Category Contrast for Unsupervised Domain Adaptation in Visual Tasks

Jiaxing Huang 1, Dayan Guan 1, Aoran Xiao 1, Shijian Lu* 1, Ling Shao 2
1 Nanyang Technological University, Singapore 2 Terminus Group, China
{Jiaxing.Huang, Dayan.Guan, Aoran.Xiao, Shijian.Lu}@ntu.edu.sg, ling.shao@ieee.org

Abstract

Instance contrast for unsupervised representation learning has achieved great success in recent years. In this work, we explore the idea of instance contrastive learning in unsupervised domain adaptation (UDA) and propose a novel Category Contrast technique (CaCo) that introduces semantic priors on top of instance discrimination for visual UDA tasks. By considering instance contrastive learning as a dictionary look-up operation, we construct a semantics-aware dictionary with samples from both source and target domains where each target sample is assigned a (pseudo) category label based on the category priors of source samples. This allows category contrastive learning (between target queries and the category-level dictionary) for category-discriminative yet domain-invariant feature representations: samples of the same category (from either source or target domain) are pulled closer while those of different categories are pushed apart simultaneously. Extensive UDA experiments in multiple visual tasks (e.g., segmentation, classification and detection) show that CaCo achieves superior performance as compared with state-of-the-art methods. The experiments also demonstrate that CaCo is complementary to existing UDA methods and generalizable to other learning setups such as unsupervised model adaptation, open-/partial-set adaptation etc.

1. Introduction

Though deep neural networks (DNNs) [20,57] have revolutionized various computer vision tasks [4, 20, 47, 57], they generally perform not well on new domains due to the cross-domain mismatch. Unsupervised domain adaptation (UDA) aims to mitigate the cross-domain mismatch via exploiting unlabelled target-domain samples. To achieve this purpose, researchers have designed different unsupervised training objectives on target-domain samples to train a well-performed model in target domain [7,30,40,59,62,63]. The existing unsupervised losses can be broadly classified into three categories: 1) adversarial loss that enforces source-like target representations [38,40,53,59,60,62,63]; 2) image translation loss that translates source images to have target-like styles and appearance [8,27,36,72,74]; and 3) self-training loss that re-trains networks iteratively with confidently pseudo-labelled target samples [15,36,80,81].

Unsupervised representation learning [5,19,41,44,58,68,73,77,78] addresses a related problem, i.e., unsupervised network pre-training which aims to learn discriminative embeddings from unlabelled data. In recent years, instance contrastive learning [5,19,42,58,68,73] has led to major advances in unsupervised representation learning. Despite different motivations, instance contrast methods can be thought of as a dictionary look-up task [19] that trains a visual encoder by matching an encoded query \( q \) with a dictionary of encoded keys \( k \): the encoded query should be similar to the encoded positive keys and dissimilar to encoded negative keys. With no labels available for unlabelled data, the positive keys are often randomly augmented versions of query samples, and all other samples are considered as negative keys.

In this work, we explore the idea of instance contrast in UDA. Considering contrastive learning as a dictionary look-up task, we hypothesize that a UDA dictionary should be category-aware and domain-mixed with keys from both source and target domains. Intuitively, a category-aware dictionary with category-balanced keys will encourage to learn category-discriminative yet category-unbiased representations, while the keys from both source and target domains will allow to learn invariant representations within and across the two domains, both being aligned with the objective of UDA.

With above motivation, this paper presents Category Contrast (CaCo) as a way of building category-aware and domain-mixed dictionaries with corresponding contrastive losses for UDA. As shown in Fig. 1, this dictionary includes keys that are evenly sampled in both categories and domains, where each target key comes with a predicted pseudo category. Take the illustrative dictionary \( K = \{ k_{cm} \} \) as an example. Each category \( c \) will have \( M \) keys while each domain has \( (C \times M)/2 \) keys.
The network learning will thus strive to minimize a category contrastive loss $L_{\text{CatNCE}}$ between target queries and dictionary keys: samples of the same category are pulled close while those of different categories are pushed away. This naturally leads to category-discriminative yet domain-invariant representations that perfectly match the objective of UDA.

With the category-aware and domain-mixed dictionary together with the category contrastive loss, the proposed Category Contrast tackles the UDA challenges with three desirable features: 1) It concurrently minimizes the intra-category variation and maximizes the inter-category distance with the category-aware dictionary design; 2) It achieves inter-domain and intra-domain alignment simultaneously thanks to the domain-mixed dictionary design by including both source and target samples; 3) It greatly mitigates the data balance issue due to the category-balanced dictionary design which allows to compute contrast losses evenly across all categories during learning.

We summarize the contributions of this paper as follows: (1) we explore instance contrast for UDA, aiming to learn discriminative representation for unlabelled target-domain samples. (2) we propose Category Contrast that builds a category-aware and domain-mixed dictionary with a category contrastive loss. It encourages to learn category-discriminative yet domain-invariant representation that perfectly matches the objective of UDA. (3) extensive experiments demonstrate that our CaCo achieves superior UDA performance consistently as compared with state-of-the-art. Additionally, CaCo complements previous UDA approaches and generalizes to other learning setups that involves unlabeled data.

## 2. Related Works

This work relates to two main fields of research, namely, unsupervised learning in unsupervised domain adaptation and instance contrast in unsupervised representation learning.

Unsupervised domain adaptation aims to leverage unlabelled target data to improve network performance in target domain. To learn from unlabelled target data, most existing works propose various unsupervised losses. We roughly sort them into three subcategories. The first subcategory is adversarial loss that enforces source-like target representation in terms of encoded features [7, 16, 38, 52, 62, 75], generated predictions [28, 40, 51, 53, 59] or converted latent representations [29, 60, 63]. The second category is image translation loss that generates source data with target-like styles and appearance via GANs [8, 10, 36] and spectrum matching [25, 72]. The third category is self-training loss that re-trains the network iteratively with pseudo-labelled target samples [14, 24, 26, 36, 64, 72, 80, 81].

We tackle UDA from a new perspective of instance contrastive learning, and propose a novel Category Contrast (CaCo) that introduces a generic category contrastive loss that can work for various UDA tasks. To the best of our knowledge, CaCo is the first effort to investigate instance contrastive learning for UDA.

**Instance Contrastive Learning** [5, 19, 42, 58, 68, 73] aims to learn an embedding space where positive samples are pulled close to an anchor and negative samples are pushed away. Despite different motivations, instance contrastive learning can be viewed as a dictionary look-up task [19] that trains a visual encoder by matching an encoded query $q$ with a dictionary of encoded keys $k$: $q$ should be similar to positive $k$ and dissimilar to negative $k$. Three
3. Method

3.1. Task Formulation

This work tackles the task of unsupervised domain adaptation, where labelled source-domain samples \( \{X_s, Y_s\} \) are accessible while only unlabelled data \( X_t \) are available in the target domain. The learning objective is to train a well-performing network \( G \) for \( X_t \). The baseline performance is acquired by training network \( G \) with annotated source-domain sample only:

\[
L_{\text{sup}} = l(G(X_s), Y_s),
\]

where \( l(\cdot) \) denotes an accuracy-related loss.

3.2. Preliminaries of Instance Contrastive Learning

The idea of instance contrastive learning [18] can be considered as training an encoder (feature extractor) for a dictionary look-up task. Given a query \( q \) and a dictionary that consists of a number of keys \( \{k_1, k_2, ..., k_N\} \), instance discriminative representations are learnt with an instance contrastive loss [18] (e.g., InfoNCE [42]), minimization of which will pull \( q \) close to its positive key and push it away from all other keys (considered negative for \( q \)).

\[
L_{\text{InfoNCE}} = \sum_{x_q \in X} - \log \frac{\sum_{i=0}^{N} \mathbb{I}(k_i \in q) \exp(q \cdot k_i / \tau)}{\sum_{i=0}^{N} \exp(q \cdot k_i / \tau)}
\]

where \( \mathbb{I}(k_i \in q) = 1 \) if \( k_i \) is the positive key of \( q \) and \( \mathbb{I}(k_i \in q) = 0 \) otherwise. Parameter \( \tau \) is a temperature parameter [68]. In general, the query representation is \( q = f_q(x^q) \) where \( f_q \) is an encoder network and \( x^q \) is a query sample (likewise in \( k = f_q(x^k) \)).

3.3. Category Contrast for Unsupervised Domain Adaptation

We tackle UDA from a perspective of instance contrastive learning. Specifically, we design Category Contrast that builds a category-aware and domain-mixed dictionary to learn category-discriminative yet domain-invariant representations under the guidance of a category contrastive loss.

**Overview.** For supervised training over a labelled source domain, we feed source samples \( \{X_s, Y_s\} \) to a model \( G \) and optimize \( G \) with Eq. 1. In this work, \( G \) consists of an encoder \( f_q \) and a classifier \( h \) that classifies the encoded embeddings into pre-defined categories, i.e., \( G(\cdot) = h(f_q(\cdot)) \). For unsupervised training over an unlabelled target domain, the training involves a query encoder \( f_q \) and a key momentum encoder \( f_k \) (the momentum update of \( f_q \), i.e., \( \theta_{f_k} = b \theta_{f_k} + (1 - b) \theta_{f_q} \), and \( b \) is a momentum coefficient) as illustrated in Fig. 1. During the training, we evenly sample the key \( x_k \) from both source and target domains (i.e., \( X_s \) and \( X_t \)) and feed them to the key encoder \( f_k \) to build a category-aware dictionary \( K \). We sample query \( x_q \) from the target domain (i.e., \( X_t \)) only and feed them to the query encoder \( f_q \) for category contrastive learning with the category-aware dictionary \( K \).

**Categorical domain-mixed dictionary.** One key component in the proposed CaCo is a category-aware and domain-mixed dictionary with keys from both source and target domains. The dictionary allows to perform category contrastive learning: the embeddings of the same category are pulled close together while those of different categories are pushed apart. The category awareness encourages the network to learn category-discriminative embeddings. This feature is critical to various visual tasks (e.g., segmentation, classification and detection) that require to learn discriminative features and classify them to pre-defined categories. In addition, the dictionary is domain-mixed which encourages to learn invariant representations within and across domains as category contrast is computed between target queries and keys from both source and target domains.
As stated in the Overview, given an encoded key \( k = f_k(x_k) \) \((x_k \in X_u \cup X_v)\), the classifier \( h \) predicts a category label \( \hat{y}_k \) and converts \( k \) into a categorical key \( k^c \) which is further queued into the categorical dictionary \( K \). These processes are carried out in parallel for a mini-batch of inputs, and the formal definition of the categorical dictionary \( K \) is presented in Definition 1.

**Definition 1** A Categorical Dictionary \( K \) with \( C \)-category is defined by:

\[
K = \{ k^1, k^2, \ldots, k^C \},
\]

where the categorical key \( k^c \in K \) is defined as the key \( k \) that belongs to the \( c \)-th semantic category \( (c = \arg \max_y \hat{y}_k^{(c)} ) \) and the predicted category label \( \hat{y}_k \) of \( k = f_k(x_k) \) is derived by:

\[
\arg \max_{\hat{y}_k} \sum_{c=1}^{C} \hat{y}_k^{(c)} \log p(c; k, \theta_h), \text{ s.t. } \hat{y}_k \in \Delta^C, \forall k,
\]

where \( h \) is the category classifier that predicts \( C \)-category probabilities for each embedding \((\text{e.g., } k)\), and \( \hat{y} = (\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(C)}) \) is the predicted category label. The key \( x_k \) is sampled from a training dataset \( X \) and encoded by the momentum encoder \( f_k \) to get the encoded key \( k = f_k(x_k) \). \( \Delta^C \) denotes a probability simplex, with which a point can be represented by \( C \) non-negative numbers that add up to 1.

**Remark 1** It is worth highlighting that Eq. 3 only shows one group of categorical keys for the simplicity of illustration and theoretical proof. In practice, we take the same strategy as [19] and maintain a dynamic categorical dictionary with \( M \)-size queue \((\text{i.e., } \{ k_m^c \}_{1 \leq c \leq C, 1 \leq m \leq M} )\), where the categorical keys are progressively updated in a category-wise manner. Specifically, for the queue of each category, we have \( \{ k_m^1, k_m^2, \ldots, k_M^C \} \), in which the oldest key is dequeued and the currently sampled key (belongs to \( c \)-th semantic category) is enqued.

**Category contrastive loss.** Given the categorical dictionary \( K = \{ k_m^c \}_{1 \leq c \leq C, 1 \leq m \leq M} \) defined in Definition 1, the proposed CaCo performs contrastive learning on unlabeled target data \( X_t \) via a category contrastive loss CatNCE that is defined by:

\[
\mathcal{L}_{\text{CatNCE}} = \sum_{x_q \in X_t} \left( \frac{1}{M} \sum_{m=1}^{M} \log \frac{\sum_{c=1}^{C} \exp(q \cdot k_m^c/\tau_m^c)(\hat{y}_q \times \hat{y}_k^{c_m})}{\sum_{c=1}^{C} \exp(q \cdot k_m^c/\tau_m^c)} \right),
\]

where \( q = f_q(x_q) \), \( (\hat{y}_q \times \hat{y}_k^{c_m}) \) is equal to 1 if both refer to the same category and 0 otherwise, \( \tau_m^c \) is a temperature hyper-parameter and the \( \cdot \) denotes the inner (dot) product. For each group of categorical keys \( \{ k_m^1, k_m^2, \ldots, k_m^C \} \), only one key is positive for the current query \( q \) (i.e., \( (\hat{y}_q \times \hat{y}_k^{c_m}) = 1 \)) as every sample belongs to a single category. This loss is thus the log loss of a \( C \)-way softmax-based classifier that strives to classify \( q \) as the positive key (of same category).

**Remark 2** Note that the CatNCE loss in Eq.5 has a similar form as the InfoNCE loss in Eq.2. Therefore, InfoNCE can be interpreted as a special case of CatNCE, where each instance (with its augmentations) itself is a category and the temperature is fixed \((\text{i.e., } \tau_m^c = \tau, \forall c, m)\). For CaCo, we assign different temperatures to different keys as their predicted labels have different uncertainties, i.e., scaled by the prediction entropy \( \mathcal{H}(. \cdot) \). The adjustable temperature parameter has also been explored in [5, 17, 31].

**Remark 3** Note that our category contrastive loss serves as an unsupervised objective function for training the encoder networks that represent the queries and keys [18]. In general, the query representation is \( q = f_q(x^q) \) where \( f_q \) is an encoder network and \( x^q \) is a query sample (likewise, \( k = f_k(x_k) \)). Their instantiations depend on the specific pretext task. The input \( x^q \) and \( x_k \) can be images [18,68,73], patches [42] or context consisting of a set of patches [42], etc. The networks \( f_q \) and \( f_k \) can be identical [18,66,73], partially shared [2,42], or different [19,58].

**Relations to existing instance contrast methods.** Beyond instance-discriminative representations as learnt by instance contrast [5, 19,42,58,68,73], CaCo learns category-discriminative yet domain-invariant representation.

**3.4. Theoretical Insights**

The category contrast (CaCo) is inherently connected with some probabilistic models. Specifically, CaCo can be modeled as an example of Expectation Maximization (EM):

**Proposition 1** The category contrastive learning can be modeled as a maximum likelihood (ML) problem optimized via Expectation Maximization (EM).

**Proposition 2** The categorical contrastive learning is convergent under certain conditions.

The proofs of Propositions 1 and 2 are provided in the Appendix.

**4. Experiments**

This section presents experimental results. Sections 4.1 and 4.2 describe the dataset and implementation details. Sections 4.3, 4.4 and 4.5 present the UDA experiments in segmentation, detection and classification, respectively. Section 4.6 discusses different features of the proposed method.
Table 1. Results over unsupervised domain adaptive semantic segmentation task GTA5-to-Cityscapes: CaCo-S, CaCo-T and CaCo construct the category-aware dictionary by sampling key sample xk from the source dataset Xs only, the target dataset Xt only, and both datasets, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>Road</th>
<th>SW</th>
<th>Build</th>
<th>Wall</th>
<th>Fence</th>
<th>Pole</th>
<th>TL</th>
<th>TS</th>
<th>Veg.</th>
<th>Terrain</th>
<th>Sky</th>
<th>PR</th>
<th>Rider</th>
<th>Car</th>
<th>Truck</th>
<th>Bus</th>
<th>Train</th>
<th>Motor</th>
<th>Bike</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>75.8</td>
<td>16.8</td>
<td>77.2</td>
<td>12.5</td>
<td>21.0</td>
<td>25.5</td>
<td>30.1</td>
<td>20.1</td>
<td>81.3</td>
<td>24.6</td>
<td>70.3</td>
<td>53.8</td>
<td>26.4</td>
<td>49.9</td>
<td>17.2</td>
<td>25.9</td>
<td>6.5</td>
<td>25.3</td>
<td>36.0</td>
<td>36.6</td>
</tr>
<tr>
<td>CaCo-S</td>
<td>91.1</td>
<td>54.4</td>
<td>79.6</td>
<td>23.4</td>
<td>23.3</td>
<td>23.9</td>
<td>35.2</td>
<td>14.8</td>
<td>83.4</td>
<td>33.3</td>
<td>73.6</td>
<td>58.5</td>
<td>27.6</td>
<td>73.7</td>
<td>32.5</td>
<td>35.4</td>
<td>3.9</td>
<td>30.1</td>
<td>28.1</td>
<td>42.4</td>
</tr>
<tr>
<td>CaCo-T</td>
<td>92.0</td>
<td>53.5</td>
<td>81.6</td>
<td>28.9</td>
<td>26.3</td>
<td>36.5</td>
<td>42.7</td>
<td>36.3</td>
<td>81.8</td>
<td>37.2</td>
<td>73.5</td>
<td>59.8</td>
<td>26.5</td>
<td>84.9</td>
<td>40.0</td>
<td>44.9</td>
<td>11.6</td>
<td>27.0</td>
<td>29.9</td>
<td>48.3</td>
</tr>
<tr>
<td>CaCo</td>
<td>91.9</td>
<td>54.3</td>
<td>82.7</td>
<td>31.7</td>
<td>25.0</td>
<td>38.1</td>
<td>46.7</td>
<td>39.2</td>
<td>82.6</td>
<td>39.7</td>
<td>76.2</td>
<td>63.5</td>
<td>23.6</td>
<td>85.1</td>
<td>38.6</td>
<td>47.8</td>
<td>10.3</td>
<td>23.4</td>
<td>35.1</td>
<td>49.2</td>
</tr>
</tbody>
</table>

Table 2. Results of unsupervised domain adaptive semantic segmentation task SYNTHIA-to-Cityscapes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Road</th>
<th>SW</th>
<th>Build</th>
<th>Wall</th>
<th>Fence</th>
<th>Pole</th>
<th>TL</th>
<th>TS</th>
<th>Veg.</th>
<th>Terrain</th>
<th>Sky</th>
<th>PR</th>
<th>Rider</th>
<th>Car</th>
<th>Truck</th>
<th>Bus</th>
<th>Train</th>
<th>Motor</th>
<th>Bike</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>55.6</td>
<td>23.8</td>
<td>74.6</td>
<td>9.2</td>
<td>0.2</td>
<td>24.4</td>
<td>6.1</td>
<td>12.1</td>
<td>74.8</td>
<td>79.0</td>
<td>55.3</td>
<td>19.1</td>
<td>39.6</td>
<td>23.3</td>
<td>13.7</td>
<td>25.0</td>
<td>33.5</td>
<td>38.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PatAlign [60]</td>
<td>82.4</td>
<td>38.0</td>
<td>78.6</td>
<td>8.7</td>
<td>0.6</td>
<td>26.0</td>
<td>3.9</td>
<td>11.1</td>
<td>75.5</td>
<td>84.6</td>
<td>53.5</td>
<td>21.6</td>
<td>71.4</td>
<td>32.6</td>
<td>19.3</td>
<td>31.7</td>
<td>40.0</td>
<td>46.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdapSeg [59]</td>
<td>84.3</td>
<td>42.7</td>
<td>77.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4.7</td>
<td>7.0</td>
<td>77.9</td>
<td>82.5</td>
<td>54.3</td>
<td>21.0</td>
<td>72.3</td>
<td>32.2</td>
<td>18.9</td>
<td>32.3</td>
<td>6.7</td>
<td>46.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLAN [81]</td>
<td>81.3</td>
<td>37.0</td>
<td>80.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>16.1</td>
<td>13.7</td>
<td>82.8</td>
<td>81.5</td>
<td>53.4</td>
<td>21.2</td>
<td>73.0</td>
<td>32.9</td>
<td>23.0</td>
<td>30.7</td>
<td>-</td>
<td>47.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AdvEnt [63]</td>
<td>85.6</td>
<td>42.2</td>
<td>79.7</td>
<td>8.7</td>
<td>0.4</td>
<td>25.9</td>
<td>5.4</td>
<td>8.1</td>
<td>80.4</td>
<td>84.1</td>
<td>57.9</td>
<td>23.8</td>
<td>73.3</td>
<td>36.4</td>
<td>14.2</td>
<td>33.0</td>
<td>41.2</td>
<td>48.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDA [43]</td>
<td>84.3</td>
<td>37.7</td>
<td>79.5</td>
<td>5.3</td>
<td>0.4</td>
<td>24.9</td>
<td>9.2</td>
<td>8.4</td>
<td>80.0</td>
<td>84.1</td>
<td>57.2</td>
<td>23.0</td>
<td>78.0</td>
<td>38.1</td>
<td>20.3</td>
<td>36.5</td>
<td>41.7</td>
<td>48.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoCDA [28]</td>
<td>86.2</td>
<td>44.9</td>
<td>79.5</td>
<td>8.3</td>
<td>0.7</td>
<td>27.8</td>
<td>9.4</td>
<td>11.8</td>
<td>78.6</td>
<td>86.5</td>
<td>57.2</td>
<td>26.1</td>
<td>76.8</td>
<td>39.9</td>
<td>21.5</td>
<td>32.1</td>
<td>42.9</td>
<td>50.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIR [32]</td>
<td>92.6</td>
<td>53.2</td>
<td>79.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.6</td>
<td>7.5</td>
<td>78.6</td>
<td>84.4</td>
<td>52.6</td>
<td>20.0</td>
<td>82.1</td>
<td>34.8</td>
<td>14.6</td>
<td>39.4</td>
<td>-</td>
<td>49.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIM [67]</td>
<td>89.1</td>
<td>47.0</td>
<td>84.8</td>
<td>43.3</td>
<td>28.7</td>
<td>31.6</td>
<td>35.0</td>
<td>17.6</td>
<td>84.7</td>
<td>43.3</td>
<td>85.3</td>
<td>57.0</td>
<td>31.5</td>
<td>83.8</td>
<td>42.6</td>
<td>48.5</td>
<td>1.9</td>
<td>30.4</td>
<td>49.2</td>
<td></td>
</tr>
<tr>
<td>+CaCo</td>
<td>92.9</td>
<td>55.0</td>
<td>88.3</td>
<td>34.2</td>
<td>31.1</td>
<td>34.9</td>
<td>40.7</td>
<td>34.0</td>
<td>85.2</td>
<td>40.1</td>
<td>87.1</td>
<td>61.0</td>
<td>31.2</td>
<td>85.6</td>
<td>32.3</td>
<td>43.9</td>
<td>0.3</td>
<td>36.4</td>
<td>51.0</td>
<td></td>
</tr>
</tbody>
</table>

4.1. Datasets

Adaptation for semantic segmentation: It involves three public datasets over two challenging UDA tasks, i.e., GTA5 [48]-to-Cityscapes [9] and SYNTHIA [49]-to-Cityscapes. Specifically, GTA5 is a synthesized dataset with 24,966 images and 19 common categories with Cityscapes. SYNTHIA is a synthesized dataset with 9,400 images and 16 common categories with Cityscapes. Cityscapes is a real-image dataset with 2975 training samples and 500 validation samples.

Adaptation for object detection: It involves three public datasets over two adaptation tasks, i.e., Cityscapes-to-Foggy Cityscapes [54] and Cityscapes-to-BDD [7]. Specifically, Foggy Cityscapes is a synthesized dataset that applies simulated fog on Cityscapes images. BDD is a real dataset with 70k samples in training set, 10k samples for validating and 7 common classes with Cityscapes dataset. As in [7, 52, 71], only a subset of BDD “daytime set” is used for experiments.

Adaptation for classification tasks: It involves two domain adaptive classification datasets VisDA17 [45] and Office-31 [50]. The former consists of a source domain with 152, 409 synthesized samples with twelve classes and a target domain with 55, 400 real samples. The latter consists of images of 31 categories which were collected from Amazon (2817 images), Webcam (795 images) and DSLR (498 images), respectively. The evaluation is on every pair of them as in [50, 55, 80].
### Table 3. Results over unsupervised domain adaptive object detection task Cityscapes-to-Foggy-Cityscapes.

<table>
<thead>
<tr>
<th>Method</th>
<th>person</th>
<th>rider</th>
<th>car</th>
<th>truck</th>
<th>bus</th>
<th>train</th>
<th>mcycle</th>
<th>bicycle</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [47]</td>
<td>24.4</td>
<td>30.5</td>
<td>32.6</td>
<td>10.8</td>
<td>25.4</td>
<td>9.1</td>
<td>15.2</td>
<td>28.3</td>
<td>22.0</td>
</tr>
<tr>
<td>MAF [21]</td>
<td>28.4</td>
<td>39.5</td>
<td>43.9</td>
<td>23.8</td>
<td>39.9</td>
<td>33.3</td>
<td>29.2</td>
<td>33.9</td>
<td>34.0</td>
</tr>
<tr>
<td>SCDA [79]</td>
<td>33.5</td>
<td>38.0</td>
<td>48.5</td>
<td>26.5</td>
<td>39.0</td>
<td>23.3</td>
<td>28.0</td>
<td>33.6</td>
<td>33.8</td>
</tr>
<tr>
<td>DA [7]</td>
<td>25.0</td>
<td>31.0</td>
<td>40.5</td>
<td>22.1</td>
<td>41.8</td>
<td>28.7</td>
<td>30.5</td>
<td>36.5</td>
<td>36.0</td>
</tr>
<tr>
<td>MLDA [70]</td>
<td>30.8</td>
<td>40.5</td>
<td>44.3</td>
<td>27.2</td>
<td>38.4</td>
<td>34.5</td>
<td>28.4</td>
<td>32.2</td>
<td>34.6</td>
</tr>
<tr>
<td>DMA [33]</td>
<td>41.9</td>
<td>38.7</td>
<td>56.7</td>
<td>22.6</td>
<td>41.5</td>
<td>26.8</td>
<td>35.5</td>
<td>36.0</td>
<td>33.8</td>
</tr>
<tr>
<td>CAFA [23]</td>
<td>36.2</td>
<td>35.3</td>
<td>43.5</td>
<td>30.0</td>
<td>29.9</td>
<td>42.3</td>
<td>32.2</td>
<td>34.0</td>
<td>36.0</td>
</tr>
<tr>
<td>SWDA [52]</td>
<td>32.9</td>
<td>31.3</td>
<td>46.3</td>
<td>19.5</td>
<td>18.9</td>
<td>17.3</td>
<td>23.8</td>
<td>26.9</td>
<td>37.4</td>
</tr>
<tr>
<td>+CaCo</td>
<td>39.3</td>
<td>46.1</td>
<td>48.0</td>
<td>32.4</td>
<td>45.7</td>
<td>38.7</td>
<td>31.3</td>
<td>35.3</td>
<td>39.6</td>
</tr>
<tr>
<td>CRDA [71]</td>
<td>32.9</td>
<td>43.8</td>
<td>49.2</td>
<td>27.2</td>
<td>45.1</td>
<td>36.4</td>
<td>30.3</td>
<td>34.6</td>
<td>37.4</td>
</tr>
<tr>
<td>+CaCo</td>
<td>39.4</td>
<td>47.4</td>
<td>47.9</td>
<td>32.5</td>
<td>46.4</td>
<td>39.9</td>
<td>32.7</td>
<td>35.4</td>
<td>40.2</td>
</tr>
<tr>
<td>CaCo</td>
<td>38.3</td>
<td>46.7</td>
<td>48.1</td>
<td>33.2</td>
<td>45.9</td>
<td>37.6</td>
<td>31.0</td>
<td>33.0</td>
<td>39.2</td>
</tr>
</tbody>
</table>

### Table 4. Results over unsupervised domain adaptive object detection task Cityscapes-to-BDD.

<table>
<thead>
<tr>
<th>Method</th>
<th>person</th>
<th>rider</th>
<th>car</th>
<th>truck</th>
<th>bus</th>
<th>train</th>
<th>mcycle</th>
<th>bicycle</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [47]</td>
<td>26.9</td>
<td>22.1</td>
<td>44.7</td>
<td>17.4</td>
<td>16.7</td>
<td>17.1</td>
<td>18.8</td>
<td>23.4</td>
<td>24.0</td>
</tr>
<tr>
<td>DA [7]</td>
<td>29.4</td>
<td>26.5</td>
<td>44.6</td>
<td>14.3</td>
<td>16.8</td>
<td>15.8</td>
<td>20.6</td>
<td>25.3</td>
<td>29.5</td>
</tr>
<tr>
<td>SWDA [52]</td>
<td>30.2</td>
<td>29.5</td>
<td>45.7</td>
<td>15.2</td>
<td>18.4</td>
<td>17.1</td>
<td>21.2</td>
<td>25.6</td>
<td>29.5</td>
</tr>
<tr>
<td>+CaCo</td>
<td>32.1</td>
<td>32.9</td>
<td>51.6</td>
<td>20.5</td>
<td>23.7</td>
<td>20.1</td>
<td>25.6</td>
<td>30.2</td>
<td>32.7</td>
</tr>
<tr>
<td>CRDA [71]</td>
<td>31.4</td>
<td>31.3</td>
<td>46.3</td>
<td>19.5</td>
<td>18.9</td>
<td>17.3</td>
<td>23.8</td>
<td>26.9</td>
<td>32.5</td>
</tr>
<tr>
<td>+CaCo</td>
<td>32.5</td>
<td>34.1</td>
<td>51.1</td>
<td>21.6</td>
<td>25.1</td>
<td>20.5</td>
<td>26.5</td>
<td>30.2</td>
<td>32.5</td>
</tr>
<tr>
<td>CaCo</td>
<td>32.7</td>
<td>32.2</td>
<td>50.6</td>
<td>20.2</td>
<td>23.5</td>
<td>19.4</td>
<td>25.0</td>
<td>29.1</td>
<td>32.5</td>
</tr>
</tbody>
</table>

### 4.2. Experiment Detail

**Segmentation Task:** As in [59, 81], DeepLabV2 [4] is employed as segmentation architecture and ResNet-101 [20] is adopted as the backbone. We employ SGD [3] as the optimizer with momentum 0.9, weight decay 0.0001 and learning rate 0.00025. We follow previous works [59, 81] to schedule the learning rate [4].

**Detection Task:** We follow previous works [7, 34, 52, 71] to conduct experiments, where VGG16-based [57] Faster R-CNN [47] is employed base detecting backbone. For network optimization, Stochastic gradient descent optimizer [3] is adopted with a momentum of 0.9 and a weight decay of 0.0005. The shorter side of input image is set to 600 and RoIAlign is employed for feature extraction. The learning rate is set as 0.001 for 50,000 training iterations and adjusted as 0.0001 in following 20,000 training iterations [7, 52, 71].

**Classification Task:** Following [50, 55, 80], we employ ResNet101 (for VisDA17 dataset) and ResNet50 [20] (for Office-31 dataset) as the base backbones. For optimization, Stochastic gradient descent optimizer [3] is employed with momentum 0.9, weight decay 0.0005, learning rate 0.001 and batch size 32.

We set the length of dictionary queue $M$ at 100 in all experiments except in parameter analysis. In addition, we set the momentum update coefficient $b$ at 0.999 and the basic temperature $\tau$ at 0.07 as in [19].

### 4.3. UDA for Semantic Segmentation

Table 1 reports semantic segmentation results on the task GTA5-to-Cityscapes. It can be seen that the proposed CaCo achieves comparable performance with state-of-the-art methods. In addition, CaCo is complementary to existing UDA approaches that exploit adversarial loss, image translation loss and self-training loss. As shown in Table 1, incorporating CaCo as denoted by “+CaCo” boosts the performance of state-of-the-art methods clearly and consistently. Fig. 2 presents the qualitative comparisons.

**Ablation studies.** We perform ablation studies over a widely adopted Baseline [20] as shown on the top of Table 1, where CaCo-S, CaCo-T and CaCo mean that the category-aware dictionary is built with keys from source domain, target domain and both domains, respectively. We can observe that CaCo-S and CaCo-T outperform the Baseline clearly. CaCo-S and CaCo-T provide orthogonal self-supervision signals, where CaCo-S focuses on inter-domain category contrastive learning between target samples and source keys and CaCo-T focuses on intra-domain category contrastive learning between target samples and target keys. In addition, CaCo performs clearly the best, showing that the keys from the source and target domains are complementary.

Table 2 reports semantic segmentation results on the
Table 5. Results over UDA-based classification benchmark VisDA17.

<table>
<thead>
<tr>
<th>Method</th>
<th>Aero</th>
<th>Bike</th>
<th>Bus</th>
<th>Car</th>
<th>Horse</th>
<th>Knife</th>
<th>Motor</th>
<th>Person</th>
<th>Plant</th>
<th>Skateboard</th>
<th>Train</th>
<th>Truck</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [20]</td>
<td>55.1</td>
<td>53.3</td>
<td>61.9</td>
<td>59.1</td>
<td>80.6</td>
<td>17.9</td>
<td>79.7</td>
<td>31.2</td>
<td>81.0</td>
<td>26.5</td>
<td>73.5</td>
<td>8.5</td>
<td>52.4</td>
</tr>
<tr>
<td>MMD [37]</td>
<td>87.1</td>
<td>63.0</td>
<td>76.5</td>
<td>42.0</td>
<td>90.3</td>
<td>42.9</td>
<td>85.9</td>
<td>53.1</td>
<td>49.7</td>
<td>36.3</td>
<td>85.8</td>
<td>20.7</td>
<td>61.1</td>
</tr>
<tr>
<td>DANN [11]</td>
<td>81.9</td>
<td>77.7</td>
<td>82.8</td>
<td>44.3</td>
<td>81.2</td>
<td>29.5</td>
<td>65.1</td>
<td>28.6</td>
<td>51.9</td>
<td>54.6</td>
<td>82.8</td>
<td>7.8</td>
<td>57.4</td>
</tr>
<tr>
<td>ENT [13]</td>
<td>80.3</td>
<td>75.5</td>
<td>75.8</td>
<td>48.3</td>
<td>77.9</td>
<td>27.3</td>
<td>69.7</td>
<td>40.2</td>
<td>46.5</td>
<td>46.6</td>
<td>79.3</td>
<td>16.0</td>
<td>57.0</td>
</tr>
<tr>
<td>MCD [53]</td>
<td>87.0</td>
<td>60.9</td>
<td>83.7</td>
<td>64.0</td>
<td>89.2</td>
<td>61.8</td>
<td>88.9</td>
<td>76.9</td>
<td>88.6</td>
<td>40.3</td>
<td>83.0</td>
<td>25.8</td>
<td>71.9</td>
</tr>
<tr>
<td>ADR [51]</td>
<td>87.8</td>
<td>79.5</td>
<td>83.7</td>
<td>53.2</td>
<td>73.2</td>
<td>87.8</td>
<td>60.0</td>
<td>85.5</td>
<td>32.3</td>
<td>74.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SimNet-Res152 [46]</td>
<td>94.3</td>
<td>82.3</td>
<td>73.5</td>
<td>47.2</td>
<td>87.9</td>
<td>49.2</td>
<td>75.1</td>
<td>79.7</td>
<td>85.3</td>
<td>68.5</td>
<td>81.1</td>
<td>50.3</td>
<td>72.9</td>
</tr>
<tr>
<td>GTA-Res152 [55]</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>CBST [81] +CaCo</td>
<td>90.4</td>
<td>80.7</td>
<td>78.8</td>
<td>57.0</td>
<td>88.9</td>
<td>87.0</td>
<td>81.3</td>
<td>79.4</td>
<td>88.7</td>
<td>86.8</td>
<td>86.8</td>
<td>63.9</td>
<td>80.9</td>
</tr>
</tbody>
</table>

Table 6. Results over domain adaptive image classification task Office-31.

<table>
<thead>
<tr>
<th>Method</th>
<th>A→W</th>
<th>D→-W</th>
<th>W→-D</th>
<th>A→-D</th>
<th>D→-A</th>
<th>W→-A</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [20]</td>
<td>68.4</td>
<td>96.7</td>
<td>99.3</td>
<td>68.9</td>
<td>62.5</td>
<td>60.7</td>
<td>76.1</td>
</tr>
<tr>
<td>DAN [57]</td>
<td>80.5</td>
<td>97.1</td>
<td>99.6</td>
<td>78.6</td>
<td>63.6</td>
<td>62.8</td>
<td>80.4</td>
</tr>
<tr>
<td>RTN [38]</td>
<td>84.5</td>
<td>96.8</td>
<td>99.4</td>
<td>77.5</td>
<td>66.2</td>
<td>64.8</td>
<td>81.6</td>
</tr>
<tr>
<td>DANN [11]</td>
<td>82.0</td>
<td>96.9</td>
<td>99.1</td>
<td>79.7</td>
<td>68.2</td>
<td>67.4</td>
<td>82.2</td>
</tr>
<tr>
<td>ADDA [62]</td>
<td>86.2</td>
<td>96.2</td>
<td>98.4</td>
<td>77.8</td>
<td>69.5</td>
<td>68.9</td>
<td>82.9</td>
</tr>
<tr>
<td>JAN [39]</td>
<td>85.4</td>
<td>97.4</td>
<td>99.8</td>
<td>84.7</td>
<td>68.6</td>
<td>70.0</td>
<td>84.3</td>
</tr>
<tr>
<td>GTA [55]</td>
<td>89.5</td>
<td>97.9</td>
<td>99.8</td>
<td>87.7</td>
<td>72.8</td>
<td>71.4</td>
<td>86.5</td>
</tr>
<tr>
<td>CBST [81] +CaCo</td>
<td>87.8</td>
<td>98.5</td>
<td>100.0</td>
<td>86.5</td>
<td>71.2</td>
<td>70.9</td>
<td>85.8</td>
</tr>
<tr>
<td>CRST [80] +CaCo</td>
<td>89.4</td>
<td>98.9</td>
<td>100.0</td>
<td>88.7</td>
<td>72.6</td>
<td>70.9</td>
<td>86.8</td>
</tr>
<tr>
<td>CaCo</td>
<td>89.7</td>
<td>98.4</td>
<td>100.0</td>
<td>91.7</td>
<td>73.1</td>
<td>72.8</td>
<td>87.6</td>
</tr>
</tbody>
</table>

4.6. Discussion and Analysis

Generalization ability: We investigate the generalization of the proposed CaCo via assessing it on several cornerstone visual UDA applications, i.e., segmentation, detection and classification. We present the experimental results in Tables 1-6, which demonstrate CaCo generates comparable performance consistently.

Complementariness ability: We investigate the synergistic benefits of our CaCo by combining it with existing UDA approaches. We present experimental results in Tables 1-6 (the rows with ‘+CaCo’), which show that CaCo when incorporated improves all existing methods consistently across different visual tasks.

Comparisons with existing unsupervised representation learning methods: We compared CaCo with unsupervised representation learning methods over the UDA task. Most existing methods achieve unsupervised representation learning through certain pretext tasks, such as instance contrastive learning [2, 5, 6, 18, 19, 22, 42, 68, 73], patch ordering [41], rotation prediction [12], and denoising/context/colorization auto-encoders [44, 77, 78]. The experiments (shown in Appendix) over the UDA task GTA→Cityscapes show that existing unsupervised representation learning works not well on UDA tasks. The main reason lies in that these approaches are designed for learning instance-discriminative representations without considering semantic priors and domain gaps. CaCo also performs unsupervised learning but works for UDA effectively, largely because it learns category-discriminative yet domain-invariant representations which is essential to various visual UDA tasks.

Parameter studies: The parameter M (in the proposed CaCo) controls the length (or size) of the categorical dictionary. We investigate M via varying it from 50 to 150 progressively. The experiments (shown in Appendix) over the
UDA segmentation task GTA-to-Cityscapes demonstrate that $M$ does not affect UDA clearly while it changes from 50 to 150.

**Generalization across different learning setups:** We studied the scalability of the proposed CaCo from the view of learning setups. Specifically, we evaluated CaCo over a variety of tasks that involve unlabeled data learning and certain semantic priors such as unsupervised model adaptation, and partial-set/open-set UDA. We present the experimental results in Appendix, which illustrates that CaCo generates comparable performance robustly.

**Category-aware dictionary:** We studied three variant designs of the proposed category-aware dictionary: 1) Assign all keys with the same temperature; 2) Using two individual dictionaries (for source and target data) instead of a single domain-mixed dictionary; 3) Update the dictionary by memory bank [68] or current mini-batch [5]. Experiments (in Appendix) verify the superiority of the design as described in this paper.

5. Conclusion

This paper presents CaCo, a category contrast technique that introduces a generic category contrastive loss that can work for various visual UDA tasks effectively. We construct a semantics-aware dictionary with samples from both source and target domains where each target sample is assigned a (pseudo) category label based on the category priors of source samples. This allows category contrastive learning (between target queries and the category-level dictionary) for category-discriminative yet domain-invariant feature representations: samples of the same category (from either source or target domain) are pulled close together while those of different categories are pushed away simultaneously. Extensive experiments over multiple visual tasks (e.g., segmentation, classification and detection) show that the simple implementation of CaCo achieves superior performance as compared with state-of-the-art methods. In addition, we demonstrate that CaCo is also complementary to existing UDA methods and generalizable to other learning setups such as unsupervised model adaptation, open/partial-set adaptation etc.

Acknowledgement

This research was conducted at Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU), which is a collaboration between Singapore Telecommunications Limited (Singtel) and Nanyang Technological University (NTU) that is supported by A*STAR under its Industry Alignment Fund (LOA Award number: I1701E0013).
References


1213
