2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

PCT: Perspective Cue Training Framework for Multi-Camera BEV Segmentation

Haruya Ishikawa[1], Takumi Iida[2], Yoshinori Konishi[2], and Yoshimitsu Aoki[1]

***Abstract*— Generating annotations for bird's-eye-view (BEV) segmentation presents significant challenges due to the scenes' complexity and the high manual annotation cost. In this work, we address these challenges by leveraging the abundance of unlabeled data available. We propose the Perspective Cue Training (PCT) framework, a novel training framework that utilizes pseudo-labels generated from unlabeled perspective images using publicly available semantic segmentation models trained on large street-view datasets. PCT applies a perspective view task head to the image encoder shared with the BEV segmentation head, effectively utilizing the unlabeled data to be trained with the generated pseudo-labels. Since image encoders are present in nearly all camera-based BEV segmentation architectures, PCT is flexible and applicable to various existing BEV architectures. In this paper, we applied PCT for semi-supervised learning (SSL) and unsupervised domain adaptation (UDA). Additionally, we introduce strong input perturbation through Camera Dropout (CamDrop) and feature perturbation via BEV Feature Dropout (BFD), which are crucial for enhancing SSL capabilities using our teacher-student framework. Our comprehensive approach is simple and flexible but yields significant improvements over various baselines for SSL and UDA, achieving competitive performances even against the current state-of-the-art.**

I. INTRODUCTION

Bird's-eye-view (BEV) segmentation is crucial for autonomous driving and navigation systems, providing a comprehensive spatial understanding of the vehicle's surroundings. However, creating accurate BEV annotation is notoriously tricky and resource-intensive, requiring sensor calibrations and accurate 3D annotations. Additionally, the performance of BEV segmentation models often suffers from domain shifts, where models trained in one environment underperform in another due to differences in geography, weather, and lighting conditions. Semi-supervised learning (SSL) and unsupervised domain adaptation (UDA) emerge as promising approaches to mitigate these challenges by effectively utilizing unlabeled data.

SSL and UDA are paradigms that aim to improve model performance by utilizing labeled and unlabeled data sets during training. In SSL, the unlabeled data typically come from the same distribution as the labeled data, allowing the model to learn more generalizable features. On the other hand, UDA deals with scenarios where the unlabeled data exhibit characteristics of domain gaps, such as variations in weather conditions, lighting, or urban landscapes. For instance, an autonomous vehicle trained in one city may need to adapt to different weather conditions, architectural styles, and even traffic regulations, such as driving on opposite sides of the road, when deployed in another city.

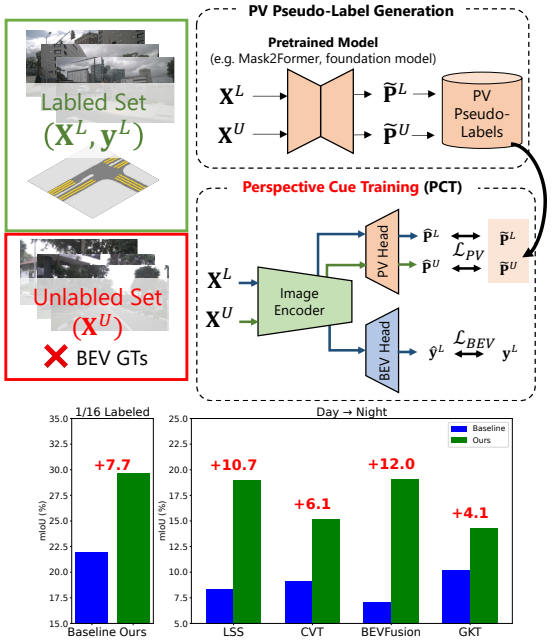

Fig. 1: **General overview of our proposed Perspective Cue Training (PCT) framework and the impact it has on tasks requiring the utilization of unlabeled data.** PCT framework utilizes PV pseudo-labels generated from easily accessible models (e.g. Mask2Former for semantic segmentation) to train multi-camera BEV segmentation models. PCT is flexible and applicable to various BEV architectures. Our method significantly improves the baseline for both SSL and UDA tasks utilizing unlabeled data.

Recent progress in annotation approaches like VMA [1] and CAMA [2] has shown promise in reducing the cost of manual annotation for HD maps. However, these methods are still prone to errors and annotating all possible scenarios and semantic categories is still infeasible.

In this paper, we propose the Perspective Cue Training (PCT) framework, which utilizes perspective view (PV) tasks for settings where an abundance of unlabeled data are available, like SSL and UDA. We noticed that abundant PV images, taken from the vehicle's cameras, are unlabeled for datasets related to BEV perception [3], [4]. The recent surge in the performance of PV tasks, notably semantic segmentation, has been remarkable, thanks to publicly available pretrained models and datasets like Cityscapes and BDD100k

[1]Haruya Ishikawa and Yoshimitsu Aoki are with Department of Electrical Engineering, Keio University, Yokohama, Japan haruyaishikawa@keio.jp

[2]Takumi Iida and Yoshinori Konishi are with SenseTime Japan, Kyoto, Japan

The project page is at https://haruishi43.github.io/pct_bevseg/.

[5], [6], [7], [8], [9], [10], [11]. These advancements allow us to generate pseudo-labels for many unlabeled PV images used in BEV segmentation training. The PCT framework capitalizes on this by training the image encoder with an additional PV head in a multi-task learning manner with the BEV segmentation head, effectively utilizing the pseudo-labels generated from the PV tasks. PCT is flexible because image encoders exist in nearly all camera-based BEV segmentation architectures, making it applicable to various existing BEV architectures. Furthermore, there are no added computational costs during inference because the PV head is only required during training. Utilizing PV pseudo-labels is an easy method to boost scenarios where BEV annotations are scarce, such as SSL and UDA, as shown in Figure 1. Note that PCT only generates PV pseudo-labels, not BEV pseudo-labels.

Furthermore, we introduce additional modules such as Camera Dropout (CamDrop) and BEV Feature Dropout (BFD) to refine our training strategy further. These modules, particularly when combined with PCT, demonstrate vital performance improvements for SSL and UDA.

The contributions of this work are as follows:

- We tackle the challenge in BEV segmentation, where labeled data is limited, by leveraging PV images of labeled and unlabeled data with our proposed Perspective Cue Training (PCT) framework. We demonstrate the effectiveness of our approach in both SSL, where BEV annotations in the source domain are scarce, and UDA, where BEV annotations in the target domain are unavailable.
- To our knowledge, SSL for multi-camera BEV segmentation has not been previously explored. We propose baselines based on SSL methods from other tasks like semantic segmentation. We introduce techniques such as CamDrop and BFD to further enhance the SSL capabilities of not just our method but can also be applied to the baseline methods.
- For UDA, we compare our method against various baselines and the current state-of-the-art approach, DualCross [12], which utilizes a teacher model trained on LiDAR data. Our method shows superior performance in most benchmarks without relying on additional modalities.

II. RELATED WORK

Camera-based BEV Segmentation. Recent advances in camera-based BEV segmentation have leveraged multi-camera setups to reconstruct the 3D scene and project it onto a BEV plane. Notable methods include LSS [13], BEVDepth [14], BEVFusion [15], CVT [16], and GKT [17]. LSS encodes images from arbitrary camera rigs by implicitly unprojecting them to 3D, offering a flexible framework for BEV segmentation. BEVDepth extends LSS by leveraging explicit depth supervision of the 2D-to-BEV module. BEVFusion fuses multi-task and multi-sensor data with a unified BEV representation, showcasing the benefits of integrating diverse data sources for improved segmentation and obtaining SOTA performance on the nuScenes dataset [3]. CVT proposes a novel transformer-based 2D-to-BEV module that implicitly learns to map individual camera views into BEV representation. Finally, GKT improves upon CVT by introducing efficient and robust 2D-to-BEV representation learning with geometric cues to enhance the transformation process.

Additionally, the integration of PV cues has shown promise in augmenting BEV segmentation. X-Align [18] demonstrates that cross-modal cross-view alignment, facilitated by PV multi-task learning with PV pseudo-labels, can significantly boost BEV segmentation performances. While our work shares similarities with X-Align in utilizing PV pseudo-labels, we extend this concept by employing a PV task head to leverage unlabeled PV images. This approach is effective in scenarios with limited BEV annotations, as we show in our SSL and UDA experiments.

Semi-Supervised Learning (SSL). SSL is essential for tasks like BEV segmentation where labeled data is limited. Techniques like pseudo-labeling and consistency regularization have been explored, with the Mean-Teacher framework [19] being notable for its effectiveness across various tasks [20], [21], [22]. Input perturbations, like Cutout [23] and CutMix [24], are crucial for SSL in semantic segmentation [25]. UniMatch [26], based on FixMatch [27], and CCT [28] utilizes feature perturbations to enhance SSL capabilities.

In BEV segmentation, SSL is less explored. SkyEye [29] uses a self-supervised approach of effectively utilizing BEV pseudo-labels generated from PV semantic segmentation task, while [30] introduces conjoint rotation for augmentation to increase supervised data. Both of these methods are for monocular BEV segmentation and no literature exists on multi-camera BEV segmentation to the best of our knowledge. Our work introduces a PV sub-task to leverage unlabeled data for SSL. We also introduce CamDrop and BFD for input and feature perturbations, respectively. These enhancements are vital for applying SSL frameworks like Mean-Teacher and UniMatch to BEV segmentation.

Unsupervised Domain Adaptation (UDA). UDA aims to adapt models trained on a labeled source domain to perform well on an unlabeled target domain. Techniques such as Adversarial Discriminative Domain Adaptation [31], Maximum Classifier Discrepancy [32], and Deep Adaptation Networks [33] have been proposed to minimize domain discrepancies. In the context of pixel-wise classification methods, traditional UDA methods like Fourier Domain Adaptation (FDA) [34] and domain adversarial training with Gradient Reversal Layers (GRL) [35] along with contrastive learning provide a foundation for recent UDA methods [36], [37].

UDA for BEV domain is under-explored, with DualCross [12] and DA-BEV [38] being the only methods. DualCross employs a knowledge distillation scheme and a two-stage training process, leveraging cross-modal knowledge from LiDAR. DA-BEV, on the other hand, applies self-training with features from both the image and BEV spaces, along with adversarial learning in both spaces. We focus on a flexible camera-only framework that can be applied to various existing architectures by leveraging pseudo-labels from

unlabeled PV images.

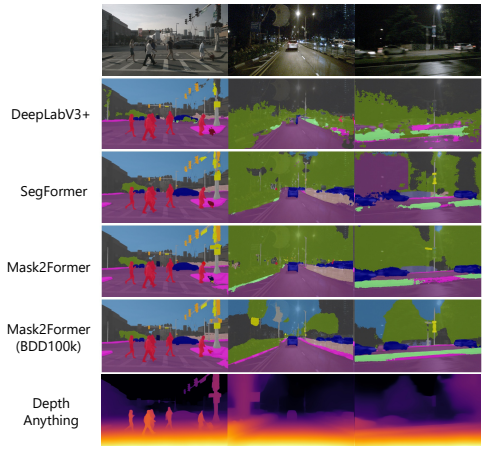

Fig. 2: **Visualization of pseudo-labels generated by easily accessible models on the nuScenes dataset.** We generate semantic segmentation pseudo-labels from pretrained models; namely DeepLabV3+ [8], SegFormer [9], and Mask2Former [10]. Out of the predictions trained on Cityscapes [5] (rows 2 to 4), Mask2Former exhibits the cleanest results especially for harder domains like nighttime. When trained on BDD100k [6], which has diverse scenarios such as weather and time of day, the pseudo-label are qualitatively better, where over- and under-segmentation occurs less frequently. We also explore the use of relative depth pseudo-labels obtained from Depth Anything [39]. Best viewed in color and zoomed in.

III. APPROACH

This work addresses the task of BEV segmentation of street-view scenes from multi-camera rigs, focusing on leveraging the abundance of unlabeled data to enhance model performance. We approach this through the paradigms of Semi-Supervised Learning (SSL) and Unsupervised Domain Adaptation (UDA), utilizing a combination of labeled and unlabeled datasets during training. As explained before, SSL and UDA aim to improve model performance by utilizing both labeled and unlabeled data sets during training.

For coherent methodology, we unify the two tasks, where we have a labeled set $\mathbb{L}$ and an unlabeled set $\mathbb{U}$. For SSL, the labeled set $\mathbb{L}$ and the unlabeled set $\mathbb{U}$ are from the same domain, with the number of labeled samples being much smaller than the number of unlabeled samples, i.e., $|\mathbb{L}| \ll |\mathbb{U}|$. In UDA, the labeled set $\mathbb{L}$ is from the source domain, while the unlabeled set $\mathbb{U}$ is from the target domain, with a domain shift between the two. The labeled set contains a set of multi-camera data $\mathbf{X}^L$ and BEV ground truth (GT) $\mathbf{y}^L$ where $(\mathbf{X}^L, \mathbf{y}^L) \in \mathbb{L}$, while the unlabeled set only contains multi-camera data $\mathbf{X}^U \in \mathbb{U}$.

We define $\mathbf{X} = \{\mathbf{X}_i\}_{i=1}^N = \{\mathbf{I}_i, \mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}_{i=1}^N$ as a batch of multi-view images, where $\mathbf{I}_i$ is the image, $\mathbf{K}_i$ is the intrinsic parameter, $\mathbf{R}_i$ and $\mathbf{t}_i$ are the extrinsic for the i-th camera in the batch, and N is the number of cameras on the vehicle. Following [15], we treat the BEV segmentation task as multi-label classification where each pixel can have multiple labels. Therefore, BEV GT is formalized as $\mathbf{y}^L \in \{0,1\}^{H \times W \times C}$, where H and W are the height and width of the BEV grid, and C is the number of classes.

Multi-camera BEV segmentation models typically comprise the following components: an Image Encoder, a 2D-to-BEV Module, and a BEV Encoder and Decoder, with the BEV Encoder sometimes omitted. Methods like CVT [16] have an additional BEV embedding as an input, but we omit these method-specific components in our discussion for brevity. The processing flow of these modules is as follows:

$$\mathbf{F}_{image} = \text{ImageEncoder}(\mathbf{I}) \tag{1}$$

$$\mathbf{f}_{bev} = \text{2DtoBEV}(\mathbf{F}_{image}, \mathbf{K}, \mathbf{R}, \mathbf{t}) \tag{2}$$

$$\mathbf{f}'_{bev} = \text{BEVEncoder}(\mathbf{f}_{bev}) \tag{3}$$

$$\hat{\mathbf{y}} = \text{BEVDecoder}(\mathbf{f}'_{bev}), \tag{4}$$

where $\mathbf{F}_{image}$ is the image features obtained from multi-camera images, $\mathbf{f}_{bev}$ is the BEV feature, and $\hat{\mathbf{y}}$ is the output BEV prediction.

For brevity, we simplify our notation by treating $\mathbf{X} = \mathbf{I}$ in subsequent sections and assume that the correct camera parameters are provided to the 2DtoBEV module.

Our method is organized as follows:

- Section III-A introduces the Perspective Cue Training (PCT) framework, which leverages pseudo-labels of perspective view task to utilize unlabeled data for BEV segmentation.
- Section III-B presents Camera Dropout (CamDrop), a novel input perturbation technique for multi-camera BEV segmentation.
- Our training strategy with PCT for UDA is formalized in Section III-C.
- Finally, Section III-D details our SSL approach, which incorporates BEV Feature Dropout (BFD) and a teacher-student network training strategy.

A. Perspective Cue Training (PCT) Framework

We propose the Perspective Cue Training (PCT) framework, which utilizes the PV pseudo-labels to train the BEV segmentation model in multi-task learning manner. This framework is only applied during the training stage of the target BEV segmentation model. In our work we first generate pseudo-labels of perspective images from all sets $\mathbf{X}^{L \cup U}$, including both labeled $\mathbf{X}^L$ and unlabeled images $\mathbf{X}^U$, where $\mathbf{X}^{L \cup U} \in (\mathbb{L} \cup \mathbb{U})$. We chose semantic segmentation for our main pseudo-labeling task because the pixel-wise classification task is similar to BEV segmentation. However, we have also experimented with relative depth estimation task, as shown in Figure 2. We generally use Mask2Former [10] trained on BDD100k [6] as our default pseudo-label generator, but we explore the effects of other network architectures and training datasets in our ablation studies. Our pseudo-label generator PLGen obtains $\tilde{\mathbf{P}} = \{\text{PLGen}(\mathbf{x}_i) \mid \forall \mathbf{x}_i \in \mathbf{X}^{L \cup U}\}$, where $\tilde{\mathbf{P}}$ are one-hot encoded pseudo-labels.

PV task head is applied to the image encoder, and the entire BEV segmentation model is trained in a multi-task learning manner, as shown in Figure 1. We utilize FPN with UPerNet [40] as our PV task head (PVHead) in the belief that utilizing all the hierarchical features of the image encoder is essential to condition the shared image encoder with PV

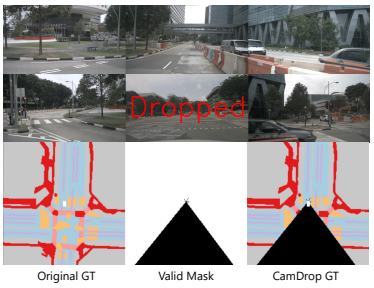

Fig. 3: **Visualization showcasing how Camera Dropout (CamDrop) augmentation is applied to perspective views and BEV ground truth (GT).** Back-viewing camera out of the six cameras is dropped and subsequent areas of the BEV GT, only visible by the dropped camera, are masked out.

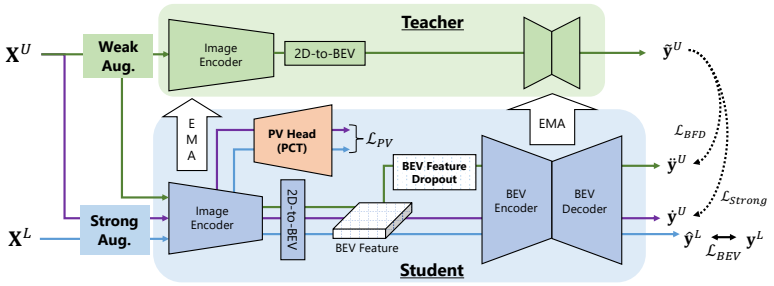

Fig. 4: **Our proposed semi-supervised learning (SSL) training framework utilizing the proposed PCT, Camera Dropout (CamDrop) augmentation, and BEV Feature Dropout (BFD).** The BEV segmentation model jointly trained with pseudo-labels obtained from the perspective view task models using PCT. We utilize the mean-teacher (MT) framework to effectively use the proposed input and feature perturbations by enforcing consistency with the teacher model.

cues. We can obtain PV predictions $\hat{\mathbf{P}}$ from the PV task head PVHead as follows:

$$\mathbf{F}^{L}_{image} = \text{ImageEncoder}(\mathbf{X}^{L}) \tag{5}$$

$$\mathbf{F}^{U}_{image} = \text{ImageEncoder}(\mathbf{X}^{U}) \tag{6}$$

$$\hat{\mathbf{P}} = \{\text{PVHead}(\mathbf{f}_i) \mid \forall \mathbf{f}_i \in [\mathbf{F}^{L}_{image}; \mathbf{F}^{U}_{image}]\}. \tag{7}$$

PV loss $\mathcal{L}_{PV}$ is computed using cross-entropy loss between the prediction $\hat{\mathbf{P}}$ and the pseudo-labels $\tilde{\mathbf{P}}$ as follows:

$$\mathcal{L}_{PV} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}(\hat{\mathbf{p}}_i, \tilde{\mathbf{p}}_i), \tag{8}$$

where $\hat{\mathbf{p}}_i \in \hat{\mathbf{P}}$ and $\tilde{\mathbf{p}}_i \in \tilde{\mathbf{P}}$ are the i-th output probability map of the PV prediction and one-hot encoded pseudo-labels, respectively.

The final multi-task training loss is as follows:

$$\mathcal{L}_{PCT} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV}, \tag{9}$$

where $\mathcal{L}_{BEV} = \text{FocalLoss}(\hat{\mathbf{y}}^{L}, \mathbf{y}^{L})$ and λ_{PV} is the weight for the PV loss. Note that $\hat{\mathbf{y}}^{L}$ is obtained from Equation 4.

B. Camera Dropout (CamDrop) Augmentation

Applying traditional pixel-wise input augmentations to BEV segmentation is challenging due to its 3D nature. For instance, masking sections of the PV image, as in Cutout, would require corresponding masking in the BEV ground truth, which is not straightforward. To address this, we introduce Camera Dropout (CamDrop), a simple yet effective input perturbation inspired by Cutout [23]. CamDrop randomly drops cameras and masks the associated visible areas in the BEV ground truth, as illustrated in Figure 3. This augmentation is efficient, as the camera parameters can easily determine the horizontal viewport. The perspective view image and the masked BEV area are labeled with ignore labels, ensuring that the dropped regions do not contribute to the loss. Importantly, we only mask regions exclusively visible from the dropped cameras, ensuring that the masked areas are not visible from the remaining cameras.

C. Training Method for UDA

The PCT framework provides a method of training on both domains using a shared image encoder, which motivates the model to learn domain-invariant features through PV tasks. The intuition can be explained using the theorem proposed in [**?**], which states that the upper-bound of the target domain error is composed of the source domain error and the domain discrepancy where the latter can be minimized through utilizing both domains with $\mathcal{L}_{PV}$.

For UDA, we formalize labeled set $\mathbb{L}$ as the source domain set and unlabeled set $\mathbb{U}$ as the target domain set. The loss for training UDA is as follows:

$$\mathcal{L}_{UDA} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV}, \tag{10}$$

where we use the same loss as in Equation 9.

Additionally, we utilize CamDrop to further enhance the model's robustness.

D. Training Method for SSL

For SSL, we introduce a teacher-student training framework following the Mean-Teacher (MT) framework [19], as shown in Figure 4. In this framework, we have two identical models, the student and the teacher, where the teacher model is an exponential moving average (EMA) of the student model, computed as:

$$\theta'_{teacher} = \alpha\theta_{teacher} + (1-\alpha)\theta_{student}, \tag{11}$$

where θ is the model parameters and α is the momentum.

For the teacher network, a weakly augmented input is used, while a strongly augmented input is passed to the student network. The consistency loss is computed between the weakly augmented teacher's prediction and the strongly augmented student's prediction:

$$\tilde{\mathbf{y}}^{U} = \text{Teacher}(\text{WeakAug}(\mathbf{X}^{U})) \tag{12}$$

$$\dot{\mathbf{y}}^{U} = \text{Student}(\text{StrongAug}(\mathbf{X}^{U})) \tag{13}$$

$$\mathcal{L}_{Strong} = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}^{U}_{i} - \dot{y}^{U}_{i})^2, \tag{14}$$

where we calculate the mean squared error for all $M = H \times W \times C$ pixels of the BEV grid. For WeakAug, we employ augmentations commonly used in BEV segmentation:

random horizontal flip, rotation, scaling, and cropping. For StrongAug, we apply the same augmentations as WeakAug but with ColorJitter, GaussianBlur, and CamDrop.

We further propose a feature perturbation called BEV Feature Dropout (BFD) inspired by UniMatch [26]. We apply Dropout [41] of 50% to the BEV feature maps to obtain perturbed BEV features. The consistency loss is then computed between the weakly augmented teacher's predictions and the perturbed prediction:

$$\mathbf{f}^{U}_{BEV} = \text{2DtoBEV}(\text{ImageEncoder}(\text{WeakAug}(\mathbf{X}^{U}))) \quad (15)$$

$$\ddot{\mathbf{y}}^{U} = \text{BEVDecoder}(\text{BEVEncoder}(\text{BFD}(\mathbf{f}^{U}_{BEV}))) \quad (16)$$

$$\mathcal{L}_{BFD} = \frac{1}{M}\sum_{i=1}^{M}(\tilde{y}^{U}_{i} - \ddot{y}^{U}_{i})^{2}. \quad (17)$$

The final loss used for SSL is as follows:

$$\mathcal{L}_{SSL} = \mathcal{L}_{BEV} + \lambda_{PV}\mathcal{L}_{PV} + \lambda_{Strong}\mathcal{L}_{Strong} + \lambda_{BFD}\mathcal{L}_{BFD}, \quad (18)$$

where λ_{Strong} and λ_{BFD} are the weights for strong consistency loss and the BFD consistency loss, respectively.

We note that BFD's benefits are primarily realized through MT framework, which empirically showed strong performance in SSL, but not for UDA.

IV. EXPERIMENTS

A. Experimental Setup

SSL Experimental Setup. We generate four SSL splits for 1/16, 1/8, 1/4, and 1/2 labeled data from the nuScenes dataset's [3] training set and treat the remaining labeled data as unlabeled data. Note that we divide the training scenes into labeled and unlabeled data, not individual sample frames. For testing, we evaluate on the validation scenes, which is the same for all SSL splits.

UDA Experimental Setup. We follow the splits introduced by DualCross [12] and use the following domain gaps: Day $\rightarrow$ Night, Dry $\rightarrow$ Rain, and Boston $\rightarrow$ Singapore for the nuScenes dataset. We added Singapore $\rightarrow$ Boston domain gap since Boston $\rightarrow$ Singapore contains mixed domain gaps, as explained in [12].

Common Experimental Setup. Following [15], we segment static categories, which includes the following: Drivable Area, Pedestrian Crossing, Walkway, Stop Line, Carpark Area, and Divider. We measure the performance using mean Intersection over Union (mIoU).

Implementation Details. We base our BEV segmentation code base on [15] and mmsegmentation [11]. The hyperparameters are all consistent to ensure fair comparison across different methods. Unless explicitly stated, the crop size is 224×480, batch size is 32, total training iteration is $30k$, the optimizer is AdamW with a learning rate of 0.004 and weight decay of 0.01, and the learning rate is scheduled using OneCycle Learning Rate Scheduler. For all experiments, we present the metric of the final checkpoint. The baseline method of our work is LSS [13] with EfficientNet-b4 backbone. For loss weights in Equations 10 and 18, we set $\lambda_{PV} = 0.1$, $\lambda_{Strong} = 0.1$, and $\lambda_{BFD} = 0.5$. As stated in Section III-D, the weak augmentations are random horizontal flip, rotation, scaling, and cropping, while the default strong augmentations are the same as the weak augmentations but with ColorJitter and GaussianBlur. For UDA, we use the strong augmentations without GaussianBlur. The baseline methods use strong augmentations by default unless explicitly stated. The momentum for the teacher model is set to $\alpha = 0.999$. Based on the original Mean-Teacher implementation, we utilize a sigmoid rampup function for the consistency loss weight, which starts at 0 and gradually increases to 1 over the first $9k$ iterations. We train and validate all of our experiments using 8 NVIDIA V100 GPUs.

TABLE I. Semi-Supervised Learning results on the nuScenes dataset. We experimented on four levels of labeled data and evaluated on the validation set. "CamDrop" is our proposed Camera Dropout and "BFD" is the BEV Feature Dropout. The results are in mIoU (%) and the best results are in **bold**.

Method	CamDrop	BFD	1/16	1/8	1/4	1/2
Sup. Only			21.9	27.4	36.4	47.0
MT			23.7	29.8	37.5	47.9
	✓		25.5	31.4	38.6	47.5
		✓	25.2	31.8	39.3	49.1
	✓	✓	27.0	32.6	40.1	49.1
UniMatch	✓	✓	22.9	29.3	39.2	49.0
PCT			24.1	29.3	37.6	48.7
	✓		24.9	30.0	38.3	48.4
PCT+MT	✓		28.6	34.0	40.7	50.4
	✓	✓	**29.6**	**35.0**	**41.9**	**51.6**

Pseudo-label Generation. The nuScenes dataset does not have semantic segmentation annotations for the PV images. We utilized mmsegmentation [11] to generate PV pseudo-labels. We retrained Mask2Former with the same configuration as the Cityscapes dataset on datasets without pretrained weights. For relative depth estimation, we use the publicly available code provided by Depth Anything [39].

B. Semi-Supervised Learning Results

In our SSL benchmark, we compare our proposed method against the following baselines:

- **Sup. Only**: LSS supervised with only the labeled data
- **Mean-Teacher (MT)**: [19] adapted for BEV segmentation (base model is LSS)
- **UniMatch**: [26] adapted for BEV segmentation with our proposed CamDrop and BFD (base model is LSS)

Note that SSL for multi-camera BEV segmentation is under-explored and there are no existing methods for comparison. We also apply CamDrop and BFD to MT to show the effectiveness of our proposed input and feature perturbations.

In Table I, we show the results of our SSL experiments on the nuScenes dataset. Compared to the supervised only method, our proposed PCT+MT method outperforms the baseline by a significant margin for all SSL splits. It can also be seen that PCT alone can perform better than supervised only baseline and can even perform competitively against SSL methods like MT and UniMatch. PCT, when combined with MT and strong perturbations (CamDrop and BFD), achieves the best performance for all SSL splits, improving on the difficult $1/16$ split by a significant margin of 7.7%. We obtain similar gains with PCT+MT with CamDrop and BFD on the Argoverse2 dataset, as shown in Appendix-A.

TABLE II. Unsupervised Domain Adaptation results on the nuScenes dataset. We experimented with four different domain gaps and evaluated on the validation set. The results are in IoU (%) and the best results are in **bold**.

Method	Drive.	Cross.	Walk.	Stop.	Car.	Div.	Mean
Day → Night							
Baseline	30.5	1.7	4.0	1.9	0	11.8	8.3
DomainAdv	47.1	16.1	10.7	5.7	0	11.2	15.1
FDA+MT	44.9	7.8	12.3	4.6	0	14.1	14.0
PCT	51.3	19.4	**16.1**	**7.6**	0	19.3	19.0
PCT+CamDrop	**52.5**	**19.8**	15.8	6.8	0	**20.6**	**19.2**
Dry → Rain							
Baseline	74.2	38.2	46.8	31.3	39.6	32.7	43.8
DomainAdv	72.0	39.8	42.0	33.7	38.9	33.6	43.3
FDA+MT	75.3	42.0	47.0	35.3	39.5	34.2	45.6
PCT	**78.3**	**45.2**	52.1	**37.6**	47.2	36.4	49.5
PCT+CamDrop	**78.3**	44.7	**52.6**	37.2	**48.7**	**37.3**	**49.8**
Singapore → Boston							
Baseline	39.5	3.1	12.8	4.1	1.0	12.1	12.1
DomainAdv	35.7	4.2	11.3	4.8	0.6	9.7	11.1
FDA+MT	41.8	6.5	14.5	7.1	1.1	10.8	13.6
PCT	47.0	8.0	19.3	6.3	0.7	13.7	15.8
PCT+CamDrop	**48.9**	**8.9**	**21.6**	**7.6**	**1.7**	**15.6**	**17.4**
Boston → Singapore							
Baseline	37.1	7.5	9.4	4.6	4.4	11.8	12.5
DomainAdv	40.0	8.3	11.7	4.8	2.2	11.6	13.1
FDA+MT	38.9	8.3	12.3	5.2	2.0	11.6	13.1
PCT	46.2	8.6	**14.2**	6.4	3.7	15.0	15.7
PCT+CamDrop	**47.2**	**9.0**	14.1	**7.7**	**4.8**	**16.4**	**16.5**

CamDrop and BFD have been shown to be effective in improving the performances of baseline and proposed method. MT with CamDrop improves MT on most splits, and MT with BFD improves MT on all splits. As shown in Figure 5, MT with CamDrop and BFD improves the performance of MT for the difficult $1/16$ split, especially for areas further away. The BEV segmentation predictions for PCT+MT with CamDrop and BFD, show improvements for dividers and smaller categories like "Stop Line."

C. Unsupervised Domain Adaptation Results

In our UDA benchmark, we compare our proposed method against the following baselines:

- **Baseline**: LSS supervised with only the source domain
- **DomainAdv**: Adversarial baseline used in [12], which adds domain classifiers to the Image Encoder and BEV Encoder and trained with GRL [35] (base model is LSS)
- **FDA+MT**: Fourier Domain Adaptation with Mean-Teacher [34] (base model is LSS)

In Table II, we show the results of our UDA experiments on the nuScenes dataset. The FDA+MT baseline improves upon the baseline LSS and DomainAdv in most domain gaps. Although DomainAdv works well in the difficult Day → Night and Boston → Singapore domain gaps, it fails to perform well on others. We believe this is due to the domain classifier not functioning correctly for the other domain gaps.

For all domain gaps, both of our proposed PCT and PCT+CamDrop significantly outperform baselines, especially for major categories like "Drivable Area" and "Walkway." As we show in Figure 5, PCT has a clearer BEV segmentation for the difficult Day → Night domain gap than the rest of the baselines, as there are fewer segmentation artifacts. The addition of CamDrop helps improve PCT in all domain gaps. We also show the efficacy of PCT on the Argoverse2 dataset in Appendix-A.

TABLE III. We evaluated pseudo-labels generated from different semantic segmentation architectures. The segmentation models are trained on the Cityscapes dataset and the "Segmentation Quality" is mIoU of the Cityscapes validation split. The experiments for SSL and UDA are conducted with 1/16 labeled split and Day → Night domain gap, respectively.

Pseudo-Labeling Model	Seg. Quality (mIoU)	SSL **1/16**	UDA **Day → Night**
DeepLabV3+ [8]	79.5	23.3	11.7
Segformer [9]	82.3	23.2	15.6
Mask2Former [10]	83.7	23.8	17.2

TABLE IV. We evaluated different datasets for training semantic segmentation models for generating pseudo-labels. Cityscapes [5], BDD100k [6], and Synthia [7] are all perspective view semantic segmentation dataset. We also report the relative depth pseudo-label generated from Depth-Anything [39].

Pseudo-Label Task	Training Dataset (or Foundation Model)	SSL **1/16**	UDA **Day → Night**
Sem. Seg.	Cityscapes	23.8	17.2
	BDD100k	24.1	19.0
	Cityscapes+BDD100k	24.1	18.4
	BDD100k+Synthia	23.8	17.6
Rel. Depth	Depth Anything	22.8	19.0

D. Ablation Study

Effect of Pseudo-Label Quality. Table III evaluates pseudo-labels generated from different semantic segmentation architectures trained on the Cityscapes dataset. Higher-quality pseudo-labels result in better performance for UDA, but the difference is not as significant for SSL. Although not experimented with, carefully annotated PV images may further improve the capability of the PCT.

Effect of Pseudo-Label Dataset. In Table IV, we evaluated different datasets for training semantic segmentation models for generating pseudo-labels. Training with pseudo-labels generated from BDD100k, known for containing various domain variations, performed the best in SSL and UDA. While we hoped for joint datasets (e.g. Cityscapes+BDD100k) to further improve the performance, the results are similar to BDD100k alone. PCT with relative depth pseudo-labels does not perform as well for SSL, but performs comparable to semantic segmentation with BDD100k for UDA. With that being said, the performance disparities between the training datasets and tasks are relatively minor, and the choice of dataset and pseudo-labeling task is flexible for PCT.

Effect of Crop Size for PCT. In Table V, we evaluated different crop sizes for training with PCT. In our PV Head, the resolution depends on the crop size, and the results show that the higher crop size results in better performance for both SSL and DA. Lower crop size results in lower resolution and less context for the semantic segmentation model to learn to produce robust feature maps.

Effect of PCT on Different BEV Architectures. We show the effect of PCT on different BEV architectures in Table VI. The image encoder and hyperparameters are consistent

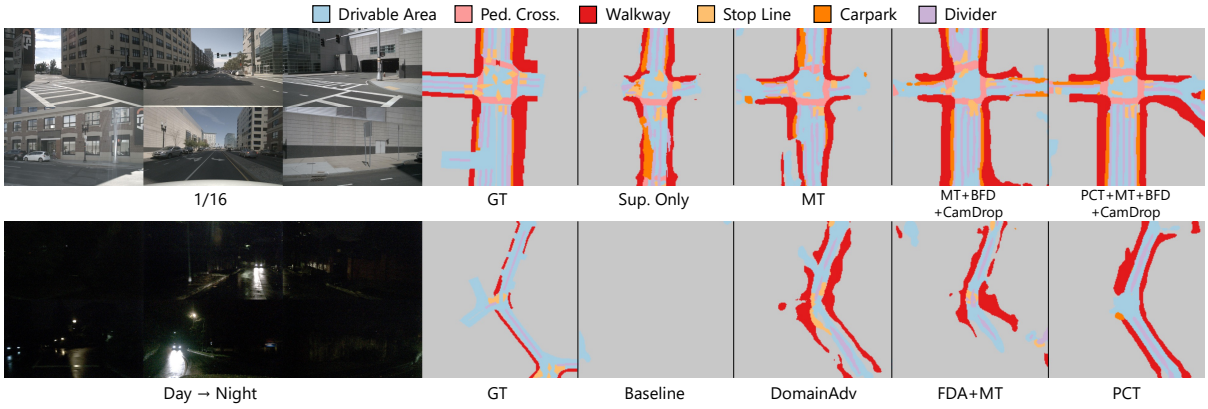

Fig. 5: **Qualitative results for semi-supervised learning (SSL) on the 1/16 split and unsupervised domain adaptation (UDA) on the "Day → Night" split. Best viewed in color and zoomed in.**

TABLE V. Different crop sizes for training with PCT.

Crop Size	SSL **1/16**	UDA **Day → Night**
128×352	20.2	16.9
224×480	24.1	19.0
360×720	24.3	19.8

TABLE VI. Effect of PCT on different BEV architectures.

BEV Architecture	PCT	SSL **1/16**	UDA **Day → Night**
LSS [13]		21.9	8.3
	✓	**24.1**	**19.0**
CVT [16]		14.4	9.1
	✓	**16.3**	**15.2**
BEVFusion [15]		21.4	7.1
	✓	**23.0**	**19.1**
GKT [17]		15.5	10.2
	✓	**16.5**	**14.3**

across different BEV architectures. Notably, all the BEV architectures benefit from PCT, and the performance improvements are consistent across different BEV architectures. The results show that our proposed PCT framework is flexible and boosts the base model performances. Additionally, PCT does not increase the network parameters in any way during inference time, as we can freely remove the PV Head.

Effect of maximum camera drops. Table VII shows the effect of varying number of cameras to drop for CamDrop. For both UDA and SSL, dropping one camera results in significant performance gains compared to no drop, and further dropping results in worse returns. For methods which does not utilize MT, such as PCT, the CamDrop may be too strong when dropping more than one camera.

E. Comparisons with Other UDA BEV Methods

Here, we compare our method against DualCross [12], which utilizes cross-modality information with a LiDAR teacher. The experimental setup used for DualCross is drastically different, but we made our network and hyperparameters consistent with the ones used in DualCross for a fair comparison. More specifically, they use a modified LSS architecture with Efficient-b0. The results are shown in Table VIII. DualCross uses a smaller crop size of 128×352, and our method can perform comparable to their method without needing multi-modal information. However, our method can perform superior to DualCross when utilizing a larger crop size of 224×480 since the PV Head greatly benefits from a larger crop size. DA-BEV [38] is another UDA method, but its implementation details are unclear (e.g., UDA splits), and their code is not publicly available at the time of writing.

TABLE VII. Effect of varying number of cameras to drop for CamDrop. The # Max Drops refers to the maximum number of cameras which can be randomly dropped at once.

# Max Drops	SSL (MT) **1/16**	UDA (PCT) **Day → Night**
0	23.7	19.0
1	**25.5**	**19.2**
2	25.1	18.8
3	25.1	18.3
4	24.9	18.3
5	22.4	18.2

TABLE VIII. We compare our method against DualCross. "Crop Size" is the input PV image size. "Modal" refers to the modalities used for training where **C** refers to camera and **L** refers to LiDAR. The results are in IoU (%) and the best results are in **bold**.

Method	Crop Size	Modal	Road	Lane	Vehicle
Day → Night					
DualCross	128×352	C+L	51.8	16.9	17.0
PCT+CamDrop	128×352	C	49.5	17.8	18.3
	224×480	C	**56.3**	**21.2**	**22.3**
Dry → Rain					
DualCross	128×352	C+L	71.9	19.5	29.6
PCT+CamDrop	128×352	C	76.3	32.8	27.2
	224×480	C	**79.3**	**36.0**	**31.5**
Boston → Singapore					
DualCross	128×352	C+L	43.1	**15.6**	20.5
PCT+CamDrop	128×352	C	44.0	13.0	19.7
	224×480	C	**47.8**	**15.6**	**21.8**

V. CONCLUSION

This work presents the Perspective Cue Training (PCT) framework, a novel training framework that effectively leverages unlabeled perspective view (PV) images through multi-tasking with PV tasks to address the challenges of limited BEV annotations and domain shifts. Our experiments showed that PCT with pseudo-labels generated from publicly available models for semantic segmentation and depth estimation shows promising results for both semi-supervised

learning (SSL) and unsupervised domain adaptation (UDA). Our approach is flexible and applicable to various existing BEV architectures and demonstrates significant improvements over baseline methods. The introduction of Camera Dropout (CamDrop) and BEV Feature Dropout (BFD) further enhances the performance of our method.

APPENDIX

A. Results for Argoverse2

TABLE IX. Semi-Supervised Learning results on the Argoverse2 dataset. The results are in mIoU (%). The best results are in **bold**.

Method	1/16	1/8	1/4	1/2
Sup. Only	35.9	42.6	48.1	53.2
PCT+MT+CamDrop+BFD	**40.9**	**45.6**	**51.2**	**54.2**

TABLE X. Unsupervised Domain Adaptation results on the Argoverse2 dataset. We experimented with two different "City-to-City" domain gaps and evaluated on the validation set. The results are in IoU (%) and the best results are in **bold**.

Method	Drivable.	Ped. Cross.	Divider	Mean
Palo Alto → Miami				
Baseline	50.1	9.5	17.7	25.8
PCT	**54.9**	**12.6**	**19.3**	**28.9**
Austin → Pittsburgh				
Baseline	47.2	7.3	27.2	27.2
PCT	**51.0**	**13.8**	**27.6**	**30.8**

We provide additional results for SSL and UDA on the Argoverse2 [4] dataset. We utilize static categories, including Drivable Area, Pedestrian Crossing, and Divider. LSS is used as the base model and the hyperparameters across the experiments are consistent with our other experiments.

For SSL, we experimented with four levels of labeled data splits, similar to our nuScenes experiment, and evaluated on a common validation set. The SSL results are provided in Table IX. Similar to our nuScenes results, our method outperforms the supervised only baseline for all SSL splits.

For UDA, we experimented with two different "City-to-City" domain gaps: Palo Alto → Miami and Austin → Pittsburgh. The UDA results are provided in Table X. The results show that our method outperforms the baseline for both domain gaps, demonstrating the effectiveness of our method for UDA on another dataset.

REFERENCES

[1] S. Chen, Y. Zhang, *et al.*, "Vma: Divide-and-conquer vectorized map annotation system for large-scale driving scene," *arxiv preprint arXiv:2304.09807*, 2023.

[2] J. Zhang, S. Chen, *et al.*, "A vision-centric approach for static map element annotation," *arxiv preprint arXiv:2309.11754*, 2023.

[3] H. Caesar, V. Bankiti, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *CVPR*, 2019.

[4] B. Wilson, W. Qi, *et al.*, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," *arxiv preprint arXiv:2301.00493*, 2023.

[5] M. Cordts, M. Omran, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[6] F. Yu, H. Chen, *et al.*, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *CVPR*, 2020.

[7] G. Ros *et al.*, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016.

[8] L.-C. Chen, Y. Zhu, *et al.*, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[9] E. Xie, W. Wang, *et al.*, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, 2021.

[10] B. Cheng, I. Misra, *et al.*, "Masked-attention mask transformer for universal image segmentation," in *CVPR*, 2021.

[11] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.

[12] Y. Man, L. Gui, *et al.*, "Dualcross: Cross-modality cross-domain adaptation for monocular bev perception," in *IROS*, 2023.

[13] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *ECCV*, 2020.

[14] Y. Li, Z. Ge, *et al.*, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," in *AAAI*, 2022.

[15] Z. Liu, H. Tang, *et al.*, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *ICRA*, 2023.

[16] B. Zhou and P. Krahenbuhl, "Cross-view transformers for real-time map-view semantic segmentation," in *CVPR*, 2022.

[17] S. Chen, T. Cheng, *et al.*, "Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer," *arxiv preprint arXiv:2206.04584*, 2022.

[18] S. Borse, M. Klingner, *et al.*, "X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation," in *WACV*, 2023.

[19] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[20] Z. Ke, D. Wang, *et al.*, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *ICCV*, 2019.

[21] Z. Chen, L. Zhu, *et al.*, "A multi-task mean teacher for semi-supervised shadow detection," in *CVPR*, 2020.

[22] M. Xu, Z. Zhang, *et al.*, "End-to-end semi-supervised object detection with soft teacher," in *ICCV*, 2021.

[23] T. Devries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arxiv preprint arXiv:1708.04552*, 2017.

[24] S. Yun, D. Han, *et al.*, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019.

[25] G. French, S. Laine, *et al.*, "Semi-supervised semantic segmentation needs strong, varied perturbations," in *BMVC*, 2019.

[26] L. Yang, L. Qi, Litong, *et al.*, "Revisiting weak-to-strong consistency in semi-supervised semantic segmentation," in *CVPR*, 2022.

[27] K. Sohn, D. Berthelot, *et al.*, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *NeurIPS*, 2020.

[28] Y. Ouali, C. Hudelot, *et al.*, "Semi-supervised semantic segmentation with cross-consistency training," in *CVPR*, 2020.

[29] N. Gosala, K. Petek, *et al.*, "Skyeye: Self-supervised bird's-eye-view semantic mapping using monocular frontal view images," in *CVPR*, 2023.

[30] J. Zhu, L. Liu, *et al.*, "Semi-supervised learning for visual bird's eye view semantic segmentation," *arxiv preprint arXiv:2308.14525*, 2023.

[31] E. Tzeng, J. Hoffman, *et al.*, "Adversarial discriminative domain adaptation," in *CVPR*, 2017.

[32] K. Saito, K. Watanabe, *et al.*, "Maximum classifier discrepancy for unsupervised domain adaptation," in *CVPR*, 2017.

[33] M. Long, Y. Cao, *et al.*, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[34] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *CVPR*, 2020.

[35] Y. Ganin, E. Ustinova, *et al.*, "Domain-adversarial training of neural networks," *JMLR*, 2015.

[36] L. Hoyer, D. Dai, *et al.*, "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation," in *CVPR*, 2021.

[37] X. Zhao, N. C. Mithun, *et al.*, "Unsupervised domain adaptation for semantic segmentation with pseudo label self-refinement," *arxiv preprint arXiv:2310.16979*, 2023.

[38] K. Jiang, J. Huang, *et al.*, "Da-bev: Unsupervised domain adaptation for bird's eye view perception," *arxiv preprint arXiv:2401.08687*, 2024.

[39] L. Yang, B. Kang, *et al.*, "Depth anything: Unleashing the power of large-scale unlabeled data," *arxiv preprint arXiv:2401.10891*, 2024.

[40] T. Xiao, Y. Liu, *et al.*, "Unified perceptual parsing for scene understanding," in *ECCV*, 2018.

[41] N. Srivastava, G. E. Hinton, *et al.*, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, 2014.