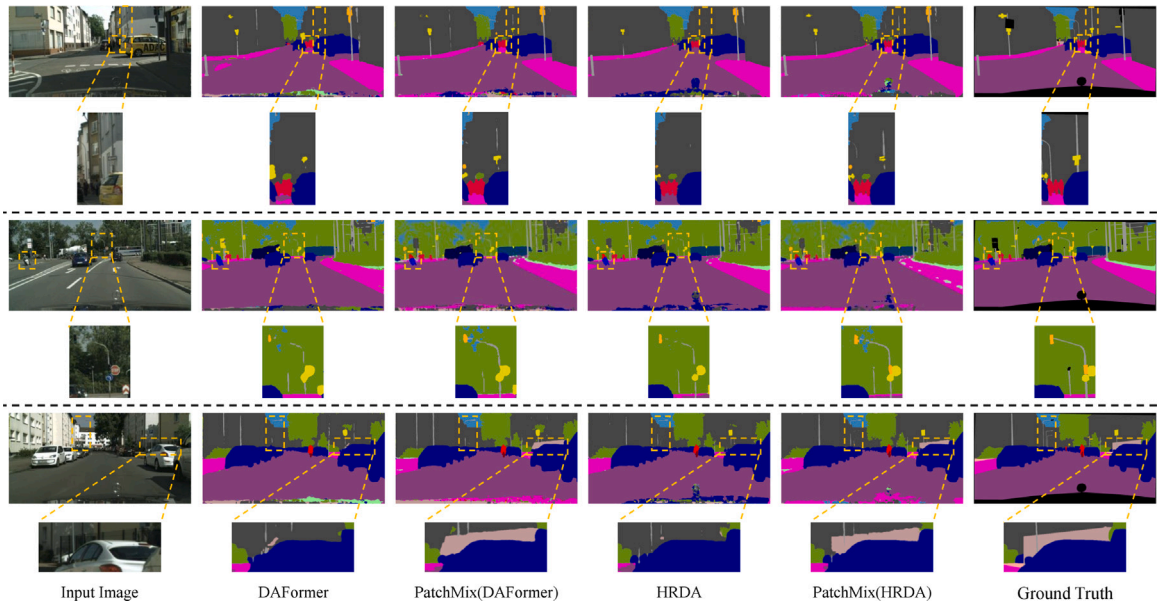


Fig. 3. Qualitative results for GTA5 → Cityscapes. From left to right: target image, DAFormer, DAFormer + PatchMix, HRDA, HRDA + PatchMix, and ground truth. Yellow dashed boxes mark representative differences.

4.2.1. Results on GTA5 → Cityscapes

We begin by evaluating PatchMix on the GTA5 → Cityscapes benchmark using the DeepLab-V2 backbone and comparing it with representative UDASS approaches [18,30,40,64], as summarized in Table 1. Several notable observations emerge:

- When combined with DeepLab-V2, PatchMix achieves 67.4% mIoU, improving the source-only baseline by 28.8 points. It exceeds QuadMix [51] by 0.6 points, MIC [40] by 3.2 points, and PDA [9] by 7.8 points within the DeepLab-V2 group.
- PatchMix brings notable gains on challenging categories. For example, it reaches 51.8% IoU on the "train" class, above several representative methods in the same DeepLab-V2 group [30,40,64]. Similar improvements occur for thing classes such as "rider", consistent with stronger local discrimination for hard-to-recognize objects.

We then apply PatchMix to Transformer-based architectures, i.e., DAFormer [21] and HRDA [22], and compare with recent transformer-based UDASS methods [40,41,53,64–67]. The results on GTA5 → Cityscapes are also reported in Table 1, from which we can draw the following conclusions:

- With DAFormer, PatchMix attains 72.2% mIoU, exceeding FDC [41] (71.1%) and MICDrop [67] (71.8%) by 1.1 and 0.4 points, respectively. On HRDA, PatchMix reaches 76.6% mIoU, matching MICDrop and remaining 0.3 points above FDC (76.3%).
- Across the 19 categories, PatchMix is competitive and gives clear gains on classes such as "traffic light", "bus", and "train", showing that the training branch also benefits Transformer-based architectures.

Under DeepLab-V2, PatchMix obtains 67.4 mIoU compared with 66.8 mIoU for QuadMix, giving PatchMix a 0.6-point advantage on GTA5 → Cityscapes.

PatchMix is not best on every class. Within the DAFormer group, its terrain, sky, traffic-sign, and truck IoUs are 48.7, 89.3, 63.2, and 79.2, below the group maxima of 52.4, 94.5, 66.7, and 82.5. With HRDA, wall and traffic-sign IoUs are 62.8 and 68.3, compared with 70.8 for PDA and 73.6 for MICDrop. These class-wise gaps identify a remaining sensitivity to both broad layout-dependent regions and thin boundaries, which may be disrupted by unrelated image contexts or a fixed patch grid. Semantic pairing, uncertainty-weighted reconstruction, and boundary-adaptive patches are possible directions for addressing this limitation.

4.2.2. Results on SYNTHIA → Cityscapes

We next report results on the SYNTHIA → Cityscapes benchmark and compare PatchMix with state-of-the-art UDASS methods [40,63]. We summarize the 16-category and 13-category evaluation performance in Table 2. The following findings can be observed:

- PatchMix attains the highest IoU in the DeepLab-V2 group for several categories, including "bus" and "motorcycle", while QuadMix records the highest "traffic light" IoU.
- PatchMix obtains 66.1% mIoU and 74.5% mIoU* over the 16- and 13-category evaluations, respectively. Relative to MIC [40], the gains are 3.3 and 3.8 points; relative to QuadMix [51], the gains are 3.5 and 2.2 points.

We also present the SYNTHIA → Cityscapes results for Transformer-based architectures in Table 2. We summarize the observations as follows:

- On the DAFormer backbone, PatchMix reaches 64.0% mIoU and 71.2% mIoU* for the 16- and 13-category evaluations. Compared with FDC [41] (63.6% / 69.8%), the corresponding gains are 0.4 and 1.4 points; PatchMix also exceeds earlier conference methods such as RTea [65].
- On HRDA, PatchMix achieves 70.1% mIoU and 77.4% mIoU*. It exceeds CDAC [53] by 1.4 and 2.0 points and FDC (68.1% / 74.4%) by 2.0 and 3.0 points on the two metrics, respectively.
- The class-wise results are particularly strong for "traffic light", "rider", and "bus", although the magnitude varies across categories.

Under DeepLab-V2, PatchMix obtains 66.1% mIoU and 74.5% mIoU*, compared with 62.6% and 72.3% for QuadMix. The corresponding gains are 3.5 and 2.2 points on the 16- and 13-category evaluations, respectively.

The most consistent SYNTHIA gaps occur for wall and fence: PatchMix obtains 37.1/5.7 with DAFormer and 44.0/7.4 with HRDA. These categories, together with pole, are excluded from the standard 13-class mIoU*. Their thin or ambiguous boundaries may be especially

Table 1
Comparison on GTA5 [12] → Cityscapes [11]. Approaches marked with † use distillation-based training. Within each architecture group, red and blue denote the highest and second-highest value in each column, respectively.

Network	Methods	road	side	buil	wall	fence	pole	light	sign	veg	terr	sky	pers	rider	car	truck	bus	train	mbike	bike	mIoU
DeepLabV2	Source Only	70.2	14.6	71.3	24.1	15.3	25.5	32.1	13.5	82.9	25.1	78.0	56.2	33.3	76.3	26.6	29.8	12.3	28.5	18.0	38.6
	AdaptSegCVPR2018 [14]	86.5	36.0	79.9	23.4	23.3	23.9	35.2	14.8	83.4	33.3	75.6	58.5	27.6	73.7	32.5	35.4	3.9	30.1	28.1	42.4
	ADVENTCVPR2019 [61]	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5
	PLCANIPS2020 [62]	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	41.8	47.7
	DACSWACV2021 [37]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
	ProDACVPR2021† [30]	87.8	56.0	79.7	46.3	**44.8**	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
	CPSLCVPR2022† [20]	92.3	59.9	84.9	45.7	29.7	**52.8**	61.5	59.5	87.9	41.5	85.0	73.0	35.5	90.4	48.7	**73.9**	26.3	**53.8**	53.9	60.8
	BLVCVPR2023 [64]	94.9	68.2	88.8	40.9	37.1	42.6	52.1	62.1	88.3	43.3	89.3	68.6	44.5	88.9	56.0	54.6	3.8	38.6	58.3	59.0
	MICCVPR2023 [40]	96.5	74.3	**90.4**	**47.1**	42.8	50.3	**61.7**	62.3	90.3	49.2	**90.7**	77.8	53.2	**93.0**	**66.2**	68.0	6.8	38.0	60.6	64.2
	RTeaICCV2023 [65]	95.4	67.1	87.9	46.1	44.0	46.0	53.8	59.5	89.7	**49.8**	89.8	71.5	40.5	90.8	55.0	57.9	22.1	47.7	**62.5**	61.9
	ARASTCSVT2023 [63]	91.9	45.2	81.8	21.9	25.6	35.5	41.5	33.4	85.1	34.8	73.8	62.5	31.6	85.9	33.8	42.5	7.3	33.8	42.8	47.9
	CAMixTCSVT2023 [66]	93.3	58.2	86.5	36.8	31.5	36.4	35.0	43.5	87.2	44.6	88.1	65.0	24.7	89.7	46.9	56.8	**27.5**	41.1	56.0	55.2
	PDAKBS2024 [9]	94.1	58.8	86.7	45.8	43.1	43.6	51.3	53.6	89.3	47.4	86.3	65.3	37.9	89.2	64.8	62.6	6.9	46.5	59.1	59.6
	QuadMixTPAMI2025 [51]	**97.1**	**78.0**	**90.4**	**49.7**	40.3	**53.4**	**61.7**	**70.9**	**90.7**	**49.7**	**92.9**	**77.9**	**53.9**	**93.5**	**72.4**	65.9	0.7	**60.5**	**68.5**	**66.8**
	PatchMix(Ours)	**96.7**	**75.1**	**90.5**	43.6	**45.9**	52.0	**64.7**	**69.5**	**90.4**	48.6	90.6	**80.1**	**59.4**	**93.0**	64.5	**75.3**	**51.8**	35.0	53.1	**67.4**
DAFormer	DAFormerCVPR2022 [21]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
	BLVCVPR2023 [64]	96.2	73.1	89.3	53.6	**55.7**	50.9	55.7	61.1	89.7	**52.4**	92.3	74.7	43.5	91.6	74.6	77.4	69.2	**58.9**	62.3	69.6
	MICCVPR2023 [40]	96.7	75.0	90.0	58.2	50.4	51.1	56.7	62.1	90.2	51.3	92.9	72.4	**47.1**	92.8	78.9	83.4	75.6	54.2	62.6	70.6
	CDACICCV2023 [53]	96.5	73.9	89.5	56.8	48.9	50.7	55.8	63.3	89.9	49.1	91.2	72.2	45.4	92.7	78.3	82.9	67.5	55.2	63.4	69.6
	RTeaICCV2023 [65]	96.1	71.7	89.1	57.8	50.4	**55.9**	59.3	**66.7**	**90.4**	48.2	**94.5**	**74.8**	46.5	**93.8**	78.7	81.6	65.8	57.1	62.8	70.6
	CAMixTCSVT2023 [66]	96.0	73.1	89.5	53.9	50.8	51.7	58.7	64.9	90.0	51.2	92.2	71.8	44.0	92.8	78.7	82.3	70.9	54.1	**64.3**	70.0
	MICDropECCV2024 [67]	96.5	74.2	**90.8**	**60.5**	52.0	**55.8**	**59.9**	65.6	90.3	**51.8**	93.0	73.1	46.9	**93.4**	**82.0**	85.8	74.3	56.6	62.8	**71.8**
	UDANeurocomp.2025 [50]	**96.8**	73.4	89.7	54.2	47.9	51.2	57.1	62.3	**91.6**	**51.8**	**93.5**	72.5	45.4	93.2	77.3	76.8	62.8	59.0	63.8	69.5
	FDCNN2025 [41]	**97.0**	**77.0**	90.3	**60.5**	48.4	52.6	56.6	**65.4**	90.2	48.5	92.7	71.3	45.5	93.1	**82.5**	**87.2**	**77.1**	52.1	62.5	71.1
	PatchMix(Ours)	96.6	**77.1**	**90.6**	**58.7**	**54.0**	54.5	**61.9**	63.2	90.0	48.7	89.3	**75.6**	**50.0**	92.4	79.2	**86.1**	**78.9**	**60.8**	**64.2**	**72.2**
HRDA	HRDAECCV2022 [22]	96.4	74.4	91.0	61.6	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79.0	52.9	93.9	84.1	85.7	75.9	63.9	67.5	73.8
	BLVCVPR2023 [64]	96.7	76.6	91.5	61.2	56.9	59.4	62.2	**72.8**	91.5	51.2	94.3	77.5	54.7	93.5	83.2	84.7	79.7	**68.1**	67.6	74.9
	MICCVPR2023 [40]	97.4	80.1	91.7	61.2	56.9	59.7	66.0	71.3	91.7	51.4	94.3	79.8	56.1	94.6	85.4	90.3	80.4	64.5	68.5	75.9
	CDACICCV2023 [53]	97.1	78.7	91.8	59.6	57.1	59.1	66.1	72.2	**91.8**	**53.1**	94.5	79.4	51.6	94.6	84.9	87.8	78.7	64.9	67.6	75.3
	RTeaICCV2023 [65]	97.1	75.2	**92.6**	**63.5**	51.8	58.2	66.5	71.2	91.1	49.0	**96.8**	**81.5**	54.2	94.2	84.8	86.6	75.7	62.2	66.7	74.7
	MICDropECCV2024 [67]	**97.6**	**81.5**	92.0	62.8	**59.4**	**62.6**	62.9	**73.6**	91.6	52.6	94.1	80.2	**57.0**	**94.8**	**87.4**	90.7	81.6	65.3	67.8	**76.6**
	PDAKBS2024 [9]	**97.8**	78.1	**92.4**	**70.8**	53.1	58.5	58.2	67.1	**92.0**	**54.5**	**94.8**	**81.4**	54.5	92.9	86.9	87.1	78.6	**68.1**	**69.7**	75.6
	FDCNN2025 [41]	97.2	79.1	91.6	59.6	**57.3**	**61.5**	**67.2**	**72.8**	**91.8**	52.9	93.8	80.2	**57.1**	**94.8**	86.6	**90.8**	**82.5**	64.2	69.4	**76.3**
	PatchMix(Ours)	97.3	**81.4**	91.7	62.8	56.5	60.7	**68.3**	68.3	91.7	50.2	94.3	81.2	54.6	**94.9**	**88.0**	**92.0**	**83.4**	**65.4**	**72.0**	**76.6**

Table 2
Evaluation on the SYNTHIA [13] → Cityscapes [11] benchmark. We report mIoU over all 16 categories and mIoU* over the standard 13-class subset; categories marked with * are excluded from mIoU*. Within each architecture group, red and blue denote the highest and second-highest value in each column, respectively.

Network	Methods	road	side	buil	wall*	fence*	pole*	light	sign	veg	sky	pers	rider	car	bus	mbike	bike	mIoU	mIoU*
DeepLabV2	Source Only	55.6	23.8	74.6	9.2	0.2	24.4	6.1	12.1	74.8	79.0	55.3	19.1	39.6	23.3	13.7	25.0	33.5	38.6
	AdaptSegCVPR2018 [14]	79.2	37.2	78.8	10.5	0.3	25.1	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	39.5	45.9
	ADVENTCVPR2019 [61]	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
	PLCANIPS2020 [62]	82.6	29.0	81.0	11.2	0.2	33.6	24.9	18.3	82.8	82.3	62.1	26.5	85.6	48.9	26.8	52.2	46.8	54.0
	DACSWACV2021 [37]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	**90.8**	67.6	38.3	82.9	38.9	28.5	47.6	48.3	54.8
	ProDACVPR2021† [30]	87.8	45.7	84.6	**37.1**	0.6	44.0	54.6	37.0	88.1	84.4	74.2	24.3	88.2	51.1	40.5	45.6	55.5	62.0
	CPSLCVPR2022† [20]	87.2	43.9	85.5	33.6	0.3	47.7	57.4	37.2	87.8	88.5	79.0	32.0	**90.6**	49.4	50.8	59.8	57.9	65.3
	BLVCVPR2023 [64]	70.4	28.9	**89.2**	25.2	**19.9**	40.2	55.2	50.3	86.9	84.2	76.4	40.5	79.6	51.3	49.2	61.2	56.8	63.3
	MICCVPR2023 [40]	84.7	45.7	**88.3**	29.9	2.8	**53.3**	61.0	**59.5**	86.9	88.8	78.2	53.3	89.4	58.8	56.0	**68.3**	**62.8**	70.7
	RTeaICCV2023 [65]	**93.2**	**59.6**	86.3	31.3	4.8	43.1	41.8	44.0	**88.6**	**90.5**	70.4	42.6	89.5	56.7	40.2	59.9	58.9	66.4
	ARASTCSVT2023 [63]	85.6	39.2	79.9	15.5	0.3	32.2	19.3	23.9	79.1	81.7	61.1	19.3	82.9	25.7	10.6	51.9	44.3	50.8
	CAMixTCSVT2023 [66]	91.8	54.9	83.6	-	-	-	23.0	29.0	83.8	87.1	65.0	26.4	85.5	55.1	36.8	54.1	-	59.7
	PDAKBS2024 [9]	88.3	48.1	**89.2**	**36.8**	0.2	46.2	51.7	44.1	88.5	89.3	78.5	26.6	89.1	50.5	42.1	51.8	57.6	64.5
	QuadMixTPAMI2025 [51]	88.5	52.9	87.1	3.7	1.4	**56.2**	**62.7**	**59.2**	87.2	89.0	**79.1**	**55.8**	87.9	**61.7**	**58.1**	**71.2**	62.6	**72.3**
	PatchMix(Ours)	**92.6**	**61.9**	88.0	36.3	**5.5**	46.7	**62.4**	52.6	**88.9**	87.9	**79.4**	**58.5**	**91.7**	**80.6**	**63.5**	60.7	**66.1**	**74.5**
DAFormer	DAFormerCVPR2022 [21]	84.5	40.7	88.4	41.5	6.5	50.0	55.0	54.6	86.0	89.8	73.2	48.2	87.2	53.2	53.9	61.7	60.9	67.4
	BLVCVPR2023 [64]	**86.7**	44.9	89.0	43.2	6.4	52.1	60.0	54.9	**88.2**	91.3	74.9	46.1	**88.6**	55.6	55.0	62.3	62.5	69.0
	MICCVPR2023 [40]	83.0	40.9	88.2	37.6	9.0	**52.4**	56.0	56.5	87.6	**93.4**	74.2	51.4	87.1	59.6	57.9	61.2	62.2	69.0
	CDACICCV2023 [53]	83.7	42.9	87.4	39.8	7.5	50.7	55.7	53.5	85.9	90.9	74.5	47.2	86.0	60.2	57.8	60.8	61.5	68.2
	RTeaICCV2023 [65]	85.9	43.2	**90.1**	45.1	6.3	52.4	**60.5**	**57.1**	87.8	92.2	**75.3**	51.8	87.4	55.9	54.1	62.6	63.0	69.5
	CAMixTCSVT2023 [66]	**87.4**	47.5	88.8	-	-	-	55.2	55.4	87.0	91.7	72.0	49.3	86.9	57.0	57.5	**63.6**	-	69.2
	MICDropECCV2024 [67]	81.0	37.1	**89.4**	**45.7**	**9.5**	51.8	57.3	**58.0**	86.7	85.0	73.6	50.4	88.2	64.7	56.8	**62.8**	62.4	68.5
	FDCNN2025 [41]	83.4	43.0	88.6	**47.0**	**12.1**	52.0	54.5	56.8	**88.2**	**92.8**	73.8	**52.5**	85.5	**67.5**	**59.3**	61.3	**63.6**	**69.8**
	PatchMix(Ours)	86.5	**48.4**	87.7	37.1	5.7	**56.7**	**60.7**	54.8	**88.9**	92.3	**76.5**	**52.0**	**89.4**	**67.0**	**58.7**	62.3	**64.0**	**71.2**
HRDA	HRDAECCV2022 [22]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	92.9	79.4	52.8	89.0	64.7	63.9	64.9	65.8	72.4
	BLVCVPR2023 [64]	87.6	47.9	**90.5**	50.4	6.9	57.1	64.3	**65.3**	86.9	93.4	78.9	54.9	89.1	62.9	65.2	**66.8**	66.8	73.4
	MICCVPR2023 [40]	86.6	50.5	89.3	47.9	7.8	59.4	**66.7**	63.4	87.1	**94.6**	81.0	58.9	90.1	61.9	67.1	64.3	67.3	74.0
	CDACICCV2023 [53]	**93.1**	**68.5**	89.8	51.2	8.9	59.4	65.5	65.3	84.7	94.4	**81.2**	57.0	90.5	56.9	66.8	66.4	**68.7**	**75.4**
	RTeaICCV2023 [65]	87.8	49.0	**90.3**	50.3	5.5	58.6	66.0	61.4	86.8	93.1	79.5	53.1	89.5	65.1	63.7	64.6	66.5	73.0
	MICDropECCV2024 [67]	82.8	42.6	**90.5**	**51.6**	**9.6**	**61.0**	65.7	**65.0**	**89.1**	**95.0**	**81.1**	**59.7**	**90.6**	**68.3**	**67.4**	66.5	67.9	74.2
	PDAKBS2024 [9]	90.2	52.1	88.3	50.5	**12.7**	54.1	65.2	61.4	86.2	90.8	80.4	55.2	89.5	65.1	64.1	**68.9**	67.1	73.6
	FDCNN2025 [41]	89.2	53.3	90.1	**54.2**	8.4	60.4	66.2	63.2	88.3	93.6	80.2	56.8	90.4	62.9	**68.0**	64.5	68.1	74.4
	PatchMix(Ours)	**91.1**	**57.0**	89.5	44.0	7.4	**62.5**	**70.0**	64.1	**90.7**	**94.6**	81.0	**61.0**	**94.7**	**81.9**	66.0	65.0	**70.1**	**77.4**

Table 3
mIoU(%) comparison of PatchMix when combined with different UDA strategies and backbone networks on GTA5 → Cityscapes. The listed approaches can be grouped into two categories: adversarial-training based methods ("AT") and self-training based methods ("ST"). Methods marked with † correspond to our reproduction using their public implementations.

	Paradigm	Network	w/o PatchMix	w/ PatchMix	Δ
AdaptSeg [14]	AT	DeepLab-V2 [2]	42.4	53.7	↑ **11.3**
ADVENT [61]	AT	DeepLab-V2 [2]	45.5	56.2	↑ **10.7**
DACS [37]	ST	DeepLab-V2 [2]	52.1	58.6	↑ **6.5**
DAFormer [21]	ST	DeepLab-V2 [2]	56.0	64.5	↑ **8.5**
HRDA [22]	ST	DeepLab-V2 [2]	63.0	67.4	↑ **4.4**
DAFormer [21]	ST	MiT-B5 [42]	68.3	72.2	↑ **3.9**
DAFormer† [21]	ST	Swin-B [43]	64.5	69.4	↑ **4.9**
HRDA [22]	ST	MiT-B5 [42]	73.8	76.6	↑ **2.8**
HRDA† [22]	ST	Swin-B [43]	68.4	74.6	↑ **6.2**

sensitive to a fixed patch grid. This remaining limitation motivates class-balanced confidence filtering and multi-scale or boundary-aware patch sampling.

4.2.3. Qualitative result

We compare qualitative segmentation results on GTA5 → Cityscapes for DAFormer [21], HRDA [22], and their PatchMix-enhanced variants, along with the ground-truth labels, in Fig. 3. The yellow dashed boxes highlight representative regions. In the first row, PatchMix more accurately segments fine structures such as "pole" and "traffic sign". In the fourth row, the boundaries of the "pole" category become smoother and more continuous with PatchMix. In the sixth row, PatchMix recovers more of the "fence" region. These selected examples illustrate qualitative improvements in both fine structures and larger regions; the mechanism-level conclusions are based on the controlled diagnostics in Section 4.3 rather than on individual images.

4.2.4. Generalization study

We apply PatchMix to a variety of UDASS methods, including adversarial and self-training approaches, and to backbone architectures based on DeepLab-V2 [2], MiT-B5 [42], and Swin-B [43], with paired gains reported in Table 3. PatchMix improves all listed paired baselines by 2.8–11.3 mIoU points. These results support the applicability of the same training branch across diverse UDASS pipelines and architectures, although the magnitude of the gain varies by baseline.

4.3. Ablation study

We now conduct a series of ablation experiments on the GTA5 → Cityscapes benchmark to analyze the contributions of different components in our framework.

4.3.1. Effectiveness of each component

We begin by examining the role of each component in PatchMix on GTA5 → Cityscapes. We progressively enable different parts of the design and report the results in Table 4 (Component).

In the "non-mixed augmented auxiliary input" row, the underlying DAFormer training is unchanged, but the auxiliary consistency branch receives an augmented target image without patch mixing, feature replacement, or mask tokens; this variant obtains 48.0 mIoU. Directly supervising the patch-mixed image with mixed labels gives 69.8 mIoU, 1.5 points above the 68.3 baseline. Replacing the second-image feature positions before decoding and using zero placeholders raises the result to 71.4. Resampling random vectors at these positions gives 70.3, whereas a learnable scale-wise token reaches 72.2. The progression distinguishes image mixing from the feature-replacement and semantic-reconstruction stages.

These controlled variants show that image mixing already strengthens local discrimination, while removing the second-image features before decoding adds a complementary context-reconstruction constraint. The learnable token then supplies a trainable representation of missing feature content rather than forcing the decoder to interpret a fixed placeholder. Together, the stages explain why the complete branch performs better than mixed-label consistency alone.

The controlled token comparison in Table 4 uses complete 40k runs under the same protocol. Zero and fixed-Gaussian tokens obtain 71.42 and 71.44 mIoU, respectively, while the learnable token obtains 72.18. Because a token is shared across all replaced positions at a scale and has neither a class index nor a position-specific parameter, it is best interpreted as an optimizable missing-feature embedding rather than an explicit class prototype. Its gradients aggregate reconstruction signals across target images, allowing a scale-dependent decoder input that is more flexible than a fixed zero or Gaussian vector.

4.3.2. In-depth analysis of local discrimination

To explicitly examine local discrimination on the target domain, we use a patch-based evaluation protocol. For each Cityscapes validation image, we preserve one 160 × 160 region and mask the surrounding pixels, producing a local-only input. Each trained model is applied to this image, and mIoU is computed only within the preserved region. Because the surrounding scene is unavailable, the model must rely on local appearance to classify the retained content. Repeating the procedure over all non-overlapping regions gives 57.8, 61.0, and 63.6 for DAFormer, MIC, and PatchMix, respectively, as reported in Table 4.

We then retain the same local region but replace its surroundings with another target image. This context-swapped input holds the local pixels fixed while introducing an unrelated surrounding scene. The decline relative to the local-only setting quantifies interference from context mismatch. Region mIoU is 52.9, 54.8, and 60.5 for DAFormer, MIC, and PatchMix. The larger PatchMix score indicates greater stability of local decisions under incorrect surrounding context, consistent with training on patch-mixed target images.

4.3.3. In-depth analysis of context utilization

We next analyze how PatchMix affects context utilization. Following MIC [40], we mask one 160 × 160 region while retaining the surrounding scene. The trained model must infer the semantics of the masked region from the remaining context, and mIoU is computed only within that region. We repeat the procedure over all non-overlapping regions of each Cityscapes validation image. DAFormer, MIC, and PatchMix obtain 18.8, 27.2, and 45.7, respectively, as reported in Table 4. This diagnostic directly tests whether the surrounding scene can support a prediction when local appearance is unavailable or unreliable.

We complement these mIoU diagnostics with a paired semantic-response analysis in Fig. 4, using the same DAFormer and PatchMix checkpoints as the 68.3 and 72.2 mIoU main results. With seed 7, we sample 100 Cityscapes validation images without replacement. For each image, a model-independent 160 × 160 region is selected by a fixed score combining rare-class occupancy, ground-truth boundary

Table 4
Ablation and diagnostic results on GTA5 → Cityscapes using DAFormer. All entries report mIoU (%).

Category	Setting	mIoU
Component	*Baseline*	68.3
	Non-mixed augmented auxiliary input	48.0
	Mixed input + mixed-label consistency	69.8
	+ Feature replacement (zero)	71.4
	+ Feature replacement (resampled random)	70.3
	+ Feature replacement (learnable token; full model)	**72.2**
Mask Token	Zero/Fixed Gaussian/Learnable	71.42/71.44/**72.18**
Local Discrimination	Local-only: DAFormer/MIC/PatchMix	57.8/61.0/**63.6**
	Context-swapped: DAFormer/MIC/PatchMix	52.9/54.8/**60.5**
Context Utilization	Masked-local: DAFormer/MIC/PatchMix	18.8/27.2/**45.7**
$\mathcal{L}_M$ Region	On Masked Patch/Unmasked Patch/All Patches	71.2/71.6/**72.2**
Mixing Domain	Source + Target/Target + Target	70.9/**72.2**
Patch Size	32/64/128/320	**72.2**/71.6/71.8/64.8
Mix Ratio	0.3/0.5/0.7	70.7/**72.2**/69.9
Reconstruction	Dual-target/Single-target	71.8/**72.2**

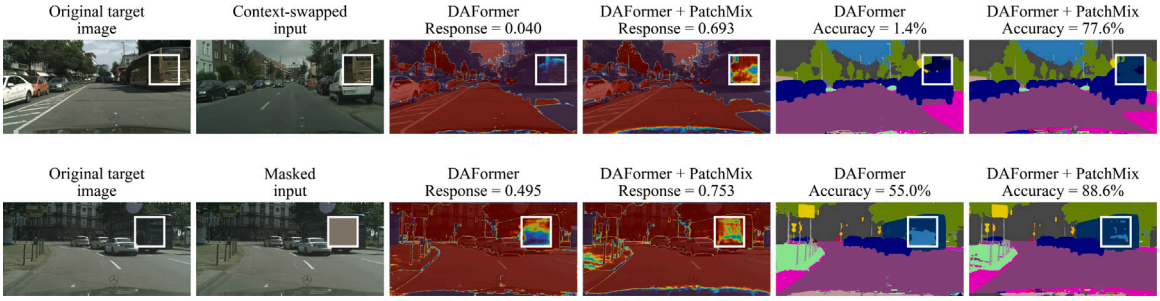


Fig. 4. Mechanism-aligned Cityscapes visualization using the checkpoints reported in the main DAFormer comparison. Top: a 160 × 160 local region is preserved while the surrounding context is replaced. Bottom: the region is masked while the surrounding scene is retained. Heatmaps show the softmax probability of the corresponding ground-truth class after applying each model's own architecturally identical, trained 1 × 1 semantic head. White boxes mark the controlled region, and all heatmaps share the range $[0, 1]$. Values above the displayed examples are per-image measurements; paired statistics over the fixed, output-independent 100-image sample are reported in the text.

density, and valid-pixel coverage; the donor image is the validation image at a fixed index offset of 137. Each model's trained 1 × 1 head converts the fused pre-classification decoder feature to logits, followed by softmax. The regional semantic response is the mean probability assigned to the ground-truth class over valid pixels in the controlled region. Confidence intervals are percentile intervals from 10,000 paired bootstrap resamples, using a fixed bootstrap seed of 20260814.

Under context swapping, PatchMix increases the mean response from 0.750 ± 0.186 to 0.786 ± 0.138 and region pixel accuracy from $76.0\% \pm 19.0\%$ to $80.9\% \pm 14.3\%$. The paired mean changes are +0.036 (bootstrap 95% CI $[0.014, 0.060]$) and +4.9 percentage points (95% CI $[2.5, 7.5]$), respectively. When the local region is masked, the mean response changes from 0.263 ± 0.186 to 0.281 ± 0.158 and accuracy from $26.5\% \pm 21.4\%$ to $29.0\% \pm 18.1\%$. The corresponding paired CIs, $[-0.002, 0.038]$ and $[-0.1, 4.9]$ percentage points, include zero; these values are therefore treated as a descriptive contextual trend, while the full-region mIoU in Table 4 remains the primary contextual diagnostic. All heatmaps share one scale, and the displayed cases are illustrative rather than the basis of the aggregate statistics.

4.3.4. Effect of $\mathcal{L}_M$ on specific patches

We further investigate how the reconstruction loss $\mathcal{L}_M$ affects different regions of the image. Supervision only on replaced regions gives 71.2 mIoU, supervision only on retained regions gives 71.6, and supervision on all regions gives 72.2, as reported in Table 4. The last setting is 0.6 points above retained-only supervision, indicating that both observed and reconstructed regions contribute to semantic consistency and help the model maintain coherent predictions across the full image.

4.3.5. Effect of mixing images from different domains

We evaluate whether patch-level mixing should occur within the target domain or across domains. As shown in Table 4 (Mixing Domain), mixing a target image with a source image leads to a 1.3-point drop in mIoU compared with mixing two target images. This result is consistent with cross-domain mixing introducing visual cues from the substantial GTA5–Cityscapes appearance gap, which may interfere with learning target-specific representations.

4.3.6. Effect of patch size

We evaluate PatchMix with different patch sizes in Table 4 (Patch Size). Sizes between 32 and 128 improve performance, with 32 yielding the best result under this protocol. Smaller patches may preserve more local detail, whereas very large patches (e.g., 320) reduce the spatial diversity of mixing and may limit fine-grained supervision.

4.3.7. Effect of mix ratios

We analyze the proportion of patches drawn from the first image while fixing the patch size to 32. As summarized in Table 4 (Mix Ratio), a balanced ratio of 0.5 yields the strongest performance, whereas ratios of 0.3 and 0.7 perform less well. This pattern is consistent with a trade-off between perturbation strength and retained local information. We therefore use 0.5 as the default configuration.

4.3.8. Effect of reconstruction strategy

We compare different reconstruction strategies in Table 4 (Reconstruction). Reconstructing only the first image from the replaced features yields 72.2 mIoU, compared with 71.8 for dual-target reconstruction. Attempting to reconstruct both images imposes two semantic

Table 5
Ablation of image-level filling content. Mean, Black, White, and Random use the image mean, zeros, ones, and uniform noise, respectively. Zoom-in uses patches from a randomly cropped enlarged view, Shuffled permutes patches within the image, and Mix uses patches from another target image.

Patch Padding	Mean	Black	White	Random	Zoom-in	Shuffled	Mix
Example							
mIoU	71.4	70.22	71.2	71.2	70.3	70.8	72.2

Table 6
Matched efficiency comparison with MiT-B5 and 640 × 640 inputs. Training uses four RTX 3090 GPUs and global batch size 4 (10 warm-up and 100 measured iterations); inference uses one RTX 3090 and batch size 1 (50 warm-up and 200 measured passes). Times are mean ± standard deviation with GPU synchronization.

Method	Training time (s/iter.)	Peak memory (GiB/GPU)	Test latency (ms/image)	Training-time change
DAFormer	0.862 ± 0.232	11.43	51.882 ± 2.780	Baseline
DAFormer + PatchMix	1.067 ± 0.132	11.43	51.497 ± 1.909	+23.9%

objectives on the same mixed representation and requires the decoder to resolve overlapping content under mismatched contexts. Under this protocol, the single-target objective performs better and also matches the feature-replacement design in Eq. (5).

4.3.9. Effect of filling content

We compare Mean, Black, White, Random, Zoom-in, Shuffled, and Mixed filling in Table 5. Patches from another target image retain realistic target-domain texture and give the best result. They can introduce boundary discontinuities, but these serve as encoder perturbations: before decoding, Eq. (5) replaces the corresponding feature positions, so the second image does not introduce a second semantic target. Zoom-in changes scale, while Shuffled disrupts within-image structure.

4.3.10. Computational efficiency

We separately measure computational efficiency under a matched DAFormer setting on four RTX 3090 GPUs, using the same MiT-B5 backbone, 640 × 640 crop, global batch size of 4, optimizer, data, and software environment for both rows. Training statistics use 10 warm-up iterations followed by 100 synchronized iterations; inference uses batch size 1 on one RTX 3090, with 50 warm-up passes followed by 200 synchronized passes. As shown in Table 6, PatchMix increases training time by 23.9%, mainly because of target pairing and the semantic-reconstruction branch, while the observed peak allocation remains 11.43 GiB per GPU. The measured test-latency difference is smaller than the timing standard deviations, consistent with the unchanged inference architecture described in Section 4.

4.3.11. Effect of reconstructing the original image

We replace semantic reconstruction with pixel-level RGB reconstruction and observe a 2.0-point mIoU decrease. Under synthetic-to-real shift, color, illumination, rendering texture, and sensor characteristics can change while the semantic class remains fixed. RGB reconstruction rewards recovery of these domain-specific appearance details and can allocate capacity to variations that do not support cross-domain classification. In contrast, semantic reconstruction matches the student's perturbed prediction to the teacher decision on the complete target image and directly regularizes the task output. This alignment between reconstruction target and adaptation objective explains why semantic reconstruction is more suitable here; it does not imply that RGB reconstruction is generally ineffective for representation pretraining [49].

5. Limitations

PatchMix has several limitations. It inherits errors from teacher pseudo labels, which can weaken supervision for rare or initially misclassified classes. Random target pairing can create semantically implausible contexts, while a fixed single-scale patch grid may split thin structures and make performance sensitive to the perturbation configuration. The auxiliary reconstruction branch also adds training computation, although it is removed for inference. These limitations motivate confidence- and semantic-aware pairing, class-balanced or uncertainty-weighted supervision, and boundary-adaptive, multi-scale patch construction.

6. Conclusion

In this work, we presented PatchMix, a training-time regularization formulation for unsupervised domain adaptation in semantic segmentation that couples target–target patch mixing with feature replacement and semantic reconstruction. The student encoder processes a mixed target image, after which features originating from the second image are replaced by scale-wise learnable tokens; guided by an EMA teacher, the decoder reconstructs the semantic prediction of the complete first image. This mix–encode–replace–semantic-reconstruct process jointly challenges local recognition under mismatched context and the recovery of missing semantics from retained evidence. Experiments across synthetic-to-real benchmarks, CNN and Transformer backbones, controlled ablations, and mechanism-oriented diagnostics show that PatchMix improves local discrimination and context utilization. Because the auxiliary branch is used only during training, the inference architecture remains unchanged.

CRediT authorship contribution statement

Meiqin Liu: Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Linfeng He:** Writing – review & editing, Writing – original draft, Visualization, Validation, Conceptualization. **Zilin Wang:** Writing – review & editing, Writing – original draft, Visualization, Validation. **Chao Yao:** Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Yunchao Wei:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All datasets used in this study (GTA5, SYNTHIA, and Cityscapes) are publicly available from their original sources.

References

[1] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.

[3] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Proceedings of the European Conference on Computer Vision, 2020.

[7] F. Lin, Z. Liang, S. Wu, J. He, K. Chen, S. Tian, StructToken: Rethinking semantic segmentation with structural prior, IEEE Trans. Circuits Syst. Video Technol. 33 (10) (2023) 5655–5663.

[8] Y. Zhang, M. Ye, Y. Gan, W. Zhang, Knowledge based domain adaptation for semantic segmentation, Knowl.-Based Syst. 193 (2020) 105444.

[9] M. Liao, S. Tian, Y. Zhang, G. Hua, W. Zou, X. Li, PDA: Progressive domain adaptation for semantic segmentation, Knowl.-Based Syst. 284 (2024) 111179.

[10] M. Liao, J. Wang, H. Deng, Y. Peng, H. Yin, Y. Wang, G. Hua, Domain-generalized token linking in vision foundation models for semantic segmentation, Knowl.-Based Syst. (2025) 114497.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.

[12] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for data: Ground truth from computer games, in: Proceedings of the European Conference on Computer Vision, 2016.

[13] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, A.M. Lopez, The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3234–3243.

[14] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, M. Chandraker, Learning to adapt structured output space for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7472–7481.

[15] A. Dundar, M.-Y. Liu, Z. Yu, T.-C. Wang, J. Zedlewski, J. Kautz, Domain stylization: A fast covariance matching framework towards domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 43 (7) (2020) 2360–2372.

[16] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, T. Darrell, CyCADA: Cycle-consistent adversarial domain adaptation, in: Proceedings of the International Conference on Machine Learning, 2018, pp. 1989–1998.

[17] Y. Yang, S. Soatto, FDA: Fourier domain adaptation for semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 4085–4095.

[18] Y. Tian, S. Zhu, Partial domain adaptation on semantic segmentation, IEEE Trans. Circuits Syst. Video Technol. 32 (6) (2022) 3798–3809.

[19] L. Melas-Kyriazi, A.K. Manrai, PixMatch: Unsupervised domain adaptation via pixelwise consistency training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12435–12445.

[20] R. Li, S. Li, C. He, Y. Zhang, X. Jia, L. Zhang, Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11593–11603.

[21] L. Hoyer, D. Dai, L. Van Gool, DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9924–9935.

[22] L. Hoyer, D. Dai, L. Van Gool, HRDA: Context-aware high-resolution domain-adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Vision, 2022.

[23] F. Shen, A. Gurram, Z. Liu, H. Wang, A. Knoll, DiGA: Distil to generalize and then adapt for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15866–15877.

[24] C. Corbiere, N. Thome, A. Saporta, T.-H. Vu, M. Cord, P. Perez, Confidence estimation via auxiliary models, IEEE Trans. Pattern Anal. Mach. Intell. 44 (10) (2021) 6043–6055.

[25] Y. Zou, Z. Yu, X. Liu, B. Kumar, J. Wang, Confidence regularized self-training, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[26] Y. Wang, J. Peng, Z. Zhang, Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2021.

[27] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, J. Huang, Progressive feature alignment for unsupervised domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 627–636.

[28] F. Pan, I. Shin, F. Rameau, S. Lee, I.S. Kweon, Unsupervised intra-domain adaptation for semantic segmentation through self-supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3764–3773.

[29] M. Kim, H. Byun, Learning texture invariant representation for domain adaptation of semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12975–12984.

[30] P. Zhang, B. Zhang, T. Zhang, D. Chen, Y. Wang, F. Wen, Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12414–12424.

[31] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

[32] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, H. Hu, SimMIM: A simple framework for masked image modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.

[33] B. Cheng, A. Schwing, A. Kirillov, Per-pixel classification is not all you need for semantic segmentation, in: Proceedings of the Advances in Neural Information Processing Systems, 2021.

[34] B. Cheng, I. Misra, A.G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1290–1299.

[35] H. Zhang, M. Cisse, Y. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: Proceedings of the International Conference on Learning Representations, 2018.

[36] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, CutMix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[37] W. Tranheden, V. Olsson, J. Pinto, L. Svensson, DACS: Domain adaptation via cross-domain mixed sampling, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1379–1389.

[38] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, C.A. Raffel, MixMatch: A holistic approach to semi-supervised learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2019.

[39] V. Olsson, W. Tranheden, J. Pinto, L. Svensson, ClassMix: Segmentation-based data augmentation for semi-supervised learning, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1369–1378.

[40] L. Hoyer, D. Dai, H. Wang, L. Van Gool, MIC: Masked image consistency for context-enhanced domain adaptation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11721–11732.

[41] C. Rao, W. Fan, C. Wang, X. Yang, X. Wei, D. Zhou, FDC: Feature dropout consistency for unsupervised domain adaptation semantic segmentation, Neural Netw. 189 (2025) 107527.

[42] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and efficient design for semantic segmentation with transformers, in: Proceedings of the Advances in Neural Information Processing Systems, 2021.

[43] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE International Conference on Computer Vision, 2021.

[44] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[45] Y. Luo, L. Zheng, T. Guan, J. Yu, Y. Yang, Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019.

[46] K. Mei, C. Zhu, J. Zou, S. Zhang, Instance adaptive self-training for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision, 2020.

[47] R. Chen, Y. Rong, S. Guo, J. Han, F. Sun, T. Xu, W. Huang, Smoothing matters: Momentum transformer for domain adaptive semantic segmentation, 2022, arXiv preprint arXiv:2203.07988.

[48] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[49] J. Liu, X. Huang, J. Zheng, Y. Liu, H. Li, MixMAE: Mixed and masked autoencoder for efficient pretraining of hierarchical vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6252–6261.

[50] W. Dong, Z. Liang, L. Wang, G. Tian, Q. Long, Unsupervised domain adaptation segmentation algorithm with cross-domain data augmentation and category contrast, Neurocomputing 623 (2025) 129393.
[51] Z. Zhang, G. Wu, J. Zhang, X. Zhu, D. Tao, T. Chai, Unified domain adaptive semantic segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 47 (8) (2025) 6731–6748.
[52] N. Araslanov, S. Roth, Self-supervised augmentation consistency for adapting semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15384–15394.
[53] K. Wang, D. Kim, R. Feris, M. Betke, CDAC: Cross-domain attention consistency in transformer for domain adaptive semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2023, pp. 11519–11529.
[54] Y. Duan, J. Zhang, L. Xue, Y. Xu, Embedded RF fingerprint interpretation: Multi-channel complex residual networks with adaptive sphere space decision boundaries, J. Syst. Eng. Electron. 37 (1) (2026) 137–147.
[55] J. Zhang, Y. Liu, G. Ding, B. Tang, Y. Chen, Adaptive decomposition and extraction network of individual fingerprint features for specific emitter identification, IEEE Trans. Inf. Forensics Secur. 19 (2024) 8515–8528.
[56] J. Zhang, S. Huang, J. Liu, X. Zhu, F. Xu, PYRF-PCR: A robust three-stage 3D point cloud registration for outdoor scene, IEEE Trans. Intell. Veh. 9 (1) (2024) 1270–1281.
[57] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Proceedings of the Advances in Neural Information Processing Systems, 2017.
[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[59] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017, arXiv preprint arXiv:1711.05101.
[60] Z. Jiang, Y. Li, C. Yang, P. Gao, Y. Wang, Y. Tai, C. Wang, Prototypical contrast adaptation for domain adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Vision, 2022, pp. 36–54.
[61] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.
[62] G. Kang, Y. Wei, Y. Yang, Y. Zhuang, A. Hauptmann, Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation, in: Proceedings of the Advances in Neural Information Processing Systems, 2020.
[63] Y. Cao, H. Zhang, X. Lu, Y. Chen, Z. Xiao, Y. Wang, Adaptive refining-aggregation-separation framework for unsupervised domain adaptation semantic segmentation, IEEE Trans. Circuits Syst. Video Technol. 33 (8) (2023) 3822–3832.
[64] Y. Wang, J. Fei, H. Wang, W. Li, T. Bao, L. Wu, R. Zhao, Y. Shen, Balancing logit variation for long-tailed semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19561–19573.
[65] D. Zhao, S. Wang, Q. Zang, D. Quan, X. Ye, R. Yang, L. Jiao, Learning pseudo-relations for cross-domain semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2023, pp. 19191–19203.
[66] Q. Zhou, Z. Feng, Q. Gu, J. Pang, G. Cheng, X. Lu, J. Shi, L. Ma, Context-aware mixup for domain adaptive semantic segmentation, IEEE Trans. Circuits Syst. Video Technol. 33 (2) (2023) 804–817.
[67] L. Yang, L. Hoyer, M. Weber, T. Fischer, D. Dai, L. Leal-Taixé, M. Pollefeys, D. Cremers, L. Van Gool, MICDrop: Masking image and depth features via complementary dropout for domain-adaptive semantic segmentation, in: Proceedings of the European Conference on Computer Vision, 2024, pp. 329–346.