Zero-Shot Deep Domain Adaptation

Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst

Siemens Corporate Technology, Princeton NJ 08540, USA
{kuanchuan.peng, ziyan.wu, jan.ernst}@siemens.com

Abstract. Domain adaptation is an important tool to transfer knowledge about a task (e.g. classification) learned in a source domain to a second, or target domain. Current approaches assume that task-relevant target-domain data is available during training. We demonstrate how to perform domain adaptation when no such task-relevant target-domain data is available. To tackle this issue, we propose *zero-shot deep domain adaptation* (ZDDA), which uses privileged information from *task-irrelevant dual-domain pairs*. ZDDA learns a source-domain representation which is not only tailored for the task of interest but also close to the target-domain representation. Therefore, the source-domain task of interest solution (e.g. a classifier for classification tasks) which is jointly trained with the source-domain representation can be applicable to both the source and target representations. Using the MNIST, Fashion-MNIST, NIST, EMNIST, and SUN RGB-D datasets, we show that ZDDA can perform domain adaptation in classification tasks without access to task-relevant target-domain training data. We also extend ZDDA to perform sensor fusion in the SUN RGB-D scene classification task by simulating task-relevant target-domain representations with task-relevant source-domain data. To the best of our knowledge, ZDDA is the first domain adaptation and sensor fusion method which requires no task-relevant target-domain data. The underlying principle is not particular to computer vision data, but should be extensible to other domains.

Keywords: zero-shot · domain adaptation · sensor fusion

1 Introduction

The useful information to solve practical tasks often exists in different domains captured by various sensors, where a domain can be either a modality or a dataset. For instance, the 3-D layout of a room can be either captured by a depth sensor or inferred from the RGB images. In real-world scenarios, it is highly likely that we can only access limited amount of data in certain domain(s). The performance of the solution (e.g. the classifier for classification tasks) we learn from one domain often degrades when the same solution is applied to other domains, which is caused by domain shift [17] in a typical domain adaptation (DA) task, where source-domain training data, target-domain training data, and a task of interest (TOI) are given. The goal of a DA task is to derive solution(s) of the TOI for both the source and target domains.

Fig. 1. We propose zero-shot deep domain adaptation (ZDDA) for domain adaptation and sensor fusion. ZDDA learns from the task-irrelevant dual-domain pairs when the task-relevant target-domain training data is unavailable. In this example domain adaptation task (MNIST [27]→MNIST-M [13]), the task-irrelevant gray-RGB pairs are from the Fashion-MNIST [46] dataset and the Fashion-MNIST-M dataset (the colored version of the Fashion-MNIST [46] dataset with the details in Sec. 4.1)

The state-of-the-art DA methods such as [1, 14–16, 25, 30, 35, 37, 39–41, 43, 44, 47, 50] are proposed to solve DA tasks under the assumption that the *task-relevant data*, the data directly applicable and related to the TOI (regardless of whether it is labeled or not), in the target domain is available at training time, which is not always true in practice. For instance, in real business use cases, acquiring the task-relevant target-domain training data can be infeasible due to the combination of the following reasons: 1) Unsuitable tools at the field. 2) Product development timeline. 3) Budget limitation. 4) Data import/export regulations. Such impractical assumption is also assumed true in the existing works of sensor fusion such as [31,48], where the goal is to obtain a dual-domain (source and target) TOI solution which is robust to noise in either domain. This unsolved issue motivates us to propose *zero-shot deep domain adaptation* (ZDDA), a DA and sensor fusion approach which learns from the task-irrelevant dual-domain training pairs without using the task-relevant target-domain training data, where we use the term *task-irrelevant data* to refer to the data which is not task-relevant. In the rest of the paper, we use T-R and T-I as the shorthand of task-relevant and task-irrelevant, respectively.

We illustrate what ZDDA is designed to achieve in Fig. 1 using an example DA task (MNIST [27]→MNIST-M [13]). We recommend that the readers view all the figures and tables in color. In Fig. 1, the source and target domains are gray scale and RGB images respectively, and the TOI is digit classification with both the MNIST [27] and MNIST-M [13] testing data. We assume that the MNIST-M [13] training data is unavailable. In this example, ZDDA aims at using the MNIST [27] training data and the T-I gray-RGB pairs from the Fashion-MNIST [46] dataset and the Fashion-MNIST-M dataset (the colored version of the Fashion-MNIST [46] dataset with the details in Sec. 4.1) to train digit classifiers for MNIST [27] and MNIST-M [13] images. Specifically, ZDDA achieves this by simulating the RGB representation using the gray scale image and building a joint network with the supervision of the TOI in the gray scale domain. We present the details of ZDDA in Sec. 3.

Table 1. Problem setting comparison between ZDDA, unsupervised domain adaptation (UDA), multi-view learning (MVL), and domain generalization (DG)

problem conditions	UDA	MVL	DG	ZDDA
given T-R target-domain training data?	Y	Y	N	N
given T-R training data in multiple (>1) domains/views?	N	Y	Y	N
example prior work	[33]	[42]	[28]	N/A

We make the following two contributions: (1) To the best of our knowledge, our proposed method, ZDDA, is the first deep learning based method performing domain adaptation between one source image modality and another different target image modality (not just different datasets in the same modality such as the Office dataset [32]) **without using the task-relevant target-domain training data**. We show ZDDA's efficacy using the MNIST [27], Fashion-MNIST [46], NIST [18], EMNIST [9], and SUN RGB-D [36] datasets with cross validation. (2) **Given no task-relevant target-domain training data**, we show that ZDDA can perform sensor fusion and that ZDDA is more robust to noisy testing data in either source or target or both domains compared with a naive fusion approach in the scene classification task from the SUN RGB-D [36] dataset.

2 Related Work

Domain adaptation (DA) has been extensively studied in computer vision and applied to various applications such as image classification [1, 14–16, 25, 30, 35, 37, 39–41, 43, 44, 47, 50], semantic segmentation [45, 51], and image captioning [8]. With the advance of deep neural networks in recent years, the state-of-the-art methods successfully perform DA with (fully or partially) labeled [8, 15, 25, 30, 39] or unlabeled [1, 14–16, 35, 37, 39–41, 43–45, 47, 50] T-R target-domain data. Although different strategies such as the domain adversarial loss [40] and the domain confusion loss [39] are proposed to improve the performance in the DA tasks, most of the existing methods need the T-R target-domain training data, which can be unavailable in reality. In contrast, we propose ZDDA to learn from the T-I dual-domain pairs without using the T-R target-domain training data. One part of ZDDA includes simulating the target-domain representation using the source-domain data, and similar concepts have been mentioned in [19, 21]. However, both of [19, 21] require the access to the T-R dual-domain training pairs, but ZDDA needs no T-R target-domain data.

Other problems related to ZDDA include unsupervised domain adaptation (UDA), multi-view learning (MVL), and domain generalization (DG), and we compare their problem settings in Table 1, which shows that the ZDDA problem setting is different from those of UDA, MVL, and DG. In UDA and MVL, T-R target-domain training data is given. In MVL and DG, T-R training data in multiple domains is given. However, in ZDDA, T-R target-domain training data

Table 2. Working condition comparison between ZDDA and other existing methods. Among all the listed methods, only ZDDA can work under all four conditions

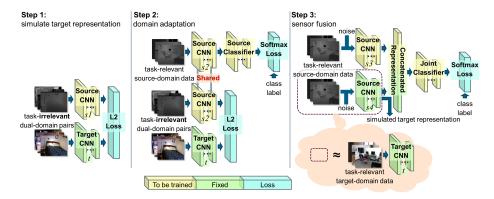| Can each method work under each condition?	[28]	[11]	[6]	[49]	[14]	[32]	[39]	ZDDA
without T-R target-domain training data	Y	N	N	Y	N	Y	N	Y
without T-R training data in >1 domains	N	Y	Y	Y	Y	Y	Y	Y
without accurate domain descriptor	Y	Y	Y	N	Y	Y	Y	Y
without class labels for **any** target domain data	Y	N	Y	Y	Y	N	N	Y
conjunction of all the above conditions	N	N	N	N	N	N	N	Y

is unavailable and the only available T-R training data is in one source domain. We further compare ZDDA with the existing methods relevant to our problem setting in Table 2, which shows that among the listed methods, only ZDDA can work under all four conditions.

In terms of sensor fusion, Ngiam et al. [31] define the three components for multimodal learning (multimodal fusion, cross modality learning, and shared representation learning) based on the modality used for feature learning, supervised training, and testing, and experiment on audio-video data with their proposed deep belief network and autoencoder based method. Targeting on the temporal data, Yang et al. [48] follow the setup of multimodal learning in [31], and validate their proposed encoder-decoder architecture using video-sensor and audio-video data. Although certain progress about sensor fusion is achieved in the previous works [31, 48], we are unaware of any existing sensor fusion method which overcomes the issue of lacking T-R target-domain training data, which is the issue that ZDDA is designed to solve.

3   Our Proposed Method — ZDDA

Given a task of interest (TOI), a source domain D_s, and a target domain D_t, our proposed method, zero-shot deep domain adaptation (ZDDA), is designed to achieve the following two goals: 1) **Domain adaptation**: Derive the solutions of the TOI for both D_s and D_t when the T-R training data in D_t is unavailable. We assume that we have access to the T-R labeled training data in D_s and the T-I dual-domain pairs in D_s and D_t. 2) **Sensor fusion**: Given the previous assumption, derive the solution of TOI when the testing data in both D_s and D_t is available. The testing data in either D_s or D_t can be noisy. We assume that there is no prior knowledge available about the type of noise and which domain gives noisy data at testing time.

For convenience, we use a scene classification task in RGB-D as an example TOI to explain ZDDA, but ZDDA can be applied to other TOIs/domains. In this example, D_s and D_t are depth and RGB images respectively. According to the our previous assumption, we have access to the T-R labeled depth data and T-I RGB-D pairs at training time. The training procedure of ZDDA is illustrated in
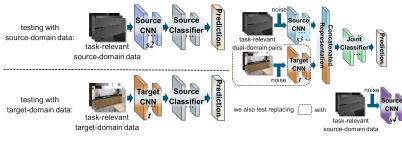
Fig. 2. An overview of the ZDDA training procedure. We use the images from the SUN RGB-D [36] dataset for illustration. ZDDA simulates the target-domain representation using the source-domain data, builds a joint network with the supervision from the source domain, and trains a sensor fusion network. In step 1, we choose to train $s1$ and fix t, but we can also train t and fix $s1$ to simulate the target-domain representation. In step 2, t can also be trainable instead of being fixed, but we choose to fix it to make the number of trainable parameters manageable. The details are explained in Sec. 3

Fig. 2, where we simulate the RGB representation using the depth image, build a joint network with the supervision of the TOI in depth images, and train a sensor fusion network in step 1, step 2, and step 3 respectively. We use the ID marked at the bottom of each convolutional neural networks (CNN) in Fig. 2 to refer to each CNN.

In step 1, we create two CNNs, $s1$ and t, to take the depth and RGB images of the T-I RGB-D pairs as input. The purpose of this step is to find $s1$ and t such that feeding the RGB image into t can be approximated by feeding the corresponding depth image into $s1$. We achieve this by fixing t and enforcing the L2 loss on top of $s1$ and t at training time. We choose to train $s1$ and fix t here, but training t and fixing $s1$ can also achieve the same purpose. The L2 loss can be replaced with any suitable loss functions which encourage the similarity of the two input representations, and our selection is inspired by [19, 21]. The design in step 1 is similar to the hallucination architecture [21] and the supervision transfer [19], but we require no T-R dual-domain training pairs. Instead, we use the T-I dual-domain training pairs.

After step 1, we add another CNN, $s2$ (with the same network architecture as that of $s1$), and a classifier to the network (as shown in step 2) to learn from the label of the training depth images. The classifier in our experiment is a fully connected layer for simplicity, but other types of classifiers can also be used. The newly added CNN takes the T-R depth images as input, and shares all the weights with the original source CNN, so we use $s2$ to refer to both of them. t is the same as that in step 1. At training time, we pre-train $s2$ from $s1$ and fix t. Our choice of fixing t is inspired by the adversarial adaptation step in ADDA [40]. t can also be trainable in step 2, but given our limited

(a) testing domain adaptation   **(b)** testing sensor fusion

Fig. 3. An overview of the ZDDA testing procedure. We use the SUN RGB-D [36] images for illustration. Different from the color coding in Fig. 2, the colors here are purely used to distinguish different CNNs/classifiers/predictions

amount of data, we choose to fix it to make the number of trainable parameters manageable. $s2$ and the source classifier are trained such that the weighted sum of the softmax loss and L2 loss are minimized. The softmax loss can be replaced with other losses suitable for the TOI.

After step 2, we expect to obtain a depth representation which is close to the RGB representation in the feature space and performs reasonably well with the trained classifier in the scene classification. Step 1 and step 2 can be done in one step with properly designed curriculum learning, but we separate them not only because of clarity but also because of the difficulty of designing the learning curriculum before training. After step 2, we can form the scene classifier in depth/RGB (denoted as C_D/C_{RGB}) by concatenating $s2/t$ and the trained source classifier (as shown in Fig. 3a), which meets our first goal, domain adaptation. We use the notation ZDDA$_2$ to refer to the method using the training procedure in Fig. 2 up to step 2 and the testing procedure in Fig. 3a.

To perform sensor fusion, we propose step 3, where we train a joint classifier for RGB-D input using only the T-R depth training data. We create two CNNs, $s3$ and $s4$ (each with the same network architecture as that of CNN_{s1}), and add a concatenation layer on top of them to concatenate their output representations. The concatenated representation is connected to a joint classifier. At training time, we pre-train $s3$ and $s4$ from $s2$ and $s1$ respectively and fix $s4$. Both $s3$ and $s4$ take the T-R depth images as the input. To train a more robust RGB-D scene classifier, we randomly select some inputs of $s3$ and $s4$, and optionally add noise to them independently. We supervise the entire network with the label of the depth training data for the scene classification, which is done by the softmax loss enforced on top of the joint classifier.

According to step 1, the output of $s4$ is expected to simulate the RGB representation as if we feed the T-R RGB image to t. This expectation is based on the assumption that the relationship between the dual-domain pairwise data is similar, regardless of whether the data is T-R or T-I. Given the simulated RGB representation, $s3$ is trained to learn a depth representation suitable for

Table 3. The statistics of the datasets we use. For NIST, we use the "by_class" dataset, remove the digits, and treat uppercase and lowercase letters as different classes. For EMNIST, we use the "EMNIST Letters" split which only contains the letters. We create the colored datasets from the original ones using Ganin's method [13] (see Sec. 4.1 for details). We refer to each dataset by the corresponding dataset ID (e.g. D_N and D_N-M refer to the NIST and the NIST-M datasets, respectively)

original dataset	MNIST [27]	Fashion-MNIST [46]	NIST [18]	EMNIST [9]	SUN RGB-D [36]
dataset ID	D_M	D_F	D_N	D_E	D_S
image content	digit	clothing	letter	letter	scene
image size	28×28	28×28	128×128	28×28	∼VGA
# classes	10	10	52	26	45
# training data	60000	60000	387361	124800	details in Sec. 4.1
# testing data	10000	10000	23941	20800	details in Sec. 4.1
class labels	0-9	dress, coat, etc.	A-Z, a-z	Aa-Zz	corridor, lab, etc.
balanced class?	N	Y	N	Y	N
example images					
colored dataset	MNIST-M	Fashion-MNIST-M	NIST-M	EMNIST-M	N / A
example images					N / A

the RGB-D scene classification without the constraint of the L2 loss in step 2. At testing time, $s4$ is replaced with t which takes the T-R RGB testing images as input with optional noise added to test the ZDDA's performance given noisy RGB-D testing data (as shown in Fig. 3b). In Fig. 3b, we also test replacing "RGB images and t" with "depth images and $s4$" to evaluate the performance of ZDDA in step 3 given only testing depth images. After the training procedure in Fig. 2, we can form three scene classifiers in RGB, depth, and RGB-D domains (one classifier per domain), and our trained RGB-D classifier is expected to be able to handle noisy input with reasonable performance degradation. The 3-step training procedure of ZDDA in Fig. 2 can be framed as an end-to-end training process with proper learning curriculum. We separate these 3 steps due to the ease of explanation. We use the notation ZDDA$_3$ to refer to the method using the training procedure in Fig. 2 up to step 3 and the testing procedure in Fig. 3b.

4 Experiment Setup

4.1 Datasets

For domain adaptation (DA), we validate the efficacy of ZDDA under classification tasks using the MNIST [27], Fashion-MNIST [46], NIST [18], EMNIST [9], and SUN RGB-D [36] datasets. For sensor fusion, we experiment on the SUN RGB-D [36] dataset. We summarize the statistics of these datasets in Table 3, where we list the dataset IDs which we use to refer to these datasets. For D_M, D_F, D_N, and D_E, we create the colored version of these datasets (D_M-M, D_F-M, D_N-M, and D_E-M) according to the procedure proposed in Ganin's work [13]

Table 4. The base network architecture (BNA) we use in our experiments. For each BNA, We specify the layer separating the source/target CNN and the source classifier in Fig. 2. The layer name in the right column is based on the official Caffe [24] and SqueezeNet_v1.1 [23] implementation of each BNA

base network architecture (BNA)	source/target CNN architecture (up to where in BNA (inclusive))
LeNet [5]	ip1
GoogleNet [38]	pool5/7x7_s1
AlexNet [26]	fc7
SqueezeNet_v1.1 [23]	fire9/concat

— blending the gray scale images with the patches randomly extracted from the BSDS500 dataset [2]. These colored datasets and the original ones are used to construct four DA tasks adapting from gray scale to RGB images. For each DA task, we use one of the other three pairs of the datasets (original and colored ones) as the T-I data. For example, for the DA task $D_M \to D_M$-M, D_F and D_F-M together are one possible choice as the T-I data. The DA task $D_M \to D_M$-M is acknowledged as one of the standard experiments to test the efficacy of the DA methods in recent works [1,7,14,20,33,34], so we adopt this experiment and extend it to D_F, D_N, and D_E.

D_S contains 10335 RGB-D pairs belonging to 45 different scenes. For each RGB-D pair, both the raw (noisy) depth image and post-processed clean depth image are provided, and we choose to use the raw depth image to simulate the real-world scenarios. Out of the 45 scenes, we select the following 10 scenes: computer room (0), conference room (1), corridor (2), dining room (3), discussion area (4), home office (5), idk (6), lab (7), lecture theatre (8), and study space (9), where the number after each scene is the scene ID we use to refer to each scene. The 8021 RGB-D pairs belonging to the other scenes are used as the T-I training data. The 10 scenes are selected based on the following two constraints: 1) Each scene contains at least 150 RGB-D pairs in D_S, which ensures a reasonable amount of T-R data. 2) The total number of the RGB-D pairs belonging to the selected 10 scenes is minimized, which maximizes the amount of the T-I training data. We empirically find that the amount and diversity of the T-I training data are important for ZDDA. To avoid the bias toward the scene with more data, for each of the selected 10 scenes, we randomly select 89/38 RGB-D pairs as the T-R training/testing data. When experimenting on different scene classification tasks using different selections of scenes, we only use the training/testing data associated with those selected scenes as the T-R data.

4.2   Training Details

We use Caffe [24] to implement ZDDA. Table 4 lists the base network architecture (BNA) we use and the layer separating the source/target CNN and the source

classifier in Fig. 2. For instance, in the case when the BNA is LeNet [5], the architecture of each source/target CNN in Fig. 2 is the LeNet [5] architecture up to the "ip1" layer, and the rest of the LeNet [5] architecture is used as the source classifier. For the DA tasks involving D_M, D_F, D_N, and D_E, we use the LeNet [5] as the BNA and train all the CNNs in Fig. 2 from scratch except that the target CNN is pre-trained from the T-I dataset and fixed afterwards. For example, when using D_F and D_F-M as the T-I data in the DA task $D_M \rightarrow D_M$-M, we train a CNN (denoted as CNN_{ref}) with the LeNet [5] architecture from scratch using the images and labels of D_F-M, and pre-train the target CNNs in Fig. 2 from CNN_{ref}. We follow similar procedures for other DA tasks and T-I datasets involving D_M, D_F, D_N, and D_E.

For the experiment involving D_S, we mostly use GoogleNet [38] as the BNA, but we also use AlexNet [26] and SqueezeNet_v1.1 [23] in the cross validation experiment with respect to different BNAs. Since only limited amount of RGB-D pairs are available in D_S, we pre-train all the CNNs in Fig. 2 from the BVLC GoogleNet model [4], BVLC AlexNet model [3], and the reference SqueezeNet model [22] when the BNA is GoogleNet [38], AlexNet [26], and SqueezeNet_v1.1 [23], respectively. These pre-trained models are trained for the ImageNet [10] classification task.

For the optionally added noise in ZDDA$_3$, we experiment on training/testing with noise-free data and noisy data. In the latter case, given that no prior knowledge about the noise is available, we use the black image as the noisy image to model the extreme case where no information in the noisy image is available. We train ZDDA$_3$ step 3 with the augmented training data formed by copying the original T-R source-domain training data 10 times and replacing $p_{train}\%$ of the images selected randomly with the black images. We follow this procedure twice independently and use the two augmented training datasets as the inputs of the two source CNNs in step 3. We empirically set $p_{train} = 20$. The testing data in Fig. 3b is constructed by replacing $p_{test}\%$ of the original testing images selected randomly with the black images, and we evaluate ZDDA under different p_{test}s. For all the experiments, the number of the output nodes of the source/joint classifiers is set to be the number of classes in the TOI, and these classifiers are trained from scratch. For the joint classifiers, we use two fully connected layers unless otherwise specified, where the first fully connected layer of the joint classifier has 1024 output nodes.

In terms of the training parameters used in Fig. 2 for the task involving D_S when the BNA is GoogleNet [38], we use a batch size of 32 and a fixed learning rate $10^{-5}/10^{-6}/10^{-3}$ for step 1/2/3. The learning rate is chosen such that the trained network can converge under a reasonable amount of time. We set the weight of the softmax loss and the L2 loss in step 2 to be 10^3 and 1 respectively such that both losses have comparable numerical values. Step 1/2/3 are trained for $10^4/10^3/10^3$ iterations. For the other training parameters, we adopt the default ones used in training the BVLC GoogleNet model [4] for the ImageNet [10] classification task unless otherwise specified. In general, we adopt the default training parameters used in training each BNA for either the MNIST [27]

Table 5. The overall / average per class accuracy (%) of the domain adaptation tasks (gray scale images → RGB images) formed by the datasets in Table 3, where we introduce the dataset IDs and use them to refer to the datasets here. The middle four rows show the performance of ZDDA$_2$. The color of each cell reflects the performance ranking in each column, where darker is better. The number in the parenthesis of the middle four rows is the semantic similarity between the T-R and T-I datasets measured by word2vec [29], where larger numbers represent higher semantic similarity. The T-R target-domain training data is only available for the row "target only"

T-I data	$D_M \to D_M$-M	$D_F \to D_F$-M	$D_N \to D_N$-M	$D_E \to D_E$-M
source only	39.04/39.31	33.77/33.77	8.59/8.79	33.70/33.70
D_M, D_M-M	N/A	51.55/51.55 (0.049)	34.25/33.35 (0.174)	71.20/71.20 (0.178)
D_F, D_F-M	73.15/72.96 (0.049)	N/A	21.93/21.24 (0.059)	46.93/46.93 (0.053)
D_N, D_N-M	91.99/92.00 (0.174)	43.87/43.87 (0.059)	N/A	N/A
D_E, D_E-M	94.84/94.82 (0.178)	65.30/65.30 (0.053)	N/A	N/A
target only	97.33/97.34	84.44/84.44	62.13/61.99	89.52/89.52

or ImageNet [10] classification tasks in the Caffe [24] and SqueezeNet_v1.1 [23] implementation unless otherwise specified.

4.3   Performance References and Baselines

To obtain the performance references of the fully supervised methods, we train a classifier with the BNA in Table 4 in each domain using the T-R training data and labels in that domain. When the BNA is LeNet [5], we train the classifier from scratch. For the other BNAs, we pre-train the classifier in the same way as that described in Sec. 4.2. After training, for each DA task, we get two fully supervised classifiers $C_{fs,s}$ and $C_{fs,t}$ in the source and target domains respectively. For the baseline of the DA task, we directly feed the target-domain testing images to $C_{fs,s}$ to obtain the performance without applying any DA method. For the baseline of sensor fusion, we compare ZDDA$_3$ with a naive fusion method by predicting the label with the highest probability from C_{RGB} and C_D in Sec. 3.

5   Experimental Result

We first compare ZDDA$_2$ with the baseline in four domain adaptation (DA) tasks (adapting from gray scale to RGB images) involving D_M, D_F, D_N, and D_E, and the result is summarized in Table 5, where the first two numbers represent the overall/average per class accuracy (%). Darker cells in each column represent better classification accuracy in each task. In Table 5, the middle four rows represent the performance of ZDDA$_2$. $\{D_N, D_N$-M$\}$ and $\{D_E, D_E$-M$\}$ cannot be the T-I data for each other because they are both directly related to the letter classification tasks. Table 5 shows that regardless of which T-I data we

Table 6. The performance comparison of the domain adaptation task MNIST→MNIST-M. The color of each cell reflects the performance ranking (darker is better). For ZDDA$_2$, we report the best overall accuracy from Table 5. **All the listed methods except ZDDA$_2$ use the MNIST-M training data.** Without the access to the MNIST-M training data, ZDDA$_2$ can still achieve the accuracy comparable to those of the competing methods (even outperform most of them) in this task

method	[14]	[34]	[20]	[33]	[7]	ZDDA$_2$
accuracy (%)	76.66	86.70	89.53	94.20	98.20	94.84

Table 7. Performance comparison with different numbers of classes in scene classification. The reported numbers are classification accuracy (%). The color of each cell reflects the performance ranking in each column, where darker color means better performance. $P_{\text{RGB-D}}$ represents the **task-irrelevant** RGB-D pairs

exp. ID	method	training modality	testing modality	number of classes								
				2	3	4	5	6	7	8	9	10
1	GoogleNet	D	D	85.53	83.33	82.89	70.00	67.11	59.02	54.28	50.88	51.84
2	ZDDA$_2$	D+$P_{\text{RGB-D}}$	D	88.16	85.96	83.55	77.89	70.18	66.92	64.80	62.28	59.74
3	ZDDA$_3$	D+$P_{\text{RGB-D}}$	D	88.16	86.84	84.87	77.89	72.37	66.92	64.47	64.33	63.16
4	GoogleNet	D	RGB	68.42	57.02	56.58	48.95	42.11	45.11	40.46	34.50	31.58
5	ZDDA$_2$	D+$P_{\text{RGB-D}}$	RGB	80.26	78.07	76.32	67.37	57.89	53.76	47.37	45.03	43.16
6	GoogleNet	RGB	RGB	88.16	85.09	84.87	79.47	78.07	68.80	70.07	69.88	63.68
7	ZDDA$_3$	D+$P_{\text{RGB-D}}$	RGB-D	88.16	85.96	85.53	76.32	72.81	68.42	65.13	63.16	63.16
selected scene IDs (defined in Sec. 4.1)				0∼1	0∼2	0∼3	0∼4	0∼5	0∼6	0∼7	0∼8	0∼9

use, ZDDA$_2$ significantly outperforms the baseline (source only). To see how the semantic similarity between the T-R dataset (denoted as D_{T-R}) and T-I dataset (denoted as D_{T-I}) affects the performance, we are inspired by [12] and use the word2vec [29] to compute the mean similarity (denoted as S) of any two labels from D_{T-R} and D_{T-I} (one from each). We report $S(D_{T-R}, D_{T-I})$ in the parenthesis of the middle four rows of Table 5, where higher S represents higher semantic similarity. Given Table 5 and the following reference S values: $S(\text{object, scene})=0.192$, $S(\text{animal, fruit})=0.171$, and $S(\text{cat, dog})=0.761$, we find that: (1) For all the listed DA tasks except $D_F \rightarrow D_F$-M, higher S corresponds to better performance, which is consistent with our intuition that using more relevant data as the T-I data improves the performance more. (2) All the listed Ss in Table 5 are close to or lower than $S(\text{animal, fruit})=0.171$, which we believe shows that our T-I data is highly irrelevant to the T-R data.

Second, in Table 6, we compare ZDDA$_2$ with the existing DA methods because the DA task $D_M \rightarrow D_M$-M is considered as one of the standard experiments in recent works [7, 14, 20, 33, 34]. Although this is not a fair comparison (because ZDDA$_2$ has no access to the T-R target-domain training data), we find

Table 8. Validation of ZDDA's performance (in mean classification accuracy (%)) with different training/testing splits and choices of classes in scene classification. GN stands for GoogleNet [38]. The definition of $P_{\text{RGB-D}}$ and the representation of the cell color in each column are the same as those in Table 7

method	training modality	testing modality	validation on train/test splits	validation on class choices
GN	D	D	52.63 ± 1.76	53.98 ± 1.68
ZDDA$_2$	D+$P_{\text{RGB-D}}$	D	56.89 ± 2.13	62.05 ± 1.97
ZDDA$_3$	D+$P_{\text{RGB-D}}$	D	58.37 ± 3.08	62.49 ± 1.74
GN	D	RGB	31.26 ± 1.76	32.60 ± 2.37
ZDDA$_2$	D+$P_{\text{RGB-D}}$	RGB	44.47 ± 2.50	45.47 ± 2.57
GN	RGB	RGB	66.26 ± 1.60	67.95 ± 2.20
ZDDA$_3$	D+$P_{\text{RGB-D}}$	RGB-D	58.68 ± 3.10	62.13 ± 1.50
# of classes / # of folds			10 / 5	9 / 10

that ZDDA$_2$ can reach the accuracy comparable to those of the state-of-the-art methods (even outperform some of them), which supports that ZDDA$_2$ is a promising DA method when the T-R target-domain training data is unavailable.
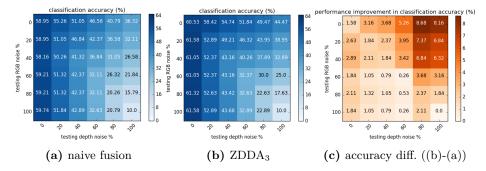
Third, we test the efficacy of ZDDA on the DA tasks constructed from D_S (adapting from depth to RGB images). We compare ZDDA with the baseline under different scene classification tasks by changing the number of scenes involved. The result is summarized in Table 7, where we list the training and testing modalities for each method. We also list the scene IDs (introduced in Sec. 4.1) involved in each task. Darker cells represent better accuracy in each column. We verify the irrelevance degree between T-R and T-I data by measuring the semantic similarity using the word2vec [29] (the same method we use in Table 5). For the 10-class experiment in Table 7, $S(D_S(\text{T-R}), D_S(\text{T-I}))=0.198$ (close to the reference $S(\text{object, scene})=0.192$), which we believe shows high irrelevance between our T-I and T-R data. For simplicity, we use E_i to refer to the experiment specified by exp. ID i in this section. For the fully supervised methods in depth domain, ZDDA (E_2, E_3) outperforms the baseline (E_1) due to the extra information brought by the T-I RGB-D pairs. We find that for most listed tasks, ZDDA$_3$ (E_3) outperforms ZDDA$_2$ (E_2), which is consistent with our intuition because the source representation in ZDDA$_2$ is constrained by the L2 loss, while the counterpart in ZDDA$_3$ is learned without the L2 constraint given the simulated target representation. The fully supervised method in RGB domain (E_6) outperforms the baseline of the domain adaptation (E_4) and ZDDA$_2$ (E_5) because E_6 has access to the T-R RGB training data which is unavailable for E_4 and E_5. The performance improvement from E_4 to E_5 is caused by ZDDA$_2$'s training procedure as well as the extra T-I RGB-D training pairs. E_3 and E_7 perform similarly, which supports that the simulated target representation in ZDDA$_3$ is similar to the real one.

Table 9. Validation of ZDDA's performance with different base network architectures in scene classification. The reported numbers are classification accuracy (%). The definition of $P_{\text{RGB-D}}$ and the representation of the cell color in each column are the same as those in Table 7

method	training modality	testing modality	base network architecture		
			GoogleNet [38]	AlexNet [26]	SqueezeNet_v1.1 [23]
BNA	D	D	51.84	49.74	48.68
ZDDA$_2$	D+$P_{\text{RGB-D}}$	D	59.74	51.05	56.32
ZDDA$_3$	D+$P_{\text{RGB-D}}$	D	63.16	51.05	56.32
BNA	D	RGB	31.58	30.26	26.58
ZDDA$_2$	D+$P_{\text{RGB-D}}$	RGB	43.16	40.00	35.79
BNA	RGB	RGB	63.68	59.47	57.37
ZDDA$_3$	D+$P_{\text{RGB-D}}$	RGB-D	63.16	51.84	56.05

To test the consistency of the performance of ZDDA compared to that of the baseline, we perform the following three experiments. First, we conduct 5-fold cross validation with different training/testing splits for the 10-scene classification. Second, we perform 10-fold validation with different selections of classes for the 9-scene classification (leave-one-class-out experiment out of the 10 selected scenes introduced in Sec. 4.1). Third, we validate ZDDA's performance with different base network architectures. The results of the first two experiments are presented in Table 8, and the result of the third experiment is shown in Table 9. The results of Table 7, Table 8, and Table 9 are consistent.

In Table 7, Table 8, and Table 9, the classification accuracy is reported under the condition of noise-free training and testing data. To let ZDDA be more robust to noisy input, we train ZDDA$_3$ step 3 with noisy training data (we use $p_{train} = 20$ as explained in Sec. 4.2), and evaluate the classification accuracy under different noise conditions for both RGB and depth testing data. The result is presented in Fig. 4, where ZDDA$_3$ (Fig. 4b) outperforms the naive fusion method (Fig. 4a) under most conditions, and the performance improvement is shown in Fig. 4c. Both Fig. 4a and Fig. 4b show that the performance degradation caused by the noisy depth testing data is larger than that caused by the noisy RGB testing data, which supports that the trained RGB-D classifier relies more on the depth domain. Traditionally, training a fusion model requires the T-R training data in both modalities. However, we show that without the T-R training data in the RGB domain, we can still train an RGB-D fusion model, and that the performance degrades smoothly when the noise increases. In addition to using black images as the noise model, we evaluate the same trained joint classifier in ZDDA$_3$ using another noise model (adding a black rectangle with a random location and size to the clean image) at testing time, and the result also supports that ZDDA$_3$ outperforms the naive fusion method. Although we only

(a) naive fusion   **(b)** ZDDA₃   **(c)** accuracy diff. ((b)-(a))

Fig. 4. Performance comparison between the two sensor fusion methods with black images as the noisy images. We compare the classification accuracy (%) of (a) naive fusion and (b) ZDDA₃ under different noise levels in both RGB and depth testing data. (c) shows that ZDDA₃ outperforms the naive fusion under most conditions

use black images as the noise model for ZDDA₃ at training time, we expect that adding different noise models can improve the robustness of ZDDA₃.

6 Conclusion and Future Work

We propose zero-shot deep domain adaptation (ZDDA), a novel approach to perform domain adaptation (DA) and sensor fusion with no need of the task-relevant target-domain training data which can be inaccessible in reality. Rather than solving the zero-shot DA problem in general, we aim at solving the problems under the assumption that task-relevant source-domain data and task-irrelevant dual-domain paired data are available. Our key idea is to use the task-relevant source-domain data to simulate the task-relevant target-domain representations by learning from the task-irrelevant dual-domain pairs. Experimenting on the MNIST [27], Fashion-MNIST [46], NIST [18], EMNIST [9], and SUN RGB-D [36] datasets, we show that ZDDA outperforms the baselines in DA and sensor fusion even without the task-relevant target-domain training data. In the task adapting from MNIST [27] to MNIST-M [13], ZDDA can even outperform several state-of-the-art DA methods which require access to the MNIST-M [13] training data. One industrial use case which we plan to apply ZDDA to in our follow-up work is training an RGD object classifier given only the textureless CAD models of those objects. In this case, depth and RGB images are source and target domains, respectively. The depth images can be rendered from the provided CAD models, and publicly available RGB-D datasets can serve as the task-irrelevant RGB-D data. We believe that ZDDA can be straightforwardly extended to handle other tasks of interest by modifying the loss functions in Fig. 2 step 2 and step 3.

References

1. Aljundi, R., Tuytelaars, T.: Lightweight unsupervised domain adaptation by convolutional filter reconstruction. In: Hua, G., Jégou, H. (eds.) ECCV Workshops. vol. 9915, pp. 508–515. Springer, Cham (2016)
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **33**, 898–916 (2011)
3. BAIR/BVLC: BAIR/BVLC AlexNet model. `http://dl.caffe.berkeleyvision.org/bvlc_alexnet.caffemodel`, accessed: 2017-03-02
4. BAIR/BVLC: BAIR/BVLC GoogleNet model. `http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel`, accessed: 2017-03-02
5. BAIR/BVLC: Lenet architecture in the Caffe tutorial. `https://github.com/BVLC/caffe/blob/master/examples/mnist/lenet.prototxt`
6. Blitzer, J., Foster, D.P., Kakade, S.M.: Zero-shot domain adaptation: A multi-view approach. In: Technical Report TTI-TR-2009-1. Technological institute Toyota (2009)
7. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3722–3731. IEEE (2017)
8. Chen, T.H., Liao, Y.H., Chuang, C.Y., Hsu, W.T., Fu, J., Sun, M.: Show, adapt and tell: Adversarial training of cross-domain image captioner. In: The IEEE International Conference on Computer Vision (ICCV). pp. 521–530. IEEE (2017)
9. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: An extension of MNIST to handwritten letters. arXiv preprint arXiv: 1702.05373 (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255. IEEE (2009)
11. Ding, Z., Shao, M., Fu, Y.: Missing modality transfer learning via latent low-rank constraint. In: IEEE Transactions on Image Processing. vol. 24, pp. 4322–4334. IEEE (2015)
12. Fu, Z., Xiang, T., Kodirov, E., Gong, S.: Zero-shot object recognition by semantic manifold distance. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2015)
13. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning (ICML-15). vol. 37, pp. 1180–1189. PMLR (2015)
14. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. Journal of Machine Learning Research (JMLR) **17**(59), 1–35 (2016)
15. Gebru, T., Hoffman, J., Li, F.F.: Fine-grained recognition in the wild: A multi-task domain adaptation approach. In: The IEEE International Conference on Computer Vision (ICCV). pp. 1349–1358. IEEE (2017)
16. Ghifary, M., Kleijn, W.B., Zhang, M.J., Balduzzi, D., Li, W.: Deep reconstruction - classification networks for unsupervised domain adaptation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) European Conference on Computer Vision (ECCV). vol. 9908, pp. 597–613. Springer, Cham (2016)
17. Gretton, A., Smola, A.J., Huang, J., Schmittfull, M., Borgwardt, K.M., Schölkopf, B.: Covariate shift and local learning by distribution matching, pp. 131–160. MIT Press, Cambridge, MA, USA (2009)

18. Grother, P., Hanaoka, K.: NIST special database 19 handprinted forms and characters database. National Institute of Standards and Technology (2016)
19. Gupta, S., Hoffman, J., Malik, J.: Cross modal distillation for supervision transfer. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2827–2836. IEEE (2016)
20. Haeusser, P., Frerix, T., Mordvintsev, A., Cremers, D.: Associative domain adaptation. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2765–2773. IEEE (2017)
21. Hoffman, J., Gupta, S., Darrell, T.: Learning with side information through modality hallucination. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 826–834. IEEE (2016)
22. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet_v1.1 model. https://github.com/DeepScale/SqueezeNet/blob/master/SqueezeNet_v1.1/squeezenet_v1.1.caffemodel, accessed: 2017-02-11
23. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv preprint arXiv: 1602.07360 (2016)
24. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R.B., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv: 1408.5093 (2014)
25. Koniusz, P., Tas, Y., Porikli, F.: Domain adaptation by mixture of alignments of second- or higher-order scatter tensors. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4478–4487. IEEE (2017)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems (NIPS) 25. pp. 1097–1105. Curran Associates, Inc. (2012)
27. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
28. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: The IEEE International Conference on Computer Vision (ICCV). IEEE (2017)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 26. pp. 3111–3119. Curran Associates Inc. (2013)
30. Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G.: Unified deep supervised domain adaptation and generalization. In: The IEEE International Conference on Computer Vision (ICCV). pp. 5715–5725. IEEE (2017)
31. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 689–696. Omnipress (2011)
32. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) European Conference on Computer Vision (ECCV). vol. 6314, pp. 213–226. Springer, Berlin, Heidelberg (2010)
33. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning (ICML-17). vol. 70, pp. 2988–2997. PMLR (2017)

34. Sener, O., Song, H.O., Saxena, A., Savarese, S.: Learning transferrable representations for unsupervised domain adaptation. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems (NIPS) 29. pp. 2110–2118. Curran Associates, Inc. (2016)

35. Sohn, K., Liu, S., Zhong, G., Yu, X., Yang, M.H., Chandraker, M.: Unsupervised domain adaptation for face recognition in unlabeled videos. In: The IEEE International Conference on Computer Vision (ICCV). pp. 3210–3218. IEEE (2017)

36. Song, S., Lichtenberg, S., Xiao, J.: SUN RGB-D: A RGB-D scene understanding benchmark suite. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 567–576. IEEE (2015)

37. Sun, B., Saenko, K.: Deep CORAL: Correlation alignment for deep domain adaptation. In: Hua, G., Jégou, H. (eds.) ECCV Workshops. vol. 9915, pp. 443–450. Springer, Cham (2016)

38. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–9. IEEE (2015)

39. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: The IEEE International Conference on Computer Vision (ICCV). pp. 4068–4076. IEEE (2015)

40. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7167–7176. IEEE (2017)

41. Venkateswara, H., Eusebio, J., Chakraborty, S., Panchanathan, S.: Deep hashing network for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5018–5027. IEEE (2017)

42. Wang, W., Arora, R., Livescu, K., Bilmes, J.: On deep multi-view representation learning. In: Bach, F., Blei, D. (eds.) Proceedings of the 32th International Conference on Machine Learning (ICML-15). vol. 37, pp. 1083–1092. PMLR (2015)

43. Wang, Y., Li, W., Dai, D., Gool, L.V.: Deep domain adaptation by geodesic distance minimization. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2651–2657. IEEE (2017)

44. Wu, C., Wen, W., Afzal, T., Zhang, Y., Chen, Y., Li, H.: A compact DNN: Approaching GoogLeNet-level accuracy of classification and domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5668–5677. IEEE (2017)

45. Wulfmeier, M., Bewley, A., Posner, I.: Addressing appearance change in outdoor robotics with adversarial domain adaptation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1551–1558. IEEE (2017)

46. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv: 1702.05374 (2017)

47. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2272–2281. IEEE (2017)

48. Yang, X., Ramesh, P., Chitta, R., Madhvanath, S., Bernal, E.A., Luo, J.: Deep multimodal representation learning from temporal data. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5447–5455. IEEE (2017)

49. Yang, Y., Hospedales, T.M.: Zero-shot domain adaptation via kernel regression on the grassmannian. In: Drira, H., Kurtek, S., Turaga, P. (eds.) BMVC Workshop on Differential Geometry in Computer Vision. BMVA Press (2015)

50. Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1859–1867. IEEE (2017)
51. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: The IEEE International Conference on Computer Vision (ICCV). pp. 2020–2030. IEEE (2017)