TextAdapter: Self-Supervised Domain Adaptation for Cross-Domain Text Recognition

Xiao-Qian Liu, Peng-Fei Zhang, Xin Luo, Zi Huang, *Senior Member, IEEE*, and Xin-Shun Xu, *Senior Member, IEEE*

***Abstract*—Text recognition remains challenging, primarily due to the scarcity of annotated real data or the hard labor to annotate large-scale real data. Most existing solutions rely on synthetic training data, where the synthetic-to-real domain gaps limit the model performance on real data. Unsupervised domain adaptation (UDA) methods have been proposed, aiming to obtain domain-invariant representations. However, they commonly focus on domain-level alignment, neglecting the fine-grained character features and thus leading to indistinguishable characters. In this paper, we propose a simple yet effective self-supervised UDA framework tailored for cross-domain text recognition, named TextAdapter, which integrates contrastive learning and consistency regularization to mitigate domain gaps. Specifically, a fine-grained feature alignment module based on character contrastive learning is designed to learn domain-invariant character representations by category-level alignment. Additionally, to address the task-agnostic problem in contrastive learning, i.e., ignoring the sequence semantics, an instance consistency matching module is proposed to perceive the contextual semantics by matching the prediction consistency among target data different augmented views. Experimental results on cross-domain benchmarks demonstrate the effectiveness of our method. Furthermore, TextAdapter can be embedded in most off-the-shelf text recognition models with new state-of-the-art performance, which illustrates the generality of our framework.**

***Index Terms*—Self-supervised learning, contrastive learning, consistency regularization, domain adaptation, text recognition.**

I. Introduction

TEXT recognition aims to read text from a cropped image, which involves vision and language modeling [1], [2]. Due to the scarcity of annotated real text or the hard labour to annotate large-scale real text, current text recognition methods heavily rely on synthetic training text [3], [4]. Nevertheless, domain gaps between synthetic and real text are significant, which limit the model performance on the real text. Therefore, it is essential to mitigate the domain discrepancy to improve the recognition performance on real text.

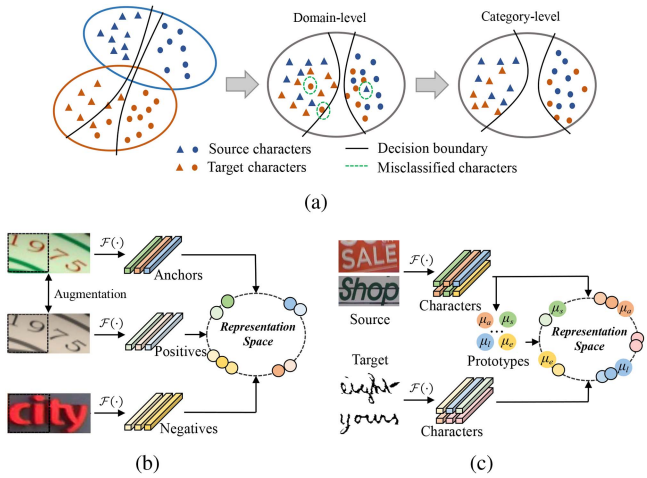


Fig. 1. (a) Adaptation of domain-level and category-level, where the green circles indicate misclassified characters. (b) Contrastive learning of previous methods, which builds positive and negative cases by a sliding window. (c) Contrastive learning of TextAdapter, which is performed on the prototypes computed by pseudo-labeled characters.

To alleviate the synthetic-to-real gaps, some researchers use additional and diverse unlabeled real data to improve performance by semi-supervised learning [5], [6]. Similarly, some regard it as UDA tasks [7], [8] by transferring knowledge learned on labeled source data to unlabeled target data. Although the domain gaps may be partly mitigated, problems still need to be further considered. It is inappropriate to consider a domain as a whole, ignoring that the minimal unit of a text is a character, and character recognition is the prerequisite for sequence prediction. For example, ASSDA [9] utilizes global and local (character) features to mitigate domain gaps based on adversarial learning. Since these methods treat all the source or target domain characters as a whole and no longer distinguish character classes within a domain, this way is called domain-level adaptation. The learned representation is, therefore, domain-level domain-invariant. Consequently, due to not considering character category information, this representation may result in indistinguishable characters, as shown by the green circles in the *Domain-level* in Fig. 1(a). In contrast, category-level adaptation aligns similar characters across domains, where character class information makes the learned representations not only domain-invariant but also category-distinguishable. We refer to

Manuscript received 17 July 2023; revised 18 February 2024; accepted 4 May 2024. Date of publication 13 May 2024; date of current version 18 September 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62172256, Grant 62202278, and Grant 62202272 and in part by the Natural Science Foundation of Shandong Province undet Grant ZR2019ZD06 and Grant ZR2020QF036. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ling-Yu Duan. *(Corresponding author: Xin-Shun Xu.)*

Xiao-Qian Liu, Xin Luo, and Xin-Shun Xu are with the School of Software, Shandong University, Jinan 250101, China (e-mail: xuxinshun@sdu.edu.cn).

Peng-Fei Zhang and Zi Huang are with the School of Electrical Engineering and Computer Science, University of Queensland, Brisbane, QLD 4072, Australia.

Digital Object Identifier 10.1109/TMM.2024.3400669

this as character-level fine-grained domain-invariant representations. Therefore, proposing a specific UDA-based text recognition method that enables fine-grained domain-invariant representations is much more valuable.

Owing to the superiority of self-supervised learning, contrastive learning-based text recognition methods have been proposed [5], [10], [11]. Actually, the goal of contrastive learning of maximizing the similarity between the anchor and its positives while contrasting the negatives agrees with the objective of UDA. These contrastive methods split an image into multiple subwords using a sliding window. As shown in Fig. 1(b), the different augmentations of the input serve as the anchor and positives, respectively, while any subword of other images can be a negative case. This subword-based contrastive learning has two limitations. First, positive cases from the same region of different augmentations align the visual information but may corrupt the sequence semantics, as a sliding window may slice a complete character. Second, a wide-range selection of negative cases may lead to misalignment, as other images containing the same text are still seen as negative cases even though they are semantically identical.

Inspired by the above observation, we present a simple yet effective self-supervised UDA framework for cross-domain text recognition, termed TextAdapter, which extracts fine-grained domain-invariant character representations to mitigate domain gaps. Firstly, a fine-grained feature alignment module is proposed, which implements character-based contrastive learning by exploiting the weakly supervised information of character classes. Specifically, we first apply pseudo-labeling to unlabeled target samples to obtain pseudo-labeled characters. Then, character-based contrastive learning is performed by character prototypes computed from pseudo-labeled characters to align the feature distribution of the different domains at the category level. As illustrated in Fig. 1(c), characters from the same category in the source and target domains should be close to the prototype while staying away from other category prototypes. In other words, the prototype of that category is a positive case, while different category prototypes are negative cases. The main difference between our character-based contrastive learning and subword-based contrastive learning is that our method takes characters, not words or subwords, as anchors, thus maintaining sequence semantics and avoiding uncontrollable negative cases.

While character-based contrastive learning is effective for fine-grained domain-invariant character representations, it ignores the contextual semantics of the text sequence. To address this, we further propose an instance consistency matching module that leverages the high-level information of character prediction logits to perceive sequence semantics effectively. To achieve this, we preserve model perception ability by matching the temporal prediction consistency among different augmented views of target data. Additionally, to improve the sample utilization and matching quality, we design a confidence-based sample selection strategy, where only high-confidence instances are selected for matching.

To summarize, the main contributions are as follows:

- We propose a simple yet effective self-supervised adaptation framework dubbed TextAdapter for cross-domain text recognition. To our knowledge, we are the first to introduce character-level fine-grained domain-invariant representations in UDA-based text recognition tasks.
- Considering the sequential nature of the text, we specially design a fine-grained feature alignment module and an instance consistency matching module to perform category-level adaptation and perceive the target semantics.
- We deploy TextAdapter to off-the-shelf text recognition models and conduct experiments on nine widely used benchmarks. The results show that TextAdapter can further improve the performance of these models and achieve new state-of-the-art (SOTA) results, highlighting the effectiveness and generality of our method.

II. Related Work

In this section, we review the literature on deep learning-based text recognition, unsupervised domain adaptation, domain adaptation for text recognition, and self-supervised text recognition.

A. Deep Learning-Based Text Recognition

Deep learning-based text recognition methods [12], [13] could be categorized into three types according to the decoders, i.e., CTC, attention-RNN, and transformer decoders. CRNN [14] is a representative CTC decoder method. However, CTC decoder methods [15], [16] have difficulty recognizing irregular text. Therefore, attention-RNN decoder methods [17], [18] have become popular due to their ability to localize text in images accurately. For instance, SAR [19] employs 2D attention to handle the complicated layout of irregular text. ASTER [20] utilizes a spatial transform network and an attention decoder for irregular text recognition. Cheng et al. [21] focused on the attention drift problem to improve recognition performance. Recently, some transformer decoder methods [22], [23] have also been proposed, such as ABINet [24], which explicitly models linguistic rules in scene text recognition (STR). However, these methods overlook the domain gaps between synthetic and real text.

B. Unsupervised Domain Adaptation

Most UDA methods [25], [26] can be broadly classified into domain-level and category-level approaches. Domain-level methods [27], [28] aims to reduce distribution differences between the entire source and target domains by pulling them towards the same distribution. Maximum mean discrepancy (MMD) [29], [30] and correlation alignment (CORAL) [31], [32] are commonly used divergence measures. Category-level UDA methods focus on aligning the distributions of each category within the domain rather than the entire domain. This is achieved by pushing the target samples to the distribution of source samples in each category. For example, Li et al. [33] achieved category-level alignment through an adversarial manner between the feature generator and domain-specific discriminator. Category-level adaptation can extract more accurate and discriminative representations in the label space than domain-level adaptation.

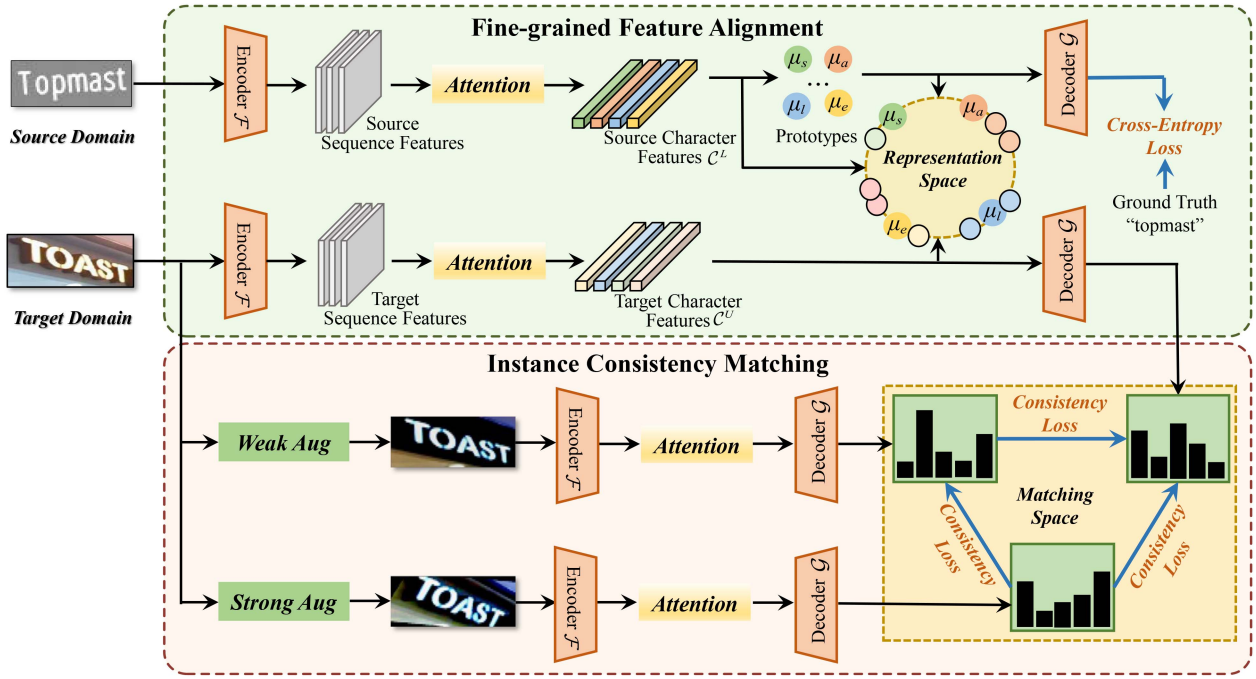


Fig. 2. The pipeline of TextAdapter. The *Fine-grained Feature Alignment* module aligns character features of source and target domains in representation space. The *Instance Consistency Matching* module aligns the temporal prediction of target images and their different augmented views in the matching space. All encoders and decoders share the weights separately.

C. Domain Adaptation for Text Recognition

Domain adaptation for text recognition can be classified into two types. The first is the adaptation of different scenes [7], [8], [9], where each scene is regarded as a domain. Thus, there are typically three domains: synthetic text, real scene text, and handwritten text. For example, Zhang et al. [9] proposed a seq2seq text recognition UDA method that uses global and local features to extract domain-invariant representations. However, it implements domain-level adaptation, and the domain-invariant representations may not accurately distinguish characters within the domain. SMILE [7] optimizes the labeled source and unlabeled target data respectively. Since the feature distribution is not explicitly aligned, it may perform poorly when the domain divergence is apparent. The second type is the adaptation of writing styles between writers in the handwritten text recognition (HTR) [34], [35], where each writer is treated as a domain. Since there is variability in the writing styles of different writers, a generalized representation of multiple writers can improve the recognition performance of new writers. For instance, metaHTR [34] treats styles adaptation as domain generalization, which learns suitable initialization parameters on multiple authors in a meta-learning manner with only a few gradient updates on new authors to improve performance. However, this approach requires the test data to be visible during training, which only satisfies bare practical situations. This paper focuses on the adaptation of different scenes.

D. Self-Supervised Text Recognition

More recently, self-supervised methods have been proposed for STR, *e.g.*, contrastive learning-based methods [11], [36] and consistency regularization (CR)-based ones. For example, SeqCLR [10] first applies contrastive learning from non-sequence tasks to sequence recognition tasks. PerSec [11] designs a hierarchical contrastive learning framework, which can simultaneously learn latent representations from low-level stroke and high-level semantic contextual spaces. DiG [5] learns discrimination and generation by integrating contrastive learning and masked image modeling. ConCLR [36] improves the performance of out-of-vocabulary text by alleviating the over-reliance on context. In contrast, CR-based methods assume the model should produce consistent predictions when fed perturbed versions of the same image [37]. For instance, Zheng et al. [6] proposed a CR-based framework that addresses character misalignment. SemiMTR [38] is a multimodal text recognizer fine-tuned via a sequential, character-level, and CR between weak and strong augmented views. However, these self-supervised methods require abundant and diverse unlabeled real scene text.

III. Our Method

As shown in Fig. 2, our framework consists of a fine-grained feature alignment module and an instance consistency matching module. The proposed method takes the attention-RNN decoder model structure as an example, and it can be deployed to most STR models based on attention-RNN and transformer decoders.

In particular, we pseudo-label characters by measuring model confidence based on character features from the attention module. Then, character-based contrastive learning is performed on pseudo-labeled character features from different domains. At the same time, the instance consistency matching module is

constructed to align instances of different augmented views to encourage model perception ability.

In the following sections, we introduce the notations of text recognition UDA task and the baseline model. Then, the fine-grained feature alignment and instance consistency matching modules are depicted, respectively. Finally, we summarize the overall loss.

A. Problem Formulation

The text recognition UDA task seeks a recognizer for a target domain when given labeled source data $\mathcal{D}^L = \{(x_i^l, y_i^l)\}_{i=1}^{N^l}$ as well as unlabeled target data $\mathcal{D}^U = \{(x_i^u)\}_{i=1}^{N^u}$, where N^l and N^u are the numbers of images, respectively. Since the supervised information of the source domain is word-level without character-level annotations, the label y^l is defined as $y^l = \{y_1^l, y_t^l, \ldots, y_T^l\}$, where T is the pre-defined decoding length. Our task aims to learn a model that performs well on target data $\mathcal{D}^U$.

B. Baseline Text Recognition Model

The attention-RNN decoder-based baseline model contains an encoder $\mathcal{F}$, an attentional block, and an RNN decoder $\mathcal{G}$. For an input x, the encoder $\mathcal{F}$ firstly extracts sequence features $\mathcal{F}(x) = [f_1, f_2, \ldots, f_T] \in \mathbb{R}^{T\times D}$, where D is the feature dimension. Then, a sequence-to-sequence attention mechanism is introduced to locate the specific character features in $\mathcal{F}(x)$. At time t, the representation in $\mathcal{F}(x)$ most relevant to the character y_t is defined as g_t,

$$g_t = \sum_{i=1}^{T} \alpha_{t,i} f_i \,, \tag{1}$$

where f_i is the i-th subregion of features $\mathcal{F}$, and $\alpha_{t,i} \in (0, 1)$ is the attention weight,

$$\alpha_{t,i} = \frac{\exp\left(e_{t,i}\right)}{\sum_{j=1}^{T} \exp\ \left(e_{t,j}\right)} \,, \tag{2}$$

$$e_{t,i} = \omega^T \tanh\left(\mathcal{W}_s s_{t-1} + \mathcal{W}_f \mathcal{F} + b\right), \tag{3}$$

where ω, $\mathcal{W}_s$, $\mathcal{W}_f$, and b are trainable parameters, and s_{t-1} is the hidden state of RNN at time $t-1$.

Next, the RNN decoder $\mathcal{G}$ updates the hidden state s_t,

$$(o_t, s_t) = \mathcal{G}\left(s_{t-1}; [E(y_{t-1}), g_t]\right), \tag{4}$$

where $E(\cdot)$ is the character embedding layer.

Then, the decoder computes the output probability of the predicted character y_t via a linear layer and a *softmax* function,

$$p(y_t|x) = softmax\left(\mathcal{W}_o s_t + b_o\right), \tag{5}$$

where $\mathcal{W}_o$ and b_o are trainable parameters.

Lastly, the baseline model is trained using only the labeled source data with a standard cross-entropy loss,

$$\mathcal{L}_{CE} = \frac{1}{T}\sum_{t=1}^{T} -\log p(y_t^l|x^l)\,, \tag{6}$$

where $p(y_t^l|x^l)$ is the predicted probability of the output being y_t^l after the *softmax*, and T is the pre-defined decoding length.

C. Fine-Grained Feature Alignment

The representation learned by domain-level adaptation is globally domain-invariant for the text recognition UDA task since all source or target domain characters are treated as a whole. Since the class information of fine-grained characters is not considered, it may lead to the indistinguishability of characters. We start from the text recognition task and implement feature alignment at the fine-grained category level, i.e., character category-level adaptation, to extract fine-grained domain-invariant representations.

As mentioned, contrastive learning among multiple subwords divided by a sliding window has two limitations. (1) *Slicing characters:* Due to the immutability of image character gaps, a fixed sliding window may slice a complete character and thus corrupt the sequence semantics of this character, as shown in Fig. 3(a). (2) *Semantic misalignment:* Since subwords of any other image can be used as negative examples, it leads to misalignment of semantics when other subwords contain the same character content as the anchor. For example, in Fig. 3(b), the subword containing 'ON' is a false negative case such that contrastive learning optimization leads to semantic ambiguity, thus resulting in misalignment. Therefore, we propose character-based contrastive learning on the text recognition UDA task to achieve character category-level adaptation.

1) Pseudo-Labeling for Fine-Grained Characters: Category-level adaptation is based on fine-grained character features, so the primary problem is to extract such features accurately. Therefore, pseudo-labeling is introduced into the unlabeled target data, leveraging the prediction capability of a pre-trained model to generate pseudo-labels for the unlabeled text sequences. Specifically, a pre-trained model is first warmed up with the labeled source data, after which an image $x^u \in \mathcal{D}^U$ is fed to the model to obtain the pseudo-label $\tilde{y}^u = \{\tilde{y}_1^u, \tilde{y}_t^u, \ldots, \tilde{y}_T^u\}$.

Fine-grained character features are defined as the context vector g_t in (1), denoted by $c = g_t$, following the approaches in [9], [36]. Since the source labels and target pseudo-labels are word-level without explicit character-level annotations, the character features are pseudo-labeled with some uncertainty. To obtain further accurate character features, a feature filter threshold η is introduced. The intuition is that if the current character feature is distinguishable, the probability that it belongs to a specific character is as high as possible and higher than those of other characters. In detail, for the label $y^l = \{y_1^l, y_t^l, \ldots, y_T^l\}$ of x^l and the pseudo-label $\tilde{y}^u = \{\tilde{y}_1^u, \tilde{y}_t^u, \ldots, \tilde{y}_T^u\}$ of x^u, the pseudo-labels of the source character feature c^l and the target character feature c^u are y_t^l and $\tilde{y}_t^u$, respectively. Unifying the pseudo-labels into $z^l = y_t^l$ and $z^u = \tilde{y}_t^u$, the fine-grained character features of the source and target domains after filtering stored in the character pool are defined as follows,

$$\mathcal{C}^L = \{(c_i^l, z_i^l)|p(z_i^l) \geq \eta\}_{i=1}^{M^l},\ \mathcal{C}^U = \{(c_i^u, z_i^u)|p(z_i^u) \geq \eta\}_{i=1}^{M^u}, \tag{7}$$

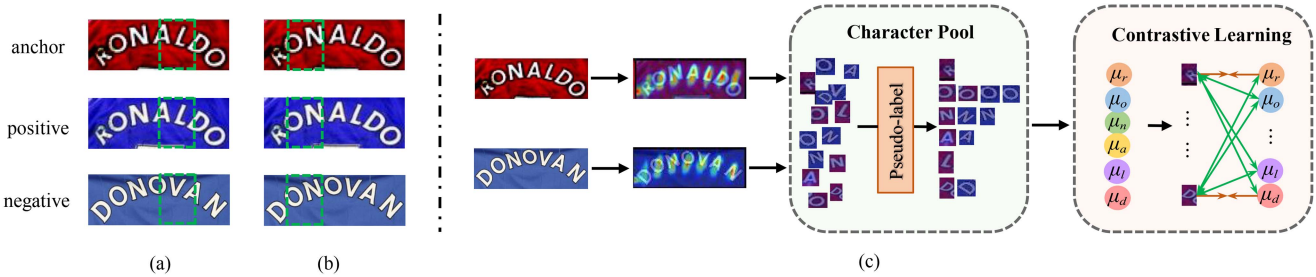


Fig. 3. (a) and (b) are examples of *anchor*, *positive*, and *negative* for subword-based contrastive learning. (a) is an example of slicing complete characters, and (b) is an example of semantic misalignment. (c) is our proposed character-based contrastive learning.

where $p(z_i^l)$ or $p(z_i^u)$ is the probability after *softmax* function, and M^l or M^u is the number of filtered character features. If the probability belonging to the source label z_i^l or the target pseudo-label z_i^u is above the threshold η, this character feature is involved in alignment; otherwise, it is discarded.

2) Character-Based Contrastive Learning: Given the pseudo-labeled character features, we argue that each character class has a corresponding prototype in the feature space. These prototypes can be calculated by taking the average of all embedded samples belonging to that class. Thus, there are three ways to compute prototypes: source-only, target-only, and source-target domains. This paper calculates the prototypes from the source-only domain because they can yield more stable class prototypes. On the one hand, only the source data contains supervision information, and the participation of the target data in the prototype generation may introduce noise; on the other hand, the semantic is cross-domain in text recognition tasks, i.e., the semantics of the same characters do not change with synthetic text, scene text, or handwriting. The related ablation studies are in Section IV-D4. Formally, the prototype μ_k of class k is defined as follows,

$$\mu_k = \frac{1}{|\mathcal{C}_k^L|} \sum_{c_i^l \in \mathcal{C}_k^L} c_i^l \,, \tag{8}$$

where $\mathcal{C}_k^L$ is the character features of class k in source domain.

Self-supervised contrastive learning is employed to align the fine-grained source and target domain features. The basic idea is that similar characters should cluster around their class prototype while staying away from other class prototypes, as illustrated in Fig. 3(c). For a character (c^*, z^*) of class k in the source or target domain, its positive case is the prototype μ_k, and its negative cases are the prototypes of the other classes. The contrastive-based contrastive loss is formulated as follows,

$$\mathcal{L}_{cont} = -\frac{1}{|\mathcal{C}^L| + |\mathcal{C}^U|} \sum_{c^* \in \mathcal{C}^L \cup \mathcal{C}^U} \log \frac{\mathcal{I}(z^* = k) \exp(c^* \cdot \mu_k / \tau)}{\sum_{k=1}^{K} \exp(c^* \cdot \mu_k / \tau)} \,, \tag{9}$$

where $\mathcal{I}(z^* = k)$ is an indicator, K is the number of categories, and τ is a temperature hyperparamers. '$\cdot$' denotes the dot production used for measuring the similarity between the character feature c^* and the class prototype μ_k.

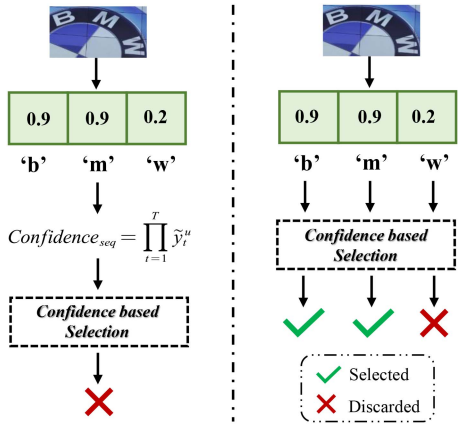


Fig. 4. The left indicates the instance selection based on the confidence of the entire sequence. The right is our proposed instance selection strategy based on character confidence.

In this way, on the one hand, it maintains the integrity of characters by avoiding sliding windows to slice particular characters. On the other hand, it mitigates misalignment due to the uncertainty of negative cases. In addition, category-level adaptation enables fine-grained character feature alignment, facilitating learning of fine-grained domain-invariant character representations.

D. Instance Consistency Matching

Text recognition is a sequence task that involves understanding the context of the sequence. Still, character-based contrastive learning only focuses on individual characters without considering their relationships within the sequence. To address this limitation, we further propose an instance consistency matching module that employs consistency regularization to capture the contextual semantics of the target data. This module aims to enhance the ability to perceive and generalize by ensuring consistent predictions on target data with different augmentation views.

Consistency regularization was first introduced to STR in [6]. However, this method filters out noisy characters based on the confidence of the entire sequence, which may lead to *low utilization of target samples*. For example, as shown in the left of Fig. 4, a high-confidence character ('b' and 'm') in a low-confidence

Fig. 5. Visualization of weak and strong augmentations.

sequence may be valuable, but it is discarded due to the low confidence of the sequence. Therefore, we present a simple instance selection strategy based on character confidence, as illustrated in the right of Fig. 4. Technically, a probability threshold δ is introduced at each time step to select high-confidence instances as pseudo labels in the proposed triple-matching consistency loss.

In addition, to ensure effective matching and better perception, triple-matching is designed by making consistent model predictions. Specifically, given a raw target sample x^u, two different augmentation ways are adopted to produce weak and strong augmented views, denoted as x^u_w and x^u_s, respectively. After decoding, the predictions are denoted as $\tilde{y}^u = \{\tilde{y}^u_1, \tilde{y}^u_t, \ldots, \tilde{y}^u_T\}$, $\tilde{y}^u_w = \{\tilde{y}^u_{w,1}, \tilde{y}^u_{w,t}, \ldots, \tilde{y}^u_{w,T}\}$, and $\tilde{y}^u_s = \{\tilde{y}^u_{s,1}, \tilde{y}^u_{s,t}, \ldots, \tilde{y}^u_{s,T}\}$, respectively. Then, triple-matching is performed between the target data with different augmentation views. The triple-matching consistency loss is defined as follows,

$$\mathcal{L}_{cons}(\tilde{y}^u, \tilde{y}^u_w, \tilde{y}^u_s) = \mathcal{L}(\tilde{y}^u, \tilde{y}^u_w) + \mathcal{L}(\tilde{y}^u, \tilde{y}^u_s) + \mathcal{L}(\tilde{y}^u_w, \tilde{y}^u_s), \tag{10}$$

$$\mathcal{L}(\tilde{y}^u, \tilde{y}^u_w) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{I}(p(\tilde{y}^u_t|x^u) \geq \delta) Dist(\tilde{y}^u_t, \tilde{y}^u_{w,t}), \tag{11}$$

$$\mathcal{L}(\tilde{y}^u, \tilde{y}^u_s) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{I}(p(\tilde{y}^u_t|x^u) \geq \delta) Dist(\tilde{y}^u_t, \tilde{y}^u_{s,t}), \tag{12}$$

$$\mathcal{L}(\tilde{y}^u_w, \tilde{y}^u_s) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{I}(p(\tilde{y}^u_{w,t}|x^u_w) \geq \delta) \times Dist(\tilde{y}^u_{w,t}, \tilde{y}^u_{s,t}), \tag{13}$$

where $\mathcal{I}(p(*) \geq \delta)$ is an indicator, δ is a scalar probability threshold for filtering out noisy characters, and $Dist(a, b)$ is a function to measure the discrepancy between a and b. Our framework adopts a standard cross-entropy as *Dist*.

The two different augmentation ways are *WeakAug* and *StrongAug*, as defined in [6]. The former includes color jitter, such as brightness, contrast, and hue changes. The latter contains color jitter and geometry transformations, except for cropping, which could potentially alter the sequence semantics. Examples of the scene and handwritten text augmentations are visualized in Fig. 5.

E. Overall Objective Function

The overall objective integrates the cross-entropy loss in (6), the character-based contrastive loss in (9), and the triple-matching consistency loss in (10). Hence, we obtain the following optimization function,

$$\mathcal{L}_{overall} = \mathcal{L}_{CE} + \lambda_{cont}\mathcal{L}_{cont} + \lambda_{cons}\mathcal{L}_{cons}, \tag{14}$$

where λ_{cont} and λ_{cons} are trade-off parameters. With this loss, TextAdapter can learn fine-grained domain-invariant character representations and perceive the target semantics simultaneously.

IV. Experiments

In this section, we experimentally examine TextAdapter on nine benchmark datasets. Firstly, datasets and experimental settings are presented. Then, comparison and ablation studies are conducted. Lastly, we show parameter analysis and visualization results.

A. Datasets

Three types of 11 datasets are used in the UDA of text recognition.

Synthetic Text: Synth90k (MJ) [39] contains 8.9 million images generated from a set of 90 k common English words. Synth-Text (ST) [40] contains 5.5 million images with English words. MJ and ST are generally used for the source domain.

Real Scene Text: Seven benchmarks are tested, including four regular datasets, i.e., IIIT5K [41], SVT [42], IC03 [43], and IC13 [44], and three irregular datasets, i.e., SVTP [45], CUTE80 [46], and IC15 [47]. Details of datasets can be found in the previous work [48].

Handwritten Text: IAM [49] is an English handwritten text dataset written by 657 writers. According to standard partition [50], IAM [1] is divided into 53841 training words, 8566 validation words, and 17616 test words. CVL [51] is a public dataset written by 310 writers for writer retrieval, writer identification, and word spotting. It contains 12289 training words and 84949 test words.

B. Experimental Settings

1) Implementation Details: All experiments are conducted using PyTorch on an NVIDIA GeForce RTX 2080Ti GPU. We evaluate the effectiveness and generality of TextAdapter by embedding it into three representative STR models, including TRBA [48], Scatter [53], and ABINet [23], following their original setting. *Adadelta* optimizer is used during the adaptation process when training TRBA and Scatter, and *Adam* optimizer

[1]https://fki.tic.heia-fr.ch/databases/iam-handwriting-database

TABLE I
EVALUATION RESULTS ON THE ADAPTATION FROM SYNTHETIC TO SCENE TEXT COMPARED WITH SOTA METHODS

	Methods	Labeled	Unlabeled	Regular Text				Irregular Text			Avg.
				IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
SOTA Methods	CRNN(TPAMI2017) [14]	MJ	-	82.90	81.60	93.10	91.10	-	-	-	-
	GRCNN(NIPS2017) [15]	MJ+PRI	-	84.20	83.70	93.50	90.90	-	-	-	-
	Char-Net(AAAI2018) [52]	MJ	-	83.60	84.40	91.50	90.80	-	-	60.00	-
	TRBA(ICCV2019) [48]	MJ+ST	-	87.90	87.50	94.90	93.60	79.20	74.00	77.60	-
	Aster(TPAMI2019) [20]	MJ+ST	-	93.40	89.50	94.50	-	78.50	79.50	76.10	-
	Scatter(CVPR2020) [53]	MJ+ST+SA	-	93.70	92.70	-	-	86.90	87.50	-	-
	SRN(CVPR2020) [54]	MJ+ST	-	94.80	91.50	-	95.50	85.10	87.80	82.70	-
	ABINet(CVPR2021) [23]	MJ+ST	-	96.20	93.50	-	97.40	89.30	89.20	86.00	-
	TPS++(IJCAI2023) [55]	MJ+ST	-	96.30	94.30	-	97.80	89.60	89.60	86.50	-
	PerSec(AAAI2022) [11]	MJ+ST	UTI(100M)	88.10	86.80	-	94.20	77.70	72.70	73.60	-
	ConCLR(AAAI2022) [36]	MJ+ST	OutText(1k)	96.50	94.30	-	97.70	89.30	91.30	85.40	-
	DiG(MM2022) [5]	MJ+ST	URD(15.77M)	**96.70**	94.60	-	96.90	**91.00**	91.30	87.10	
	Zheng *et al.*(CVPR2022)* [6]	MJ+ST	Real(6.9k)	90.70	91.96	96.16	96.15	86.98	84.72	83.77	-
	SSDAN(CVPR2019) [8]	MJ+ST	Real(6.9k)	87.60	88.10	94.60	93.80	-	73.90	78.70	-
	ASSDA(TIP2021) [9]	MJ+ST	Real(6.9k)	88.30	88.60	95.50	93.70	-	76.30	78.70	-
	SMILE(ICIP2022) [7]	MJ+ST	Real(6.9k)	89.30	87.60	96.00	94.90	-	75.60	78.90	-
	DOC(TMM2023) [56]	MJ+ST	Real(6.9k)	89.00	89.00	95.30	94.30	81.20	77.00	76.00	-
	CADA(TCSVT2023) [57]	MJ+ST	Real(6.9k)	89.30	89.03	95.35	95.10	81.55	78.40	80.12	-
Ours	TRBA*	MJ+ST	-	87.07	86.86	95.35	92.88	79.69	74.56	77.75	85.43
	TRBA-TextAdapter	MJ+ST	Real(6.9k)	90.67	90.73	96.05	95.68	83.88	81.53	82.17	89.01
	Scatter*	MJ+ST	-	89.47	89.49	96.98	94.40	82.64	77.43	80.34	87.78
	Scatter-TextAdapter	MJ+ST	Real(6.9k)	92.10	91.81	96.16	95.22	84.50	82.29	84.76	90.24
	ABINet*	MJ+ST	-	96.10	93.97	95.93	96.50	89.30	91.67	85.31	92.84
	ABINet-TextAdapter	MJ+ST	Real(6.9k)	96.33	**95.05**	**97.33**	**98.25**	90.54	**92.71**	**87.91**	**94.07**

'*' indicates the reproduced results. The numbers in parentheses denote the amounts, e.g., 100 M means 100 million.
The bold values indicate the best results and underlined ones indicate the sub-optimal results.

is used for ABINet. The learning rates are initialized to 0.1 and 1 for TRBA and Scatter and 0.0001 for ABINet, respectively. The maximum decoding length T is set to 25, and the temperature τ is set to 1 empirically. The training iteration is set to 300 k with a batch size 48.

2) Evaluation Metric: Word-level accuracy is adopted for STR. To comprehensively evaluate the performance, we introduce an average metric *Avg.* that averages the results over all samples in seven real scene datasets. Following standard practice, word error rate (WER) and character error rate (CER) are reported for HTR.

C. Comparison With SOTAs

We compare our TextAdapter framework with other SOTA methods, especially self-supervised and UDA-based ones. Three STR baseline models based on attention-RNN decoder (TRBA and Scatter) and transformer decoder (ABINet) are utilized to verify the effectiveness and generality. Specifically, the fine-grained feature alignment module is embedded in the semantic features extracted by BiLSTM of TRBA and Scatter and those extracted by the last iteration of the language model of ABINet. The instance consistency matching module is applied directly to the output of TRBA, the last decoder output of Scatter, and the last iteration output of the alignment module of ABINet. We also reproduce those STR methods with the original settings under the same training set for a fair comparison. As presented in Table I, our reproduced results, including TRBA*, Scatter*, and ABINet*, have comparable or even higher accuracies than those reported in the original papers. These reproduced models serve as pre-trained baseline models for our framework.

1) Synthetic Text to Scene Text: Since the self-supervised methods [5], [6], [11], [36] utilize diverse unlabeled real text, we reproduce [6] as a representation under the same training set, with the source data being MJ and ST and the target data being the union of IIIT5K, SVT, IC13, and IC15 training splits, following common UDA methods [9], [56]. From the results in Table I, we can observe:

- TRBA-TextAdapter, which uses the same baseline as UDA methods [7], [8], [9], [56], [57], shows significant improvement on all seven datasets. Other UDA-based methods are less effective due to the indistinguishable character semantics caused by the domain-level adaptation, especially on irregular text with complex situations. The performance gains of TRBA-TextAdapter on irregular text are more pronounced than those on regular text. Specifically, compared to CADA, TRBA-TextAdapter achieves gains of 2.33%, 3.13%, and 2.05% on irregular text, while gains of 1.37%, 1.70%, 0.05%, and 0.58% on regular text, respectively. The substantial improvement in irregular text illustrates the ability of the fine-grained feature alignment module to align character features and transfer the knowledge learned on synthetic text to irregular text.
- Our TextAdapter based models outperform the corresponding non-TextAdapter baseline models on almost all datasets. Specifically, the average results are improved by 3.58% (85.43%→ 89.01%), 2.46% (87.78%→90.24%), and 1.23% (92.84%→94.07%), respectively. However, compared to the result of Scatter* on IC03, that of Scatter-TextAdapter decreases. We believe this may be possible. Our optimization objective is global optimality, i.e., better average *Avg.*. In the case of global relative optimality, there

TABLE II
EVALUATION RESULTS ON THE TASK FROM SYNTHETIC TEXT TO HANDWRITTEN TEXT

	Methods	Syn→IAM		Syn→CVL	
		WER↓	CER↓	WER↓	CER↓
SOTA	SSDAN(CVPR2019) [8]	53.65	27.26	-	-
	ASSDA(TIP2021) [9]	43.78	19.96	-	-
	SMILE(ICIP2022)* [7]	45.57	19.35	64.63	30.34
	CADA(TCSVT2023) [57]	45.70	19.67	67.34	32.88
	DOC(TMM2023) [56]	37.44	16.52	-	-
Ours	TRBA*	57.07	30.90	72.28	40.08
	TRBA-TextAdapter	25.75	9.76	53.10	23.11
	Scatter*	53.72	29.50	71.21	40.40
	Scatter-TextAdapter	**17.76**	**7.30**	**28.28**	**12.20**
	ABINet*	46.61	31.44	58.27	38.73
	ABINet-TextAdapter	41.39	27.34	51.21	31.26

'*' denotes the reproduced results.
The bold values indicate the best results and underlined ones indicate the sub-optimal results.

TABLE III
EVALUATION RESULTS OF DIFFERENT COMPONENTS ON THE TASK FROM SYNTHETIC TEXT TO SCENE TEXT

Model	$\mathcal{L}_{cont}$	$\mathcal{L}_{cons}$	IIIT5K	SVT	IC03	IC13
			SVTP	CUTE80	IC15	Avg.
Baseline	✗	✗	87.07	86.86	95.35	92.88
			79.69	74.56	77.75	85.43
w/o $\mathcal{L}_{cons}$	✓	✗	87.47	87.94	95.47	94.82
			80.00	77.96	77.80	86.37
w/o $\mathcal{L}_{cont}$	✗	✓	90.40	90.57	95.47	95.40
			84.03	80.14	82.55	88.60
TextAdapter	✓	✓	90.67	90.73	96.05	95.68
			83.88	81.53	82.17	**89.01**

The bold value indicates the best result.

TABLE IV
EVALUATION RESULTS OF DIFFERENT MATCHING OBJECTS ON THE TASK FROM SYNTHETIC TEXT TO SCENE TEXT

Model	Matching	IIIT5K	SVT	IC03	IC13
		SVTP	CUTE80	IC15	Avg.
w/o $\mathcal{L}_{cont}$	raw-weak	89.67	88.87	95.58	95.01
		80.00	78.05	79.57	87.37
	raw-strong	90.10	89.18	95.16	94.87
		82.95	80.53	81.61	88.11
	weak-strong	90.10	89.95	95.47	94.63
		82.48	80.49	82.22	88.13
	raw-weak-strong	90.40	90.57	95.47	95.40
		84.03	80.14	82.55	**88.60**

The bold value indicates the best result.

may be local non-optimality, i.e., performance degradation of individual datasets. These improvements in average results highlight the generality of TextAdapter, further enhancing its performance with the benefit of off-the-shelf STR models.

- ABINet-TextAdapter achieves comparable results to SOTA methods, particularly self-supervised based ones. Although DiG performs slightly better on IIIT5K (96.70% *vs.* 96.33%) and SVTP (91.00% *vs.* 90.54%), it uses 15.77 M private unlabeled data while we only use 6.9 k unlabeled data. This shows that our self-supervised TextAdapter can extract distinguishing character features.

2) Synthetic Text to Handwritten Text: We further evaluate TextAdapter on adapting synthetic text to handwritten text, where domain gaps are more obvious due to unique stroke fluency and semantic characteristics of handwritten text. The source domain is MJ and ST, and the target domain is the training set of the IAM or CVL dataset. From Table II, we can see that:

- Compared to SOTA methods, TRBA-TextAdapter significantly improves the WER and CER on IAM and CVL datasets and achieves the best results using the same baseline. Specifically, compared to DOC, the WER and CER are reduced by 11.69% (37.44%→25.75%) and 6.76% (16.52%→9.76%) on the IAM. This improvement is attributed to the capability of extracting fine-grained domain-invariant representations while also enhancing the robustness of HTR through its perception of handwritten-specific semantics.
- Compared to the three non-TextAdapter baselines, our models are all enhanced. For instance, Scatter-TextAdapter reduces the WER and CER of IAM by 35.96% (53.72% →17.76%) and 22.20% (29.50%→7.30%), respectively. By utilizing the model originally specialized for STR, TextAdapter also enhances the performance of HTR.
- Compared to TRBA- and Scatter-, ABINet-TextAdapter shows only slight improvements. This may be due to the more apparent semantic discrepancy between synthetic and handwritten texts. The explicit modeling of the language model in ABINet may lead to an over-reliance on synthetic text semantics, limiting its ability to perceive semantics in a few unlabeled handwritten texts.

D. Ablation Study

Due to the simplicity of the TRBA* baseline, we used it to perform ablation experiments and analyze the proposed method. In the following experiments, 'Baseline' and 'TextAdapter' denote TRBA* and TRBA-TextAdapter in Table I.

1) Effect of Each Component: We conduct ablation experiments to analyze the effectiveness of the fine-grained feature alignment module and instance consistency matching module by removing the corresponding loss terms $\mathcal{L}_{cont}$ and $\mathcal{L}_{cons}$. The baseline is trained only on labeled synthetic text using the cross-entropy loss. The results in Table III show that combining both $\mathcal{L}_{cont}$ and $\mathcal{L}_{cons}$ with the baseline leads to a 3.58% average improvement. When only one of the loss terms is used with the cross-entropy loss, the average performance improves by 0.94% (+$\mathcal{L}_{cont}$) or 3.17% (+$\mathcal{L}_{cons}$). These results validate the positive effect of the proposed modules.

2) Effect of Matching Objects: The instance consistency matching module aligns the prediction of unlabeled images and their weak and strong augmented views. We conduct experiments of the matching objects to demonstrate the validity of the triple-matching, which contains raw-weak, raw-strong, and weak-strong matchings. Table IV shows that the raw-weak matching has a slight average boost compared to the raw-strong and weak-strong matchings, possibly due to the weaker data

TABLE V
EVALUATION RESULTS OF DIFFERENT MEASURE FUNCTIONS ON THE TASK FROM SYNTHETIC TEXT TO SCENE TEXT

Model	Dist(·)	IIIT5K	SVT	IC03	IC13
		SVTP	CUTE80	IC15	Avg.
w/o $\mathcal{L}_{cont}$	Cross-Entropy	90.40	90.57	95.47	95.40
		84.03	80.14	82.55	**88.60**
	KL-Divergence	89.17	88.10	95.58	94.63
		78.92	77.70	78.13	86.65

The bold value indicates the best result.

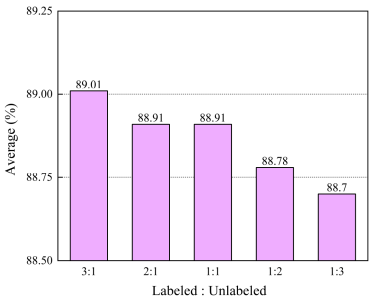


Fig. 6. Evaluation average results of different proportions of labeled and unlabeled images.

diversity of the weak augmentation. Triple-matching performs better than any single-matching, indicating a more comprehensive perception of the semantic information.

3) Effect of Measure Functions: One issue in the instance consistency matching module is whether to match pseudo-labels or the probability distribution for the matching process. Therefore, we test the cross-entropy and KL-divergence as the measure functions. Results in Table V show that matching pseudo-labels outperforms matching the probability distribution. The reason may be that the matching probability distribution of the target images without supervised information can result in over-matching when feature extraction is insufficient, thereby affecting performance.

4) Effect of Contrastive Cases: A crucial issue in the fine-grained feature alignment module is determining which features can generate robust class prototypes as positive and negative cases. Since the target domain lacks supervised information, the generated class prototypes have two options: source-only (*src*) and source-target (*src+tar*). We design several anchor ways based on these options. Table VI shows that when generating class prototypes from the source-only domain, the model can yield more stable class prototypes. Thus, contrastive learning could effectively cluster similar characters while keeping away from other class prototypes.

5) Effect of Minibatch Form: The ratio of labeled to unlabeled images in a minibatch is essential in determining the transfer intensity. Therefore, we explore the effect of different transfer intensities on model performance. As depicted in Fig. 6, using a ratio of 3:1 for labeled and unlabeled images can result in better knowledge transfer and extracting of domain-invariant representations.

6) Generality in Generative Models: To further explore the generality of TextAdapter on the generative paradigm, we deploy it to a representative generative model DiG based on its official code, named DiG-TextAdapter. For fairness, we list the results in the original paper and those we reproduced separately. From Table VII, it can be seen that some reproduced results even exceed the results in the original paper. After deploying TextAdapter, the average result improves by 2.72% (90.81%→93.53%). This illustrates that TextAdapter can also facilitate the optimization of generative models.

Extensive ablation experiments show that the carefully designed sub-modules are individually effective, and collaborative efforts promote improved behavior. In addition, TextAdapter can also further enhance generative methods.

TABLE VI
EVALUATION RESULTS OF DIFFERENT PROTOTYPES AND ANCHORS ON THE TASK FROM SYNTHETIC TEXT TO SCENE TEXT OF THE TEXTADAPTER *W/O* $\mathcal{L}_{cons}$

Prototype	Anchor	IIIT5K	SVT	IC03	IC13
		SVTP	CUTE80	IC15	Avg.
src	src	87.93	87.48	95.93	94.40
		80.00	77.35	77.86	86.17
	tar	87.90	87.79	95.47	94.17
		80.31	77.35	77.64	86.09
	src+tar	87.47	87.94	95.47	94.82
		80.00	77.96	77.80	**86.37**
	tar+aug	88.17	87.48	95.00	94.40
		79.38	75.96	77.91	86.07
	src+tar+aug	87.83	87.48	95.47	94.17
		79.38	76.66	77.80	85.98
src+tar	src	88.17	88.25	94.88	94.28
		79.38	77.00	78.63	86.31
	tar	88.37	88.25	95.12	94.28
		78.92	77.70	78.24	86.31
	src+tar	88.37	87.79	95.00	94.52
		79.85	77.70	77.58	86.21
	tar+aug	88.57	87.33	95.58	94.52
		78.61	76.66	78.30	86.33
	src+tar+aug	88.13	87.64	95.47	93.93
		80.62	77.00	78.58	86.36

'Src' and 'tar' indicate the character features of the source and target domains. 'Aug' means the weak and strong augmentations of target character features. The bold value indicates the best result.

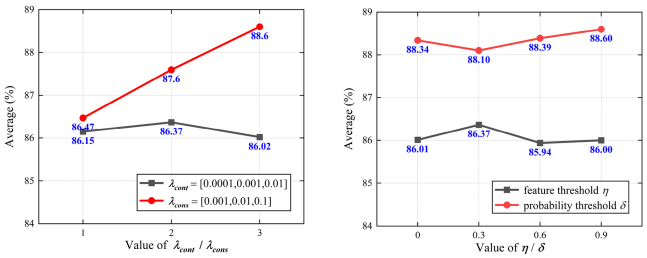
TABLE VII
EVALUATION RESULTS OF TEXTADAPTER DEPLOYED TO THE GENERATIVE MODEL DIG

Model	Labeled	Unlabeled	IIIT5K	SVT	IC03	IC13
			SVTP	CUTE80	IC15	Avg.
DiG [5]	MJ+ST	URD	96.70	94.60	-	96.90
			91.00	91.30	87.10	-
DiG*	MJ+ST	URD	86.90	94.75	97.21	96.97
			92.87	94.10	88.68	90.81
DiG-TextAdapter	MJ+ST	URD+ Real	93.27	97.06	97.67	97.55
			92.40	95.83	88.85	93.53

'*' denotes the reproduced results.

E. Algorithm Analysis

1) Parameter Sensitive Analysis: We analyze the sensitivity of the hyperparameters, including two trade-off hyperparameters and two threshold hyperparameters. We vary these



(a) λ_{cont} and λ_{cons} (b) η and δ

Fig. 7. Effect of hyperparameters on *Avg.* results on the synthetic to scene text task. The experiments of λ_{cont} and η are based on TextAdapter *w/o* $\mathcal{L}_{cons}$ model. The experiments of λ_{cons} and δ are based on TextAdapter *w/o* $\mathcal{L}_{cont}$ model.

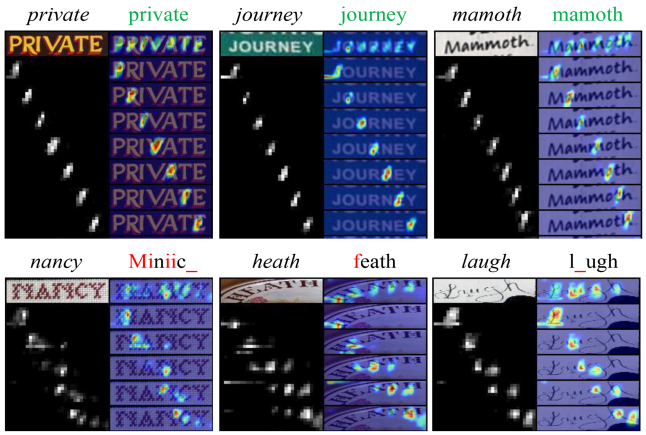
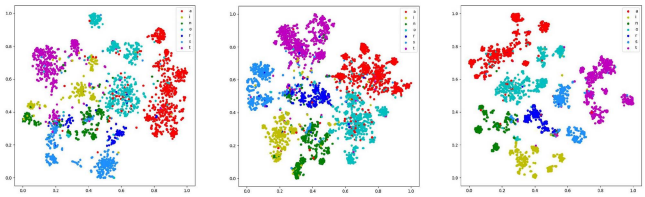


Fig. 8. Visualization of attention and prediction results of TextAdapter. *Italics* indicate ground truth. Green and red colors indicate correct and incorrect prediction results, respectively.

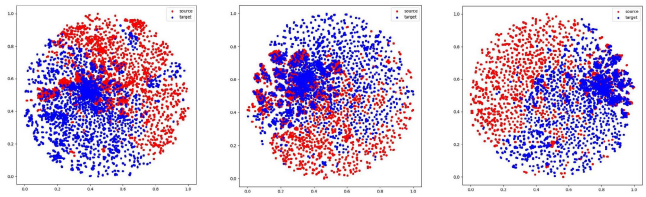


(a) Baseline (b) ASSDA (c) TextAdapter

Fig. 9. Visualization of target domain character features on Syn → IAM task.

trade-off hyperparameters $\lambda_{cont} \in \{0.0001, 0.001, 0.01\}$ and $\lambda_{cons} \in \{0.001, 0.01, 0.1\}$. From Fig. 7(a), we find that the best performance is achieved when λ_{cont} is 0.001 and λ_{cons} is 0.1. For the threshold hyperparameters, we also vary η and δ in $\{0, 0.3, 0.6, 0.9\}$. From Fig. 7(b), we observed that our default settings of η=0.3 and δ=0.9 are optimal.

2) Visualization: To provide insight into the effectiveness of TextAdapter, we first visualize the attention and prediction results. As seen from Fig. 8, accurate character localization can facilitate recognition, as error character localization can result in sequence length or prediction errors. Additionally, we use t-SNE to visualize two feature distributions: character features of the target domain and character features of both the source and target domains. For the first visualization, we randomly select several character categories to show. Compared to Fig. 9(a) baseline and Fig. 9(b) ASSDA, TextAdapter presents a more discriminative distribution of target characters. For the second visualization, as shown in Fig. 10, TextAdapter brings the two distributions closer together, making the target distribution more indistinguishable from the source one. These results once again validate the effectiveness of TextAdapter.



(a) Baseline (b) ASSDA (c) TextAdapter

Fig. 10. Visualization of source and target domain character features on Syn → IAM task.

V. Conclusion

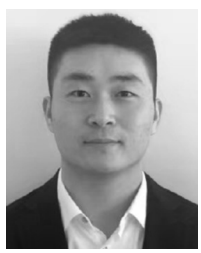
This article proposes a simple yet effective self-supervised domain adaptation framework called TextAdapter for cross-domain text recognition, aimed at bridging the domain gaps between synthetic text and real text. The framework consists of a fine-grained feature alignment module and an instance consistency matching module, which combine to extract fine-grained domain-invariant character representations. Extensive experiments on nine benchmarks demonstrate the superiority and generality of our framework.

References

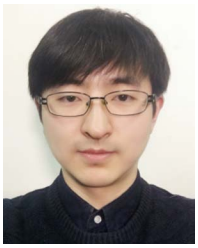
[1] P. Wang et al., "PGNet: Real-time arbitrarily-shaped text spotting with point gathering network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2782–2790.
[2] Y. He et al., "Visual semantics allow for textual reasoning better in scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 888–896.
[3] J. Baek, Y. Matsui, and K. Aizawa, "What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3113–3122.
[4] J. Lin et al., "Text recognition in real scenarios with a few labeled samples," in *Proc. IEEE Int. Conf. Pattern Recognit.*, 2020, pp. 370–377.
[5] M. Yang et al., "Reading and writing: Discriminative and generative modeling for self-supervised text recognition," in *Proc. ACM Multimedia Conf.*, 2022, pp. 4214–4223.
[6] C. Zheng et al., "Pushing the performance limit of scene text recognizer without human annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14096–14105.
[7] Y. Chang, Y. Chen, Y. Chang, and Y. Yeh, "Smile: Sequence-to-sequence domain adaptation with minimizing latent entropy for text image recognition," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 431–435.
[8] Y. Zhang et al., "Sequence-to-sequence domain adaptation network for robust text image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2740–2749.
[9] Y. Zhang, S. Nie, S. Liang, and W. Liu, "Robust text image recognition via adversarial sequence-to-sequence domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 3922–3933, 2021.
[10] A. Aberdam et al., "Sequence-to-sequence contrastive learning for text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15302–15312.
[11] H. Liu et al., "Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1702–1710.
[12] L. Wu, Y. Xu, J. Hou, C. L. P. Chen, and C. Liu, "A two-level rectification attention network for scene text recognition," *IEEE Trans. Multim.*, vol. 25, pp. 2404–2414, 2023.

[13] M. Li, B. Fu, Z. Zhang, and Y. Qiao, "Character-aware sampling and rectification for scene text recognition," *IEEE Trans. Multimedia*, vol. 25, pp. 649–661, 2023.
[14] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.
[15] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 335–344.
[16] B. Su and S. Lu, "Accurate recognition of words in scenes without character segmentation using recurrent neural network," *Pattern Recognit.*, vol. 63, pp. 397–405, 2017.
[17] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4168–4176.
[18] A. K. Bhunia et al., "Joint visual semantic reasoning: Multi-stage decoder for text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14920–14929.
[19] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8610–8617.
[20] B. Shi et al., "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.
[21] Z. Cheng et al., "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5086–5094.
[22] N. Lu et al., "MASTER: Multi-aspect non-local network for scene text recognition," *Pattern Recognit.*, vol. 117, 2021, Art. no. 107980.
[23] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7098–7107.
[24] Y. Wang et al., "From two to one: A new scene text recognizer with visual language modeling network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14174–14183.
[25] R. Wang et al., "Cross-domain contrastive learning for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 25, pp. 1665–1673, 2023.
[26] W. Deng et al., "Informative feature disentanglement for unsupervised domain adaptation," *IEEE Trans. Multimedia*, vol. 24, pp. 2407–2421, 2022.
[27] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2962–2971.
[28] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, "Unsupervised pixel-level domain adaptation with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 95–104.
[29] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A Kernel method for the two-sample-problem," in *Proc. Neural Inf. Process. Syst.*, 2006, pp. 513–520.
[30] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:abs/1412.3474*.
[31] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.
[32] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. Workshops (3)*, 2016, pp. 443–450.
[33] J. Li, G. Li, Y. Shi, and Y. Yu, "Cross-domain adaptive clustering for semi-supervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2505–2514.
[34] A. K. Bhunia et al., "Meta HTR: Towards writer-adaptive handwritten text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15830–15839.
[35] S. Azadi et al., "Multi-content GAN for few-shot font style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7564–7573.
[36] X. Zhang et al., "Context-based contrastive learning for scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 3353–3361.
[37] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 1163–1171.
[38] A. Aberdam, R. Ganz, S. Mazor, and R. Litman, "Multimodal semi-supervised learning for text recognition," 2022, *arXiv:abs/2205.03873*.
[39] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," 2014, *arXiv:abs/1406.2227*.
[40] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.
[41] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.
[42] K. Wang, B. Babenko, and S. J. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1457–1464.
[43] S. M. Lucas et al., "ICDAR 2003 robust reading competitions: Entries, results, and future directions," *Int. J. Document Anal. Recognit.*, vol. 7, no. 2/3, pp. 105–122, 2005.
[44] D. Karatzas et al., "ICDAR 2013 robust reading competition," in *Proc. IEEE Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.
[45] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 569–576.
[46] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.
[47] D. Karatzas et al., "ICDAR 2015 competition on robust reading," in *Proc. IEEE Int. Conf. Document Anal. Recog.*, 2015, pp. 1156–1160.
[48] J. Baek et al., "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4714–4722.
[49] U. Marti and H. Bunke, "The IAM-database: An english sentence database for offline handwriting recognition," *Int. J. Document Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, 2002.
[50] J. Sueiras, "Continuous offline handwriting recognition using deep learning models," 2021, *arXiv:abs/2112.13328*.
[51] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-Database: An off-line database for writer retrieval, writer identification and word spotting," in *Proc. Int. Conf. Document Anal. Recog.*, 2013, pp. 560–564.
[52] W. Liu, C. Chen, and K. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7154–7161.
[53] R. Litman et al., "SCATTER: Selective context attentional scene text recognizer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11959–11969.
[54] D. Yu et al., "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12110–12119.
[55] T. Zheng, Z. Chen, J. Bai, H. Xie, and Y. Jiang, "TPS++: Attention-enhanced thin-plate spline for scene text recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2023, pp. 1777–1785.
[56] X. Ding, X. Liu, X. Luo, and X. Xu, "DOC: Text recognition via dual adaptation and clustering," *IEEE Trans. Multimedia*, vol. 25, pp. 9071–9081, 2023.
[57] X. Liu, X. Ding, X. Luo, and X. Xu, "Unsupervised domain adaptation via class aggregation for text recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 10, pp. 5617–5630, Oct. 2023.

Xiao-Qian Liu received the M.S. degree in control engineering in 2020 from Shandong University, Jinan, China, where she is currently working toward the Ph.D. degree in artificial intelligence with the School of Software. Her research interests include deep learning, computer vision, domain adaptation, and OCR.



Peng-Fei Zhang received the B.Sc. and M.S. degrees from Shandong University, Jinan, China, in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree with the School of Electrical Engineering and Computer Science, University of Queensland, Brisbane, QLD, USA. His research interests include machine learning, information retrieval, privacy protection, and multimedia analysis and search.



Xin Luo received the Ph.D. degree in computer science from Shandong University, Jinan, China, in 2019. He is currently an Assistant Professor with the School of Software, Shandong University. His research interests include machine learning, multimedia retrieval, and computer vision. He has authored or coauthored more than 20 papers on IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Multimedia Computing, Communications, and Applications*, SIGIR, *World Wide Web*, and *International Journal of Computing and Artificial Intelligence*. He is a Reviewer for ACM International Conference on Multimedia, International Joint Conference on Artificial Intelligence, AAAI Conference on Artificial Intelligence, IEEE TRANSACTIONS ON CYBERNETICS, *Pattern Recognition*, and other prestigious conferences and journals.

Zi Huang (Senior Member, IEEE) received the B.Sc. degree from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the University of Queensland, Brisbane, QLD, Australia. She is currently a Professor with the School of Electrical Engineering and Computer Science, University of Queensland. Most of her publications have been published in leading conferences and journals, including ACM Multimedia, ACM SIGMOD, IEEE ICDE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *ACM Transactions on Information Systems*, and *ACM Computing Surveys*. Her research interests include multimedia search, social media analysis, database, and information retrieval.

Xin-Shun Xu (Senior Member, IEEE) received the M.S. degree in computer science from Shandong University, Jinan, China, and the Ph.D. degree in computer science from Toyama University, Toyama, Japan, in 2002 and 2005, respectively. He is currently a Professor with the School of Software, Shandong University, Jinan, China. He joined the School of Computer Science and Technology, Shandong University as an Associate Professor in 2005, and LAMDA Group, Nanjing University, Nanjing, China, as a Postdoctoral Fellow in 2009. From 2010 to 2017, he was a Professor with the School of Computer Science and Technology, Shandong University. He has authored or coauthored IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, AAAI, CIKM, *International Journal of Computing and Artificial Intelligence*, *Multimedia*, SIGIR, *World Wide Web,* and other venues. His research interests include machine learning, information retrieval, data mining, and image/video analysis and retrieval. He is the Founder and Leader of Machine Intelligence and Media Analysis Group, Shandong University. He is a SPC/PC Member or Reviewer for various international conferences and journals, such as AAAI, CIKM, CVPR, ICCV, *International Journal of Computing and Artificial Intelligence*, *Multimedia*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE.