

Received 1 October 2024, accepted 9 December 2024, date of publication 16 December 2024, date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3518600

RESEARCH ARTICLE

Multitask Adaptation for Unlabeled Domain Using Multiple Single-Task Domains

YOUNGWOOK KANG^{1,2}, HAWOOK JEONG², JUNSUP SHIN¹,
AND JONGWON CHOI¹, (Member, IEEE)

¹Department of Advanced Imaging, Chung-Ang University, Seoul 06974, South Korea

²RideFlux Inc., Jeju 63083, South Korea

Corresponding author: Jongwon Choi (choijw@cau.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by Korea Government [Ministry of Science and ICT (MSIT)] (Artificial Intelligence Graduate School Program, Chung-Ang University) under Grant 2021-0-01341; and in part by the IITP Grant funded by Korea Government (MSIT) through the Graduate School of Metaverse Convergence Support Program under Grant IITP-2023(2024)-RS-2024-00418847.

ABSTRACT Semantic segmentation and depth estimation tasks are crucial for autonomous driving systems, but obtaining their labels from real-world datasets is costly. To address the problem, we developed a multitask domain adaptation that uses various labeled datasets with distinct tasks to adapt the multitask model for the unlabeled domain. The proposed framework can handle multiple source domains containing various task labels, which allows us to extend the combinations of acceptable source datasets in contrast to the previous multitask domain adaptation methods. We suggest using the ‘TripleMix’ approach to obtain the integrated features from the three separate domains, including two labeled domains and one unlabeled domain. In addition, we design a task correlation network that trains multiple tasks through attentional correlation, increasing the synergies between various tasks. To validate the proposed algorithm’s state-of-the-art performance based on the interactions of the different domains and tasks, we analyze it using a variety of dataset combinations that consider two virtual domains and one real-world target domain.

INDEX TERMS Domain adaptation, depth estimation, multitask learning, semantic segmentation.

I. INTRODUCTION

With the growing interest in autonomous driving, various datasets have emerged with diverse types of tasks and capturing environments [1], [2], [3]. However, collecting real-world data for autonomous driving requires not only high costs of labeling but also expensive capturing vehicles. In addition, due to the limitation of the scale of the dataset, it is difficult to cover the large-scale and real-world environment for autonomous driving. To overcome the problems, research for Unsupervised Domain Adaptation (UDA) has been conducted to achieve satisfactory performance for unlabeled domains by employing the labeled domains with the same task to overcome these problems.

In an autonomous driving environment, the ability to consider multiple tasks, such as classifying objects ahead

and calculating the distance simultaneously, is a desired skill. Multitask training can integrate the feature extractor and reduce the computation FLOPS compared to the multiple single-task models. Multitask UDA methods have emerged to accomplish the objective, but they can be employed only when the used source dataset incorporates all of the labels required for multitask learning (MTL).

Human learning, in contrast, naturally combines numerous sources and tasks of knowledge. As a result, humans can jointly recognize and draw a scene based on diverse sources, each with a different task, such as object segmentation and depth estimation, even without actual observations. Similarly, we can drive a model to jointly learn numerous complementary tasks from various sources, in order to acquire a broad range of characteristics to avoid overfitting for various activities. Accordingly, we present a novel UDA framework in which multiple tasks can be learned using multiple sources, each carrying a unique task.

The associate editor coordinating the review of this manuscript and approving it for publication was Ganesh Naik¹.

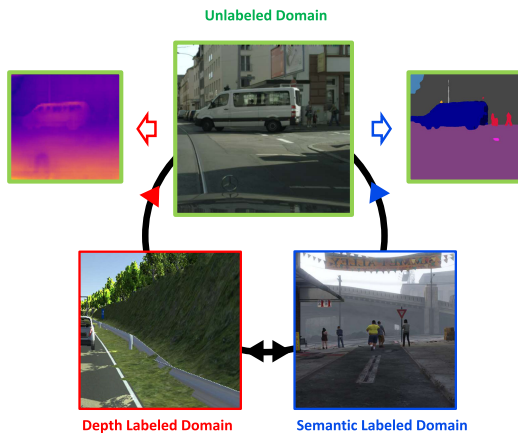


FIGURE 1. Illustration of the proposed scheme. Through the proposed algorithm, we can transfer the respective task information of the different virtual domains to the multitask model for unlabeled real-world domain.

We concentrate on the two key tasks in autonomous driving system: the semantic segmentation [4], [5], [6], [7] and depth estimation [8], [9], [10]. Depth cues provide information on object boundaries and relevant instructions that are also important for semantic segmentation, and thus both tasks are highly complemented by each other. Despite such correlations, multitask learning requires various labels as training multiple tasks which is very burdensome matter.

To solve the problems, we propose a multitask domain adaptation framework that can employ multiple source domains with the different tasks to adapt the multitask model for the target unlabeled domain. We utilize the two synthetic source domains, each respectively containing the label information of semantic segmentation and depth estimation, which are used to adapt the multitask model to the target domain for both tasks. By using synthetic labels with domain adaptation method, burden of labeling in real-world environment can be alleviated. The framework uses a *TripleMix* image and extra auxiliary decoders for domain-specific image reconstruction to generate a unified embedding feature from the multiple domains. The *TripleMix* image is created by combining three images from different domains and considering the association between semantic segmentation and depth estimation. The shared encoder trained by the *TripleMix* image as an input enables feature learning robust to domain changes, which can reduce domain gaps. To further enhance the unification of embedding features, we design the task correlation network where the multiple tasks are correlated by the attentional connections.

In addition, the task correlation network employs the auxiliary tasks where every domain is reconstructed through the respective reconstruction decoders to avoid the overfitting issue for one of the multiple tasks. The proposed framework is validated by the experimental results of semantic segmentation and depth estimation, where the two virtual source domains with the unique tasks and the real-world target domain are used. We also verify the effectiveness of

our modules in the proposed framework through numerous ablation experiments.

Our contributions can be summarized as follows:

- We create a novel multi-source UDA framework in which multiple source domains with distinct tasks can be used to train the multitask for the target domain.
- To efficiently combine the multiple domains, we propose ‘TripleMix’ method where the images from the different domains are integrated.
- We develop the task correlation network that can improve the synergy between multiple tasks while preserving the task-wise performance.
- We achieve state-of-the-art multitask performance for the real-world autonomous driving environment by using the unlabeled real-world dataset and the virtual datasets with the distinct task labels.

II. RELATED WORKS

A. MULTITASK LEARNING

A number of previous literature have explored the correlation between the semantic segmentation and depth estimation tasks [11], [12], [13], [14], [15], [16], [17], [18], [19]. Many studies have been proposed to jointly learn both tasks [20], [21], [22]. Chen et al. [20] proposed SceneNet that performs region-aware depth estimation by enforcing semantics consistency between stereo pairs.

Casser et al. [23] distilled knowledge from a teacher semantic segmentation network to a student depth estimation network, whereas others [24], [25] aimed to improve the accuracy of semantic segmentation with self-supervised depth estimation. However, such approaches are limited to the labeled environment, so datasets for MTL must be acquired, which is costly in the real-world settings.

B. DOMAIN ADAPTATION

To solve the difficulty to acquire the pixel-wise labels in the semantic segmentation, synthetic data can be utilized. However, for the real-world environment, the synthetic data causes a performance drop due to a domain gap between the real images and synthetic ones.

In several studies [26], [27], adversarial learning was employed to align the target and source features. On the other hand, batch normalization statistics were also utilized for the unified embedding features of the domains [28], [29], [30]. Choi et al. [31] constructed intermediate domain between source and target domains by consensus-based transfer. Araslanov and Roth [32] used the augmentation consistency for domain adaptation on semantic segmentation, and promotes the moving threshold method to handle the long-tail problem. Tranheden et al. [33] built a mixed image where the source and target images are integrated according to the randomly chosen classes, which is used as an input for domain adaptation.

Information of the auxiliary task is also used to enforce consistency [23], [24], [34], [35]. Hoyer et al. [24] proposed

geometric consistency with depth, comparing a class mix mask. Using the self-supervised depth estimation cues of source and target data, Wang et al. [34] employed the task feature correlation module to transfer the depth information to semantic segmentation task by using the domain discrepancy of depth prediction as the prediction error. In contrast, Lopez-Rodriguez and Mikolajczyk [36] utilized the self-supervised segmentation information to improve depth estimation performance.

C. MULTITASK DOMAIN ADAPTATION

Inspired by the annotation difficulties of the MultiTask Learning (MTL) and the benefits of using the auxiliary tasks for Unlabeled Domain Adaptation (UDA), the multitask domain adaptation methods have emerged. Peng and Dredze [37] deployed Chinese word segmentation and entity recognition simultaneously as tasks, while Ren and Lee [38] adopted depth estimation, surface normal map prediction, and instance contour map prediction for the multitask learning. The algorithms showed that MTL algorithms on UDA scenarios need to utilize auxiliary tasks as well as learning the complimentary representation. Saha et al. [39] selected semantic segmentation and depth estimation to build the cross-task relation layer capturing the dependency between the two tasks.

However, the previous studies have considered only a single synthetic dataset as the source domain where the targeting multiple tasks are well-composed. Although synthetic data typically provide relatively easy access to labels compared to real data, it is still difficult to find a single virtual dataset with all of the desired labels for various tasks. To solve this issue, we propose a novel TripleMix method that can utilize multiple data domains containing annotations of different tasks.

III. PROPOSED METHOD

In this section, we present the domain definitions for the proposed algorithm and describe the network architecture.

A. DOMAIN DEFINITIONS

The proposed method considers three domain types: two virtual source domains and one real-world target domain. According to our intuition that the multiple sources have a different task, the two virtual source domains are separately labeled by the semantic segmentation and the depth estimation, which are denoted by D_s and D_d , respectively. We define D_t for the target domain captured in the real-world environment without any label information. Thus, the objective of the proposed method is to adapt the label information of D_s and D_d to the multitask prediction for D_t .

B. MODEL ARCHITECTURE

Fig. 2 illustrates that the model consists of one shared encoder, three bottleneck modules, three attention modules, five decoders, and five initial decoders. We use reconstruction task as an additional auxiliary task for multitask learning.

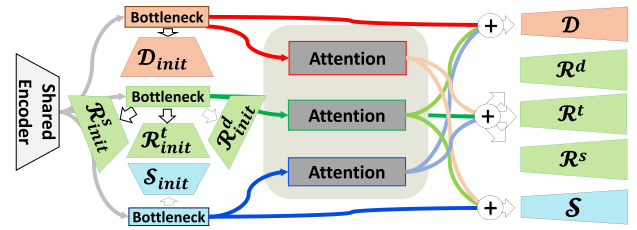


FIGURE 2. Task Correlation Network. This framework demonstrates how our main model is constructed. Using the shared encoder, the initial outputs are earned with the initial decoder. The final outputs are obtained with the task correlation network using the feature of each task and the attention feature of the corresponding auxiliary task.

Depth estimation and semantic segmentation task are not only auxiliary task for each other, but also main task of the network. Furthermore, reconstruction task is not label-needed task, so no labels are required for only auxiliary tasks in attention network.

We employ a ResNet-101 [40] as our encoder model. The feature from the encoder is fed into the following bottleneck modules, which results in the task-wise features for depth estimation, semantic segmentation, and reconstruction. Then, the task-wise features from the bottleneck modules are given to the attention module for the attention features, which are utilized as the input for the five final decoders. The attention module employs the self-attention block [41] where two vectors are multiplied to each other. The two vectors are estimated by one convolution layer and another convolution layer followed by a sigmoid function, respectively. The five decoders are composed of one depth decoder, one segmentation decoder, and three domain-wise reconstruction decoders respectively for D_d , D_s , and D_t . The depth and segmentation decoders are built with a DeeplabV2 architecture [5]. For the given input image of I^{D*} , we use the notation of $D(I^{D*})$, $S(I^{D*})$, $R^d(I^{D*})$, $R^s(I^{D*})$, and $R^t(I^{D*})$ to represent the results from the depth, segmentation, and reconstruction decoders for D_s , D_d , and D_t , respectively.

The composition and the architecture of the initial decoders are the same as those of the decoders, but the initial decoders are fed by the features from the bottleneck modules directly before the attention modules. We denote the results of the depth initial decoder, the segmentation initial decoder, and the reconstruction initial decoders for D_s , D_d and D_t by $D_{init}(I^{D*})$, $S_{init}(I^{D*})$, $R_{init}^d(I^{D*})$, $R_{init}^s(I^{D*})$, and $R_{init}^t(I^{D*})$, respectively. Bottleneck layers and initial prediction decoders are trained simultaneously with multitask layers.

C. ADDITIONAL AUXILIARY MODEL FOR PSEUDO-LABEL

In addition, we utilize an exponential moving average (EMA) model to acquire stable pseudo-labels for domain adaptation. At every iteration, after we update the original model $\mathcal{M}_{ORG}$ by using a gradient-based optimizer, the EMA model $\mathcal{M}_{EMA}$ is also updated. The architecture of the EMA model is the same as the original model, and the parameters of the EMA model are updated by the moving averages of the trained

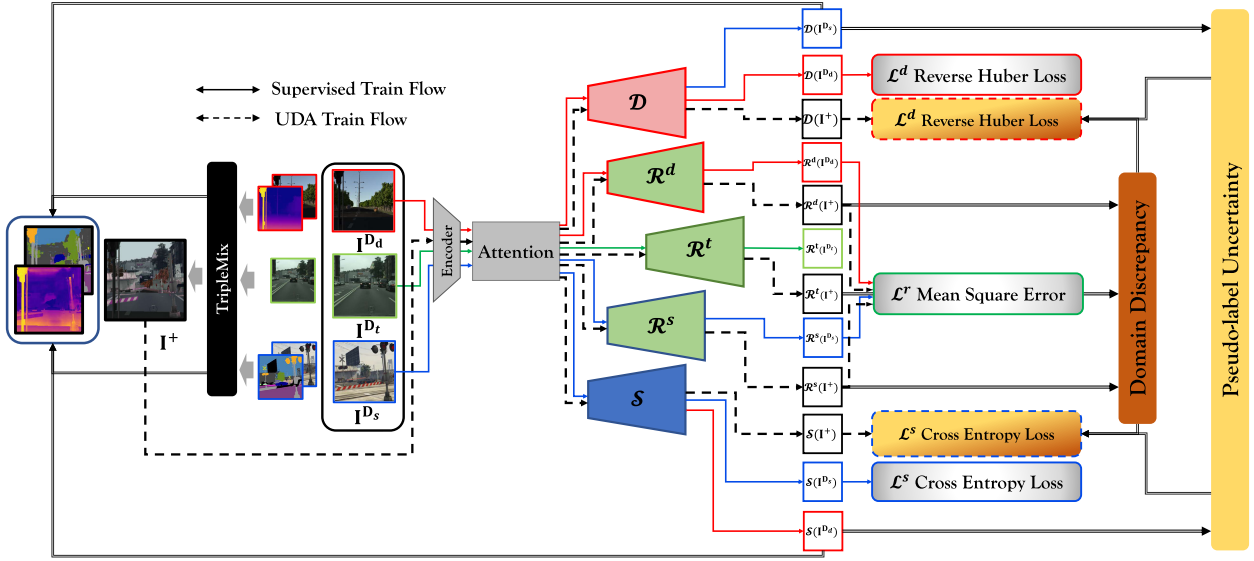


FIGURE 3. Overview of Training Phase. This figure shows how our training method works. On supervised learning step, each domain is trained on the corresponding labeled tasks, which is notated by solid line. In the unsupervised domain adaptation (UDA) phase, the images and labels (real labels and EMA pseudo-labels) from each domain are mixed, which are indicated by the dashed lines. The model is trained on all tasks with triplemix images and triplemix labels. The coefficients used for UDA loss are obtained through pseudo-label uncertainty and domain discrepancy.

parameters from the original model as follows:

$$\alpha = \min \left(1 - \frac{1}{i+1}, \alpha^{init} \right),$$

$$\mathcal{M}_{EMA}^{(i)} = \alpha \odot \mathcal{M}_{EMA}^{(i-1)} + (1 - \alpha) \odot \mathcal{M}_{ORG}^{(i-1)} \quad (1)$$

where i means a training iteration, α defines the teacher alpha parameter and α^{init} is a user-defined hyperparameter. In addition, $\mathcal{M}^{(i)}$ is a model parameter matrix on i -th train step, and $\mathcal{M}_{ORG}$ and $\mathcal{M}_{EMA}$ mean the original model and EMA model, respectively. The results from the EMA model are distinguished from the results of the original model by using the suffix of e (i.e. $eD(I^{D*})$, $eS(I^{D*})$, $eR^d(I^{D*})$, $eR^s(I^{D*})$, and $eR^t(I^{D*})$).

D. TRIPLEMIX IMAGE GENERATION FOR TRAINING

The TripleMix image can be obtained by mixing the three images of I^{D_s} , I^{D_d} , and I^{D_t} , which are extracted from the training sets of D_s , D_d , and D_t , respectively. Fig. 4 presents the two examples of TripleMix images. On A., B., C. row, the black and white binary images on the right side of each example are the sampling masks applied to the images on the left side. The white part of the sampling masks is the part where the pixel is selected. Two images on D. row are examples of TripleMix images mixed by three images above. As shown in the examples, although the three domains have different types of appearance, the semantic regions and the depth continuity are kept after the mixture of the three images.

To obtain the TripleMix image, we first estimate the mix masks for I^{D_s} and I^{D_d} , which determine the regions to be extracted. Then, the remaining region is filled with the pixels from I^{D_t} . We define the mix masks for I^{D_s} and I^{D_d} by M_s and M_d , respectively.

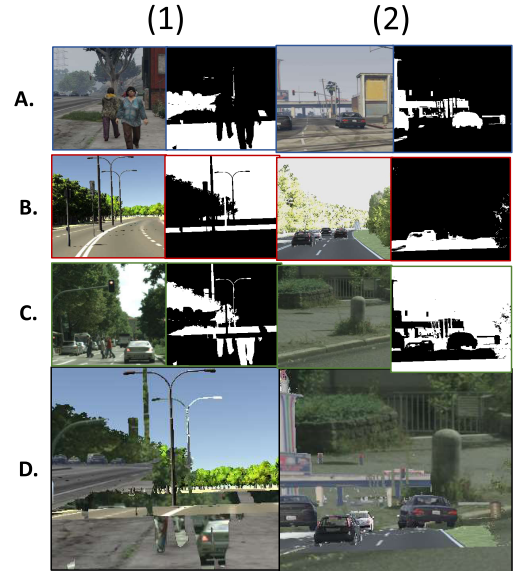


FIGURE 4. Examples of TripleMix image. Row A., B., C. show images and sampling masks of D_s , D_d , and D_t , respectively. Row D. shows results of TripleMix image.

We first obtain the initial mixing mask for segmentation (M_s^0) to generate the TripleMix image to preserve the regions of the randomly selected classes in I_s . To handle the long-tail problem, we divide the number of entire pixels by the number of class-wise pixels of I_s as follows:

$$\mathcal{P}_c = \frac{N/n_c}{\sum_{c' \in C} N/n_{c'}}, \quad (2)$$

where $c \in \{1, \dots, C\}$ is the class index for C categories, n_c means the number of pixels labeled by c -th class, and N represents the number of the entire pixels. By using $\mathcal{P}_c$,

we can build $\mathbf{M}_s^o$ to sample rare classes more frequently than dominant classes. According to the class-wise sampling probability of $\mathcal{P}_c$, we select a half of the entire categories and sample the pixels belonging to the categories, which would remain in $\mathbf{M}_s^o$.

Before estimating the initial mixing mask for depth ($\mathbf{M}_d^o$), we first separate the continuous depth values by the specified ranges to build the discrete pixel-wise categories for depth estimation. We fix the thresholds to determine the depth categories without any overlapped range as $[\theta_0, \theta_1, \theta_2, \theta_3, \dots, \theta_{N_d}]$ where N_d represents the number of depth categories. In addition, θ_0 and θ_{N_d} are minimum and maximum depth values, respectively. The threshold values are determined for the pixels to be uniformly distributed in every discrete depth category as follows:

$$\begin{aligned} \mathcal{D}_k &= \{d_{\mathbf{x}} | \theta_{k-1} < d_{\mathbf{x}} < \theta_k, \mathbf{x} \in \mathbf{X}\} \\ \mathbf{n}(\mathcal{D}_1) &= \mathbf{n}(\mathcal{D}_2) = \dots = \mathbf{n}(\mathcal{D}_{N_d}), \quad (3) \end{aligned}$$

where $\mathcal{D}_k$ is a set of pixels where its depth $d_{\mathbf{x}}$ ranged from θ_{k-1} to θ_k . $\mathbf{X}$ represents a set of all possible pixels and $\mathbf{n}(\mathcal{D}_k)$ is the number of elements in the set $\mathcal{D}_k$. Then, we randomly sample half of the N_d depth categories with the assumption of uniform random distribution, and then $\mathbf{M}_d^o$ has the value of 1 only at the pixels of the selected depth categories and otherwise, 0.

Before mixing $\mathbf{I}^{\text{Dd}}$ and $\mathbf{I}^{\text{Ds}}$, we need to remove the overlapped regions between $\mathbf{M}_s^o$ and $\mathbf{M}_d^o$. To preserve the geometric consistency between $\mathbf{I}^{\text{Dd}}$ and $\mathbf{I}^{\text{Ds}}$, we utilize the predicted depth map of $e\mathcal{D}(\mathbf{I}^{\text{Ds}})$. Since the close objects would occlude far regions, the image pixels are chosen for the overlapped pixels of $\mathbf{M}_s^o$ and $\mathbf{M}_d^o$. Thus, we refine $\mathbf{M}_s^o$ and $\mathbf{M}_d^o$ to avoid the overlapped regions in the two masks, which result in the refined mixing masks $\mathbf{M}_s$ and $\mathbf{M}_d$ as follows:

$$\begin{aligned} \mathbf{M}_s(\mathbf{x}) &= 0, \text{ if } \mathbf{GT}^{\text{Dd}}(\mathbf{x}) > e\mathcal{D}(\mathbf{I}^{\text{Ds}})(\mathbf{x}) \\ \mathbf{M}_d(\mathbf{x}) &= 0, \text{ otherwise,} \end{aligned} \quad (4)$$

where $\mathbf{x}$ defines the position where $\mathbf{M}_s^o$ and $\mathbf{M}_d^o$ overlap and $\mathbf{GT}^{\text{Dd}}(\mathbf{x})$ means the ground-truth depth value for $\mathbf{I}^{\text{Dd}}$ at the position $\mathbf{x}$.

Then, we obtain the TripleMix image by using $\mathbf{M}_s$ and $\mathbf{M}_d$. At first, the regions in $\mathbf{I}^{\text{Ds}}$ and $\mathbf{I}^{\text{Dd}}$ are extracted according to $\mathbf{M}_s$ and $\mathbf{M}_d$, respectively. Then, the remaining regions are extracted from $\mathbf{I}^{\text{Dt}}$ to fill in the blank regions after mixing $\mathbf{I}^{\text{Ds}}$ and $\mathbf{I}^{\text{Dd}}$. Thus, when we denote the mask for the remaining region by $\mathbf{M}_r = \mathbf{1} - \mathbf{M}_s - \mathbf{M}_d$ where $\mathbf{1}$ is the one-valued matrix, the TripleMix image $\mathbf{I}^+$ can be estimated as:

$$\mathbf{I}^+ = \mathbf{M}_s \odot \mathbf{I}^{\text{Ds}} + \mathbf{M}_d \odot \mathbf{I}^{\text{Dd}} + \mathbf{M}_r \odot \mathbf{I}^{\text{Dt}}, \quad (5)$$

where $\mathbf{1}$ is the one-valued matrix and $\odot$ represents the pixel-wise multiplication.

The pseudo-labels of $\mathbf{I}^+$ are obtained by using the ground-truth maps and the predictions from the EMA model. We define the ground-truth segmentation map and depth map by $\mathbf{GT}^{\text{Ds}}$ and $\mathbf{GT}^{\text{Dd}}$, respectively. Then, the pseudo-labels

of $\mathbf{I}^+$ for the semantic segmentation ($\mathbf{S}^+$) and the depth estimation ($\mathbf{D}^+$) are obtained as follows:

$$\begin{aligned} \mathbf{S}^+ &= \mathbf{M}_s \odot \mathbf{GT}^{\text{Ds}} + \mathbf{M}_d \odot e\mathcal{S}(\mathbf{I}^{\text{Dd}}) \\ &\quad + \mathbf{M}_r \odot e\mathcal{S}(\mathbf{I}^{\text{Dt}}) \\ \mathbf{D}^+ &= \mathbf{M}_s \odot e\mathcal{D}(\mathbf{I}^{\text{Ds}}) + \mathbf{M}_d \odot \mathbf{GT}^{\text{Dd}} \\ &\quad + \mathbf{M}_r \odot e\mathcal{D}(\mathbf{I}^{\text{Dt}}). \end{aligned} \quad (6)$$

As for post-processing, we refine the pseudo-labels of $\mathbf{I}^+$ by using the additional prior knowledge. At first, since the depth for 'sky' pixels should be infinite, we set the value of $\mathbf{D}^+$ to the infinite value at the 'sky' pixels in $\mathbf{S}^+$. Accordingly, we change the class of the pixels in $\mathbf{S}^+$ to the 'sky' class if their depth value of $\mathbf{D}^+$ is infinite.

IV. TRAINING METHOD

The training loss for the network is the summation of the two training losses: *supervision loss* and *adaptation loss*. The supervision loss is designed to train the network by using the original input images and their corresponding ground-truth labels, whereas the adaptation loss is to perform the integrated training to reduce the domain gaps by using the TripleMix image. The overview of the training method is shown in Fig. 3.

A. RELIABILITY FOR PSEUDO-LABELS OF TRIPLEMIX IMAGE

Before estimating the training loss, we first predict the pixel-wise reliability for the pseudo-labels of TripleMix image. The reliability of TripleMix image is estimated by a multiplication between the discrepancy map of the reconstructed images and the uncertainty score of the pseudo-labels.

1) DISCREPANCY MAP OF RECONSTRUCTED IMAGES

Despite being separately trained by different domains, the output from the three reconstruction modules should be the same for the TripleMix image when the embedding features are well unified at the backbone network. Based on intuition, we utilize the discrepancy of the reconstructed images from the three reconstruction modules.

Based on the Euclidean distance between the reconstructed images, we obtain segmentation-task discrepancy maps for segmentation unlabeled domains and depth-task discrepancy maps for depth unlabeled domains, respectively. Thus, we can estimate the discrepancy maps as follows:

$$\mathbf{E}(\mathbf{D}_1, \mathbf{D}_2) = \left[1 - \left(|\mathcal{R}^{\text{D1}}(\mathbf{I}^+) - \mathcal{R}^{\text{D2}}(\mathbf{I}^+)| / \widehat{\mathcal{R}}^+ \right) \right]_0^1, \quad (7)$$

where $\widehat{\mathcal{R}}^+ = \mathbf{M}_s \odot \mathcal{R}^s(\mathbf{I}^+) + \mathbf{M}_d \odot \mathcal{R}^d(\mathbf{I}^+) + \mathbf{M}_r \odot \mathcal{R}^t(\mathbf{I}^+)$ and $[\bullet]_0^1$ represents the clipping operation to limit the input value from 0 to 1.

2) UNCERTAINTY SCORE OF PREDICTION

We utilize the maximum probability to determine the uncertainty score of the segmentation pseudo-labels from $\mathbf{I}_d$ and $\mathbf{I}_r$.

To estimate the uncertainty score u_s of the segmentation pseudo-label, we count the number of pixels where the maximum probability for the semantic segmentation is over the predefined threshold as follows:

$$\begin{aligned} u_s(\mathbf{I}^{\text{D}_d}) &= \sum_{\mathbf{x} \in \mathbf{X}} \delta \left(\max (e\mathcal{S}(\mathbf{I}^{\text{D}_d})(\mathbf{x})) > \zeta_s \right), \\ u_s(\mathbf{I}^{\text{D}_t}) &= \sum_{\mathbf{x} \in \mathbf{X}} \delta \left(\max (e\mathcal{S}(\mathbf{I}^{\text{D}_t})(\mathbf{x})) > \zeta_s \right), \end{aligned} \quad (8)$$

where $\mathbf{X}$ contains every possible image position, and the output from δ function is 1 when the given condition satisfies and 0, otherwise. In addition, ζ_s is a user-defined hyperparameter.

The uncertainty score of depth estimation is estimated by the augmentation variance. Based on the intuition that the uncertain prediction would be unstable for the augmented input images, we estimate the uncertainty score u_d for the depth pseudo-label as:

$$\begin{aligned} u_d(\mathbf{I}^{\text{D}_s}) &= \sum_{\mathbf{x} \in \mathbf{X}} \delta \left(\sigma ([e\mathcal{D}(\mathbf{I}^{\text{D}_s}_{(1)})(\mathbf{x}), \dots, e\mathcal{D}(\mathbf{I}^{\text{D}_s}_{(J)})(\mathbf{x})]) < \zeta_d \right), \\ u_d(\mathbf{I}^{\text{D}_t}) &= \sum_{\mathbf{x} \in \mathbf{X}} \delta \left(\sigma ([e\mathcal{D}(\mathbf{I}^{\text{D}_t}_{(1)})(\mathbf{x}), \dots, e\mathcal{D}(\mathbf{I}^{\text{D}_t}_{(J)})(\mathbf{x})]) < \zeta_d \right), \end{aligned} \quad (9)$$

where σ function is to estimate the variance of its input vector, $\mathbf{I}_{(j)}$ is the augmented images of $\mathbf{I}$ with $j \in \{1, \dots, J\}$, and ζ_d is a user-defined hyperparameter.

We employ the random brightness and contrast changes to obtain $\mathbf{I}^{(j)}$ images as follows:

$$\begin{aligned} \mathbf{I}_b &= \nabla \odot \mathbf{I} + (1 - \nabla) \odot 0 \\ \mathbf{I}_c &= \nabla \odot \mathbf{I} + (1 - \nabla) \odot \mathbf{I}_g, \end{aligned} \quad (10)$$

where ∇ is uniformly sampled from range 0.5 to 1.5, $\mathbf{I}_b$ and $\mathbf{I}_c$ are the brightness-manipulated and contrast-manipulated images, respectively. $\mathbf{I}_g$ is a gray-scale image from $\mathbf{I}$.

3) RELIABILITY OF PSEUDO-LABELS

From the perspective of the labeled domain, the other two domains are unlabeled. Therefore, we need to calculate the reliability of the pseudo-labels of those two unlabeled domains. We acquire $\mathbf{B}_s$ and $\mathbf{B}_d$ that are the reliability maps for segmentation and depth task, respectively. By multiplying the estimated discrepancy maps and the uncertainty scores, we estimate $\mathbf{B}_s$ and $\mathbf{B}_d$ as follows:

$$\begin{aligned} \mathbf{B}_s &= \mathbf{M}_s + \mathbf{M}_d \odot (u_s(\mathbf{I}^{\text{D}_d}) \cdot \mathbf{E}(\mathbf{D}_d, \mathbf{D}_s)) \\ &\quad + \mathbf{M}_r \odot (u_s(\mathbf{I}^{\text{D}_t}) \cdot \mathbf{E}(\mathbf{D}_t, \mathbf{D}_s)) \\ \mathbf{B}_d &= \mathbf{M}_s \odot (u_d(\mathbf{I}^{\text{D}_s}) \cdot \mathbf{E}(\mathbf{D}_s, \mathbf{D}_d)) + \mathbf{M}_d \\ &\quad + \mathbf{M}_r \odot (u_d(\mathbf{I}^{\text{D}_t}) \cdot \mathbf{E}(\mathbf{D}_t, \mathbf{D}_d)). \end{aligned} \quad (11)$$

B. TRAINING LOSS

The proposed network has three tasks including semantic segmentation, depth estimation, and domain-wise reconstruction tasks. The entire training loss is composed of a supervision loss $\mathcal{L}_{su}$ and an adaptation loss $\mathcal{L}_{da}$ as:

$$\mathcal{L} = \mathcal{L}_{su} + \mathcal{L}_{da}. \quad (12)$$

The integrated losses are used for updating the entire network architecture including the backbone network.

The supervision and adaptation losses are composed of semantic segmentation loss $\mathcal{L}^s$, the depth estimation loss $\mathcal{L}^d$, and the reconstruction loss $\mathcal{L}^r$.

The semantic segmentation loss is designed by using the cross-entropy loss as follows:

$$\mathcal{L}^{+s}(\mathbf{S}, \hat{\mathbf{S}}, \mathbf{B}) = -\mathbb{E}_{\mathbf{x} \in \mathbf{X}} [\mathbf{B}(\mathbf{x}) \cdot \mathbf{S}(\mathbf{x})^T \log(\hat{\mathbf{S}}(\mathbf{x}))], \quad (13)$$

where $\mathbf{S}$ and $\hat{\mathbf{S}}$ represent the target and predicted segmentation maps, respectively, $\mathbf{B}$ is the weighted map for the expectation, and $\mathbf{X}$ is a set containing every position of the input image.

The depth estimation loss is estimated by the difference between the ground-truth depth and the predicted depth map as:

$$\mathcal{L}^d(\mathbf{D}, \hat{\mathbf{D}}, \mathbf{B}) = \mathbb{E}_{\mathbf{x} \in \mathbf{X}} [\mathbf{B}(\mathbf{x}) \odot \|\mathbf{D}(\mathbf{x}) - \hat{\mathbf{D}}(\mathbf{x})\|_{rh}], \quad (14)$$

where $\mathbf{D}$ and $\hat{\mathbf{D}}$ are the target and predicted depth map, respectively, $\|\bullet\|_{rh}$ means the reversed Huber norm [10], and $\mathbf{X}$ is a set containing every position of the input image.

In the case of the reconstruction loss, a Frobenius norm is used as follows:

$$\mathcal{L}^r(\mathbf{I}, \hat{\mathbf{I}}) = \frac{1}{N} \|\mathbf{I} - \hat{\mathbf{I}}\|_F^2, \quad (15)$$

where $\mathbf{I}$ and $\hat{\mathbf{I}}$ denote the target and the reconstructed images, respectively, $\|\bullet\|_F$ means the Frobenius norm, and N is the number of entire pixels.

For the supervision loss $\mathcal{L}_{su}$, we utilize the losses estimated by using the ground-truth annotation of the labeled domains. The supervision loss can be designed to consider the training of the initial task modules as follows:

$$\begin{aligned} \mathcal{L}_{su} &= \mathcal{L}^s(\mathbf{GT}_s, \mathcal{S}_{init}^{(\mathbf{I}_s)}, \mathbf{1}) + \mathcal{L}^s(\mathbf{GT}_s, \mathcal{S}^{(\mathbf{I}_s)}, \mathbf{1}) \\ &\quad + \lambda_d \left\{ \mathcal{L}^d(\mathbf{GT}_d, \mathcal{D}_{init}^{(\mathbf{I}_d)}, \mathbf{1}) + \mathcal{L}^d(\mathbf{GT}_d, \mathcal{D}^{(\mathbf{I}_d)}, \mathbf{1}) \right\} \\ &\quad + \lambda_r \sum_{\mathbf{I} \in \{\mathbf{I}_s, \mathbf{I}_d, \mathbf{I}_r\}} \left\{ \mathcal{L}^r(\mathbf{I}, \mathcal{R}_{init}^{(\mathbf{I})}) + \mathcal{L}^r(\mathbf{I}, \mathcal{R}^{(\mathbf{I})}) \right\}. \end{aligned} \quad (16)$$

For the adaptation loss, $\mathcal{L}_{da}$ is designed to train the reliable pseudo-labels of the TripleMix image based on their reliability. Thus, we obtain the training loss for the TripleMix image by considering the pseudo-labels as the target maps as

TABLE 1. Semantic segmentation accuracy for GTA5+vkitti → cityscapes.

Method	D_s	D_d	MTL	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	mIoU
DACS [33]	G	×	×	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
corDA [34]	G	×	×	94.3	59.0	87.9	33.6	44.1	41.1	48.7	53.4	88.1	47.7	89.2	67.4	37.9	90.0	43.3	54.3	0.0	17.6	44.1	54.8
Src only	G	v	O	81.8	19.2	74.4	18.6	17.3	24.2	27.1	13.2	80.4	29.5	82.8	55.9	11.3	80.1	33.1	32.6	3.8	21.7	17.6	38.1
Ours	G	v	O	90.9	36.8	87.0	32.8	34.8	35.9	46.5	54.4	86.8	47.9	86.5	66.3	28.3	89.3	54.1	54.4	0.0	33.5	52.7	53.6

follows:

$$\begin{aligned}
\mathcal{L}_{da} = & \mathcal{L}^s \left(\mathcal{S}_e^{(\mathbf{I}^+)}, \mathcal{S}_{init}^{(\mathbf{I}^+)}, \mathbf{B}_s \right) + \mathcal{L}^s \left(\mathcal{S}_e^{(\mathbf{I}^+)}, \mathcal{S}^{(\mathbf{I}^+)}, \mathbf{B}_s \right) \\
& + \lambda_d \left\{ \mathcal{L}^d \left(\mathcal{D}_e^{(\mathbf{I}^+)}, \mathcal{D}_{init}^{(\mathbf{I}^+)}, \mathbf{B}_d \right) + \mathcal{L}^d \left(\mathcal{D}_e^{(\mathbf{I}^+)}, \mathcal{D}^{(\mathbf{I}^+)}, \mathbf{B}_d \right) \right\} \\
& + \lambda_r \sum_{\mathbf{I} \in \{\mathbf{I}_s, \mathbf{I}_d, \mathbf{I}_t\}} \left\{ \mathcal{L}^r \left(\mathbf{I}^+, \mathcal{R}_{init}^{(\mathbf{I}^+)} \right) + \mathcal{L}^r \left(\mathbf{I}^+, \mathcal{R}^{(\mathbf{I}^+)} \right) \right\}.
\end{aligned} \quad (17)$$

V. EXPERIMENTS

A. DATASET

Various datasets are used including the Cityscapes [1] for the unlabeled domain, GTA5 [2] for the semantic labeled domain, and virtual KITTI [3] for the depth labeled domain. For various domain combinations, SYNTHIA [44] for the semantic labeled domain and KITTI [45] for the unlabeled domain are also used. For simplicity, we abbreviate Cityscapes, GTA5, virtual KITTI, SYNTHIA, and KITTI datasets by C., G., vK., S., and K., respectively.

1) CITYSCAPES

Cityscapes dataset [1] is a real-world dataset including the annotation of fine semantic segmentation with 19 classes and consists of 2,975 training images as well as 500 validation images. Following the experimental protocol used by [34], we use the training set without labels as the target domain for the adaptation and report our results on the validation set.

2) GTA5

The GTA5 dataset [2] is generated from a game environment, which contains 24 966 images with Cityscapes-style annotations (19 classes). We employed the GTA5 dataset as a segmentation labeled source domain.

3) SYNTHIA

The SYNTHIA dataset [44] is a large collection of images with high variability due to changes in illumination, textures, pose of dynamic objects and camera viewpoints on a virtual environment. SYNTHIA contains 16 classes from Cityscapes-style annotation except for terrain, truck, and train class. We employed the SYNTHIA dataset as a segmentation labeled source domain.

4) KITTI

The KITTI dataset [45] is a real-world dataset. We used raw data including left, and right images and Velodyne Lidar points. We computed the depth from the calibration

file and Velodyne points and used the computed depth for evaluation.

5) VIRTUAL KITTI

The virtual KITTI (vKITTI) dataset [3] is a photo-realistic synthetic video dataset designed to learn and evaluate computer vision models for multiple video understanding tasks. The dataset contains 50 high-resolution monocular videos with 21,260 frames generated from five different virtual worlds in urban settings under different imaging and weather conditions. We employed 12,756 images with various weather conditions (sunset, fog, and over-cast).

B. IMPLEMENTATION DETAILS

The batch size is set to 2, and we use stochastic gradient descent with a learning rate of 2.5×10^{-4} with weight decay 5×10^{-4} and we train 150,000 iterations. For experiments, RTX 3090 GPU and Pytorch 1.7 were used. The training time per batch with batch size 2 takes 2.4 s, and the training memory takes 20.1G. The inference time per image takes 0.1 s, and the inference memory takes 11.9G. We proceed inference without removing unnecessary parts. Therefore, the inference memory is relatively high. All training steps are capable with RTX 3090. The hyper-parameters are set as $\alpha_{lprc} = 0.9$, $N^{AUG} = 3$, $\zeta_s = 0.968$, and $\zeta_d = 0.001$. Following corDA, we set λ_d and λ_r by 0.01 and 0.001, respectively.

C. SEMANTIC SEGMENTATION

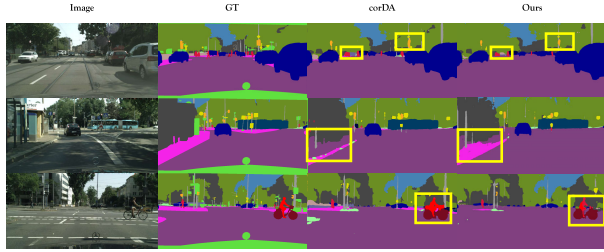
We evaluate the effectiveness of the proposed model on various episodes for semantic segmentation task. We report the IoU (class IoU) and mIoU (mean IoU) performance. The right arrow ($\rightarrow$) indicates a domain adaptation from the source domain to the target domain. We earned source-only results by training without $\mathcal{L}_{da}$.

1) RESULTS ON GTA5, vKITTI $\rightarrow$ CITYSCAPES

Table 1 shows semantic segmentation performance on GTA5, vKITTI $\rightarrow$ Cityscapes environment of our proposed model and other methods evaluated. reported the IoU and mIoU for 19 classes. We exploit the semantic segmentation information from the GTA5 domain and depth information from the vKITTI domain. We compare our performance with DACS [33] and corDA [34]. Although our overall performance is lower than that of corDA, our result achieves better performance in the categories of the truck, terrain, motorcycle, and bike, which rarely appear on the images and

TABLE 2. Semantic segmentation accuracy for SYNTHIA+vkitti → cityscapes.

Method	D_s	D_d	MTL	Road	SW	Build	Wall	fence	pole	TL	TS	Veg	Sky	Person	Rider	Car	Bus	MC	Bike	mIoU
Advent [42]	S.	×	×	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2
DADA [43]	S.	×	×	89.2	44.8	81.4	6.8	0.3	26.2	8.6	11.1	81.8	84.0	54.7	19.3	79.7	40.7	14.0	38.8	42.6
CTRL [39]	S.	S.	O	86.9	43.0	80.7	19.2	0.9	27.2	11.6	12.6	81.3	83.2	60.7	24.0	84.2	46.2	22.0	44.2	45.5
Ours	S.	v.	O	94.7	68.4	82.1	20.2	0.2	24.6	22.4	14.5	85.0	81.1	65.3	34.3	86.4	41.9	31.5	48.1	50.0

**FIGURE 5.** Qualitative results on semantic segmentation task in Cityscapes for models trained in virtual KITTI, GTA 5 → Cityscapes. Yellow bounding boxes refer to areas of the prediction more accurate compared to the corDA.

have a low proportion in the images. This result shows our low pixel rate class sampling works well. We present the qualitative results in Fig. 5. The boundaries of bicycle class and person class pixels are more explicit.

2) RESULTS ON SYNTHIA, vkitti → CITYSCAPES

We perform additional experiments using the source domains of SYNTHIA and vkitti to adapt the target domain of Cityscapes to validate the generality of the proposed algorithm. Instead of the GTA5 dataset, we use the SYNTHIA dataset as the semantic segmentation domain, while the depth estimation domain is vkitti as the original experiment. Table 2 shows semantic segmentation performance on SYNTHIA, vkitti → Cityscapes environment of our proposed model and CTRL [39] evaluated. We report the IoU for 16 classes and mIoU. CTRL which uses the same tasks for multitask domain adaptation, is trained with SYNTHIA → Cityscapes environment. Our model shows superior performance compared to CTRL. Our approach demonstrates an improvement in accuracy for recognizing the pedestrians, including the classes of *Person* and *Rider*, which are critical for the safety of autonomous driving.

3) PERFORMANCE ENHANCEMENT DIFFERENCE BY SOURCE DOMAIN DATASET

We report Fig. 6 which contains sample images from SYNTHIA, GTA5 and Cityscapes dataset each. As you can see from the figure, GTA5 dataset is much more similar to the Cityscapes dataset than SYNTHIA is. Therefore, the performance by the domain adaptation algorithm is enhanced more in the SYNTHIA dataset.

D. DEPTH ESTIMATION

We also evaluated the effectiveness of the proposed model on various episodes for depth estimation task. We report

**FIGURE 6.** Sample images of various datasets. (A), (B), (C) images each refers to SYNTHIA, GTA5 and Cityscapes dataset respectively.**TABLE 3.** Depth estimation evaluation for various dataset episodes.

Method	src. ^S	src. ^D	tgt.	Lower is better				Higher is better		
				Rel	Rel ²	RMS	LRMS	σ_1	σ_2	σ_3
CTRL [39]	S.	S.	C.	0.3	6.3	14.8	0.6	0.30	0.58	0.77
Src only	G.	vk.	C.	0.4	8.8	21.9	0.4	0.45	0.66	0.80
Ours	G.	vk.	C.	0.3	3.7	13.7	0.3	0.52	0.81	0.92
Ours	S.	vk.	C.	0.3	3.6	13.9	0.3	0.55	0.81	0.91
Ours	G.	vk.	K.	0.2	1.9	7.2	0.2	0.62	0.86	0.95

Absolute Relative Difference ($|Rel|$), Squared Relative Difference (Rel^2), Root Mean Squared Error (RMS), its log-variant LRMS. We also report the accuracy metrics [8] as denoted by σ_1 , σ_2 , σ_3 . We provide a result with ground truth depth capped at 80/m. Table 3 shows depth estimation performance on various domain compositions. We report an episode which includes the KITTI dataset as the target domain as well. Since the virtual KITTI dataset and the KITTI dataset are similar, depth estimation task in the corresponding data domain composition shows the best performance. Since CTRL utilizes only one source domain, CTRL is limited in that it is necessary to select a dataset with labels for both the segmentation task and the depth task. The proposed method can use the source dataset with only part of the multitask annotations. Fig. 7 qualitatively visualizes the experimental results of the depth estimation, displaying well-represented details, such as traffic lights, riders, and bicycles.

E. ANALYSIS

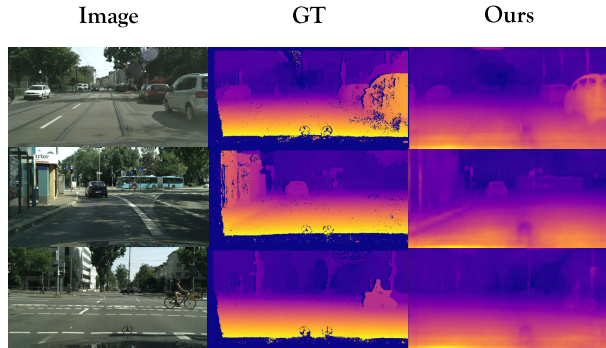
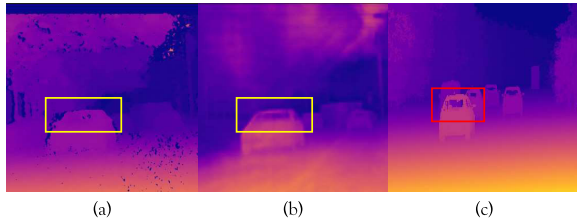
1) DOMAIN SPECIFIC LABEL PROPERTY: VIRTUAL KITTI

We provided qualitative results of the depth estimation using the GTA5 dataset as a labeled semantic segmentation source domain, virtual KITTI as a labeled depth estimation source domain, and Cityscapes as an unlabeled target domain.

As shown in Fig. 8 (b), we can find that the prediction's vehicle window appear to be hollow, which is filled in the GT depth map of Cityscape dataset (Fig. 8 (a)). The reason can

TABLE 4. Ablation test on low rate class sampling.

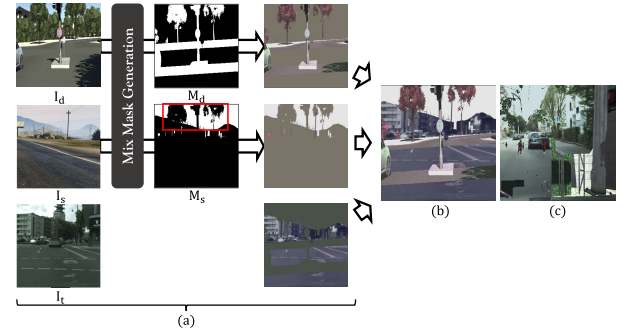
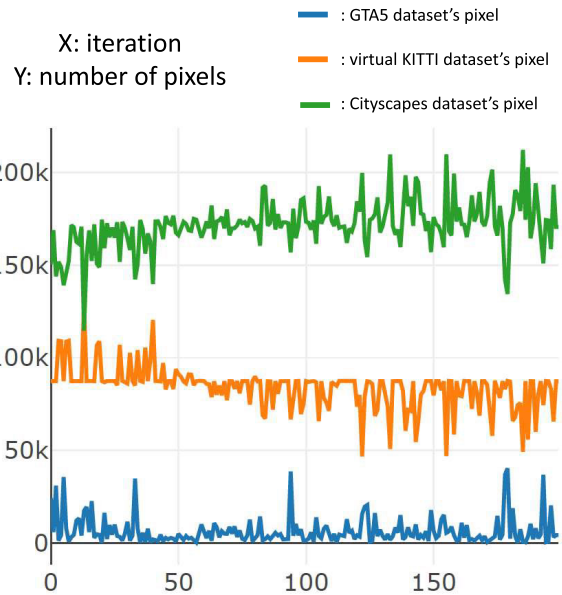
	lrps.	Road	SW	Build	Wall	fence	pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	mIoU
Ours	×	92.7	53.8	86.0	31.1	28.4	35.4	39.7	43.1	85.9	43.0	83.3	63.8	28.1	87.9	49.7	53.6	0.0	33.8	52.1	52.2
Ours	O	90.9	36.8	87.0	32.8	34.8	35.9	46.5	54.4	86.8	47.9	86.5	66.3	28.3	89.3	54.1	54.4	0.0	33.5	52.7	53.6

**FIGURE 7.** Qualitative results on depth estimation task in Cityscapes for models trained in virtual KITTI, GTA 5 → Cityscapes.**FIGURE 8.** (a) is a GT depth map of Cityscapes dataset, (b) shows our predicted depth map, and (c) presents a GT depth map of virtual KITTI dataset.

be found from Fig. 8 (c) showing the ground-truth(GT) depth map of virtual KITTI dataset that is our depth-labeled source domain. While it is infeasible to acquire the GT depth map for the hollow regions in the real-world GT depth map as shown in Fig. 8 (a), we can achieve the detail GT depth map from the synthetic dataset. Through this, we can verify that our model learn the depth information from the virtual KITTI well, which also confirms that we can predict even the difficult parts in real-world domain by using the characteristics of our source domain.

2) TRIPLEMIX EXPLANATION AND CHANGES BY ITERATION

In Fig. 9, (a) is a figure showing the process of TripleMix in more detail. According to our proposed method, as the learning progresses, the depth of the semantic segmentation labeled data is also learned. The difference between (b) and (c) reveals that the overlapping pixels are selected well according to the depth. The red box in (a) shows the details. Although the sky class pixels are sampled in the segmentation domain, the tree pixels in the depth domain are selected according to the distance comparison. Fig. 10 shows pixel ratio of each domain's image which are used in TripleMix. Blue line, orange line, and green line refers to pixels amount sampled from image of D_s , D_d , D_t . By low rate pixel class sampling, pixels from I^D_s occupy the smallest percentage.

**FIGURE 9.** (a) is Explanation of TripleMix, (b) is TripleMix image from 45,000 iteration and (c) is TripleMix image from 1,000 iteration.**FIGURE 10.** Pixel ratio for each domain of images used in TripleMix..

3) LOW RATE PIXEL CLASS SAMPLING

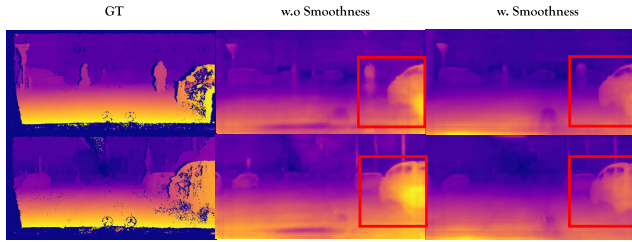
We implement low rate class sampling to handle long-tail classes such as motorcycle, bicycle by utilizing $\mathcal{P}_c = \frac{N/n_c}{\sum_c N/n_c}$. Table 4 reports the comparison of results according to how the class is sampled at the mixmask generation. Compared to the random algorithm without using the long-tail class handling, the proposed algorithm can successfully improve the sparse classes such as fence and bike.

4) DEPTH PSEUDO LABEL UNCERTAINTY

Table 5 reveals how the depth pseudo label uncertainty affects our performance. We compare depth estimation performance between a variant calculating depth pseudo label uncertainty and another variant where the uncertainty is fixed by 0.001. Furthermore, we show how our method affects segmentation

TABLE 5. Ablation test on depth pseudo label uncertainty and depth smoothness.

Method	U	smooth.	Lower is better				Higher is better			
			Rel	Rel ²	RMS	LRMS	σ_1	σ_2	σ_3	mIoU
Ours	×	O	0.3	4.2	14.5	0.5	0.49	0.71	0.82	52.2
Ours	O	×	0.3	4.0	14.6	0.4	0.51	0.79	0.91	52.8
Ours	O	O	0.3	3.7	13.7	0.3	0.52	0.81	0.92	53.6

**FIGURE 11.** Comparison of qualitative evaluation with and without smoothness.**TABLE 6.** Ablation test on different pseudo-labeling method.

	init.	RMSE	mIoU
Ours	×	15.3	51.3
Ours	O	13.7	53.6

performance by reporting mIoU. According to the above results, we show the depth pseudo label uncertainty is helpful for stable learning.

5) SMOOTHNESS TERM

Fig. 11 shows the comparison of qualitative evaluation with and without smoothness term. The detailed parts of the depth map are highlighted through correlating auxiliary tasks, but some values are out of range. By employing the L2 smoothness term, we can obtain the result where the out-of-range values are smoothed well. In addition, we can find the effect of the smoothness through the reduced artifacts of the final result. Table 5 also presents the effectiveness of the smoothness term quantitatively.

6) VARIANTS FOR PSEUDO-LABELING METHOD

Different results are obtained depending on which prediction the pseudo label used in the UDA step is selected. To confirm this, we compared using an initial prediction as the pseudo-label at the beginning of training and using a task correlated output as the pseudo label for the whole training. Table 6 confirms that using a simpler initial output as a pseudo-label rather than a complex task-correlated result at the beginning of learning helps more stable learning.

VI. CONCLUSION

In this work, we introduced a novel unsupervised domain adaptation approach that applies multiple source domains with individual tasks. To represent the diverse tasks and domains into a unified feature space, we proposed the TripleMix image that is made of three images from different source domains. Also, extra auxiliary decoders for domain reconstruction were used to generate a unified embedding feature from the multiple domains. The proposed framework

was validated by the experimental results of semantic segmentation and depth estimation, and the results show that knowledge from various domains are combined to enhance the multitask performance on an unlabeled target domain. For future work, more various visual task can be added to our method such as object detection. While maintaining our method architecture, designing additional constraint loss which connects learning tasks can be studied.

REFERENCES

- [1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.
- [2] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2016, pp. 102–118.
- [3] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "VirtualWorlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4340–4349.
- [4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
- [5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
- [6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 801–818.
- [7] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
- [8] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Jun. 2014, pp. 1–9.
- [9] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2002–2011.
- [10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.
- [11] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3712–3722.
- [12] D. Xu, W. Ouyang, X. Wang, and N. Sebe, "PAD-Net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 675–684.
- [13] R. Cipolla, Y. Gal, and A. Kendall, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7482–7491.
- [14] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint task-recursive learning for semantic segmentation and depth estimation," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2018, pp. 238–255.
- [15] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, "Pattern-affinitive propagation across depth, surface normal and semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4101–4110.
- [16] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2502–2511.
- [17] T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *Proc. ICML*, vol. 1, Jul. 2020, pp. 9120–9132.

- [18] S. Vandenhende, S. Georgoulis, and L. V. Gool, "MTI-Net: Multi-scale task interaction networks for multi-task learning," in *Proc. ECCV*, Jan. 2020, pp. 527–543.
- [19] M. Kanakis, D. Bruggemann, S. Saha, S. Georgoulis, A. Obukhov, and L. Van Gool, "Reparameterizing convolutions for incremental multi-task learning without task interference," in *Proc. ECCV*, Jul. 2020, pp. 689–707.
- [20] P.-Y. Chen, A. H. Liu, Y.-C. Liu, and Y. F. Wang, "Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2619–2627.
- [21] J. Jiao, Y. Cao, Y. Song, and R. W. H. Lau, "Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss," in *Proc. ECCV*, Jan. 2018, pp. 55–71.
- [22] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. D. Stefano, "Geometry meets semantics for semisupervised monocular depth estimation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 298–313.
- [23] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 8001–8008.
- [24] L. Hoyer, D. Dai, Y. Chen, A. Köring, S. Saha, and L. Van Gool, "Three ways to improve semantic segmentation with self-supervised depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11130–11140.
- [25] H. Jiang, G. Larsson, M. Maire, G. Shakhnarovich, and E. Learned-Miller, "Self-supervised relative depth learning for urban scene understanding," in *Proc. ECCV*, Jan. 2018, pp. 20–37.
- [26] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2962–2971.
- [27] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1989–1998.
- [28] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018.
- [29] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "AutoDIAL: Automatic domain alignment layers," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5077–5085.
- [30] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7346–7354.
- [31] J. Choi, Y. Choi, J. Kim, J. Chang, I. Kwon, Y. Gwon, and S. Min, "Visual domain adaptation by consensus-based transfer to intermediate domain," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 10655–10662.
- [32] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15384–15394.
- [33] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "DACs: Domain adaptation via cross-domain mixed sampling," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1379–1389.
- [34] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, "Domain adaptive semantic segmentation with self-supervised depth estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8515–8525.
- [35] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," 2020, *arXiv:2002.12319*.
- [36] A. Lopez-Rodriguez and K. Mikolajczyk, "DESC: Domain adaptation for depth estimation via semantic consistency," 2020, *arXiv:2009.01579*.
- [37] N. Peng and M. Dredze, "Multi-task domain adaptation for sequence tagging," 2016, *arXiv:1608.02689*.
- [38] Z. Ren and Y. J. Lee, "Cross-domain self-supervised multi-task feature learning using synthetic imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 762–771.
- [39] S. Saha, A. Obukhov, D. P. Paudel, M. Kanakis, Y. Chen, S. Georgoulis, and L. Van Gool, "Learning to relate depth and semantics for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8197–8207.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.
- [42] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2517–2526.
- [43] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. P. Pérez, "DADA: Depth-aware domain adaptation in semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7364–7373.
- [44] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3234–3243.
- [45] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.



YOUNGWOOK KANG received the B.S. degree in integrative engineering from Chung-Ang University, Seoul, South Korea, in 2020, and the M.S. degree from the Department of Advanced Imaging, GSAIM, Chung-Ang University, in 2022. He is currently with RideFlux Inc., a startup that develops urban autonomous driving software.



HAWOOK JEONG received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2009, and the Ph.D. degree from the Department of Electrical Engineering and Computer Sciences, Seoul National University, in 2015. From 2015 to 2018, he was in charge of developing AI-based image software for mobile devices with Samsung Electronics. He is currently an Executive Vice President with RideFlux Inc., a startup that develops urban autonomous driving software, and is particularly interested in computer vision and machine learning-based solutions for autonomous driving software.



JUNSUP SHIN received the B.S. degree in integrative engineering from Chung-Ang University, Seoul, South Korea, in 2020, and the M.S. degree from the Department of Advanced Imaging, GSAIM, Chung-Ang University, in 2022. He is currently with MediPixel, a startup that develops an AI-based medical platform.



JONGWON CHOI (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from KAIST, Daejeon, South Korea, in 2012 and 2014, respectively, and the Ph.D. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2018. From 2018 to 2020, he was with the Research Intelligence Research Center, Samsung SDS, Seoul, as a Research Engineer. In 2020, he joined the Department of Advanced Imaging, Chung-Ang University, Seoul, where he is currently an Associate Professor. His research interests include deep learning, computer vision, and AI safety.

...