SLiDE: Self-supervised LiDAR De-snowing through Reconstruction Difficulty

Gwangtak Bae[1], Byungjun Kim[1], Seongyong Ahn[2],
Jihong Min[2], and Inwook Shim[3]*

[1]Seoul National University,[2]Agency for Defense Development,[3]Inha University
{tak3452,peterbj95,seongyong.ahn,happymin77}@gmail.com, iwshim@inha.ac.kr

Abstract. LiDAR is widely used to capture accurate 3D outdoor scene structures. However, LiDAR produces many undesirable noise points in snowy weather, which hamper analyzing meaningful 3D scene structures. Semantic segmentation with snow labels would be a straightforward solution for removing them, but it requires laborious point-wise annotation. To address this problem, we propose a novel self-supervised learning framework for snow points removal in LiDAR point clouds. Our method exploits the structural characteristic of the noise points: low spatial correlation with their neighbors. Our method consists of two deep neural networks: Point Reconstruction Network (PR-Net) reconstructs each point from its neighbors; Reconstruction Difficulty Network (RD-Net) predicts point-wise difficulty of the reconstruction by PR-Net, which we call *reconstruction difficulty*. With simple post-processing, our method effectively detects snow points without any label. Our method achieves the state-of-the-art performance among label-free approaches and is comparable to the fully-supervised method. Moreover, we demonstrate that our method can be exploited as a pretext task to improve label-efficiency of supervised training of de-snowing.

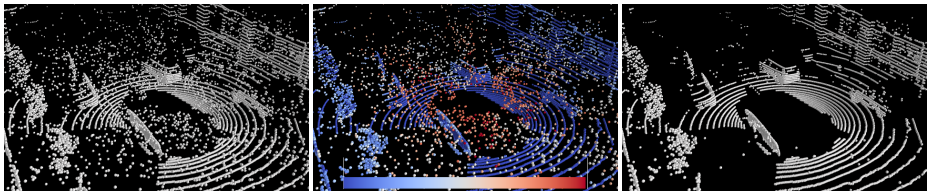
Keywords: LiDAR de-snowing, vision for adverse weather, self-supervised de-snowing

1 Introduction

Robust and accurate 3D scene measurement is an essential component of outdoor machine perceptions, e.g., autonomous vehicles. LiDAR is a commonly used 3D measurement sensor that gives reliable 3D point clouds in favorable weather conditions. However, in snowy weather conditions, LiDAR frequently generates a large number of particle noise points by detecting solid snowflakes [42]. These noise points could have a fatal impact on point cloud applications for outdoor systems [14, 22, 50].

Conventional filter-based approaches [5, 40, 47] have been presented for the LiDAR de-noising task to alleviate this problem. They attempt to remove the noise points by evaluating their spatial vicinity, but these approaches often suffer

* Inwook Shim is the corresponding author.



(a) Noisy point cloud (b) Reconstruction difficulty (c) De-snowed point cloud

Fig. 1: An example of the proposed LiDAR de-snowing process estimating how difficult to reconstruct each point from neighboring points. In (b), the closer to the red, the more difficult to reconstruct.

from misclassification since they only rely on simple spatial sparsity. Following the success of deep learning in various 3D point cloud applications (e.g., classification [43, 44], detection [49, 59], segmentation [27, 38], etc.), a deep learning-based LiDAR de-noising approach, WeatherNet [16], is recently introduced. It takes advantage of deep learning-based semantic segmentation methods to detect point-wise LiDAR noise points.

While WeatherNet [16] outperforms the conventional approaches by a significant margin, it requires point-wise annotations. Even though there were attempts on efficient 3D point annotation [32, 33, 41], manual annotation of 3D point cloud is still laborious and time-consuming [10]. Moreover, labeling snow noise points requires even more efforts. Labeling the snow points is hard to take advantage of the assistance of camera view, which is the convention in point cloud labeling process [4, 11, 52], since camera images are also heavily degraded, e.g., snowflakes on the lens and in the air. Besides, in most cases, the degradation is not consistent with the noise points of LiDAR.

To address this issue, we propose a self-supervised learning framework for de-noising LiDAR point clouds in snowy weather, i.e., *LiDAR de-snowing*. We focus on the structural characteristic of the noise points in snowy weather that they have low spatial correlations with their neighbors. Therefore, they are difficult to be reconstructed from their neighboring points. Based on this insight, we propose a novel self-supervised learning approach for snow points removal in LiDAR point clouds. Our method consists of two deep neural networks: Point Reconstruction Network (PR-Net) reconstructs randomly selected target points from their neighbors; Reconstruction Difficulty Network (RD-Net) predicts point-wise error of PR-Net reconstruction, which we call *reconstruction difficulty*. An example of the estimated reconstruction difficulty and the de-snowing result is visualized in Fig. 1. A set of reconstruction target points for training is selected from the point cloud itself, which leads to a self-supervised training scheme. The two deep neural networks are jointly trained with a shared loss function, and then only RD-Net, followed by simple post-processing, is used to detect the noise points in the inference step. Our model trained without any labeled data

outperforms previous label-free approaches with a large margin and achieves a comparable performance to the fully supervised model.

While our self-supervised method is basically label-free, it can be extended to a semi-supervised training scheme where training data consist of a small amount of labeled data and a large amount of unlabeled data. We demonstrate that an extension of our method as a pretext task enables supervised training of de-snowing in a label-efficient manner. Our semi-supervised extension yields better performance with fewer labeled data, which shows that our method as a pretext task is well-aligned with supervised training of de-snowing.

Our contributions can be summarized as follows:

- To the best of our knowledge, we are the first to introduce a self-supervised approach for LiDAR de-noising under snowy weather, i.e., *LiDAR de-snowing*. Our method identifies noise points that have low spatial correlations with their neighboring points.
- Our method outperforms previous label-free approaches by a large margin and achieves comparable results to the supervised method.
- We demonstrate that our self-supervised approach can be exploited as a pretext task for supervised de-snowing, which largely improves label-efficiency.

2 Related Work

2.1 LiDAR Point Cloud De-noising

Conventional filter-based methods classify noise points by their spatial sparsity [5, 40, 46, 47, 58]. Radius outlier removal (ROR) [47] finds the number of neighbors within a fixed radius and classifies points as noises if the number of neighbors is lower than a predefined threshold. ROR often fails at far-distant points because the density of the LiDAR point cloud decreases as the distance increases. Dynamic radius outlier removal (DROR) [5] employs varying search radii according to the distance to overcome the limitation of the previous works and successfully removes snow noise points. While those filter-based methods are easy to use and have a low computational burden, they still cannot handle point cloud's irregularity well, which often results in a significantly worse performance.

Recently, a supervised learning-based approach, WeatherNet [16], is introduced. It formulates the LiDAR de-noising problem as 2D semantic segmentation on the range image representation. Despite WeatherNet outperforms the filter-based methods, it requires expensive point-wise annotated training data.

2.2 Self-supervised Image De-noising

Self-supervised deep learning approaches have accomplished remarkable advances in the image de-noising task. Noise2Noise [31] is a pioneering work that restores a noisy image using another noisy image generated from the same clean image source. Since it is difficult to obtain a pair of noisy images in dynamic scenes,

following works [1, 25] improved the idea to work with a single noisy image by predicting a de-noised version of each pixel without depending on the pixel itself.

Despite the success of self-supervised de-noising in the image domain [1, 23, 25, 31, 45] and dense point cloud domain [17, 34, 35], there are problems in utilizing those methods for the LiDAR de-noising task in snowy weather, i.e., the *de-snowing* task. First, the image de-noising cannot produce sufficiently reliable results when a pixel value is difficult to be precisely predicted by its neighboring pixel information [25]. Second, the objective of LiDAR de-noising [5, 40, 46, 47] mainly focuses on removing noise points, not restoring the original points. For many applications that use LiDAR data (e.g., grasping an object, avoiding obstacles), rather than restoring the clean version of every elements, which is the main objective of the image de-noising tasks, it is essential to discard unreliable measurements while preserving clean measurements.

2.3 Semi-supervised Learning

Semi-supervised learning is a training scheme to learn from both labeled data and unlabeled data [54]. How to design an unsupervised loss function for leveraging sufficient unlabeled data is the main concern of semi-supervised learning. The categories of semi-supervised learning methods include unsupervised pretraining followed by fine-tuning [7,19,28,56], consistency regularization [26,48,53], pseudo labeling [18, 29, 55], and combination of these methods [2, 51, 60]. While our method is label-free, i.e., self-supervised, we demonstrate that our method can be extended to a semi-supervised training scheme. Combining our self-supervised loss with a supervised loss from a limited number of labeled data, the large performance gain shows that our method as a pretext task is well-aligned with supervised learning, which leads to an effective semi-supervised training scheme.

3 Proposed Method

Section 3.1 describes the representation of input LiDAR data. Section 3.2 explains our self-supervised learning framework for detecting noise points. Section 3.3 and 3.4 give detailed information on multi-hypothesis point reconstruction and post-processing process, respectively. Section 3.5 explains the extension of our self-supervised method into a semi-supervised training scheme.

3.1 Input Representation

The input to the proposed method is a 2D range image representation, which is the raw data structure of rotating LiDAR [9,37,38], e.g., Velodyne LiDAR. Such representation simplifies the 3D position reconstruction problem into 1D depth reconstruction along the LiDAR rays. The range image is generated as follows: Let P be a finite LiDAR point cloud, which contains K points: $P = \{\mathbf{p}_1, ..., \mathbf{p}_K\}$. Each point $\mathbf{p} = (p_x, p_y, p_z)$ is projected onto the image plane as $\mathbf{u} = (u, v)$ via a mapping function $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$. By following [16, 37], column u is defined

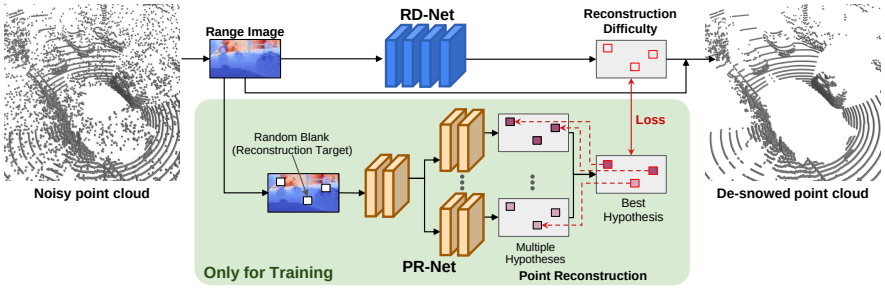


Fig. 2: The overall structure of our proposed LiDAR de-snowing method.

by $u = (\pi - arctan(p_y, p_x))/\delta_h$, where δ_h is the horizontal resolution of the LiDAR we used. Row v represents the laser id of $\mathbf{p}$, which corresponds to one of the sender/receiver modules in LiDAR. The projected point at (u, v) has a range value $r = (p_x^2 + p_y^2 + p_z^2)^{1/2}$. For each scan of LiDAR, we generate a corresponding range image $\mathrm{R} \in \mathbb{R}^{\mathrm{n}\times\mathrm{m}}$.

3.2 Self-supervised Learning Framework

We propose a self-supervised approach of LiDAR de-snowing that does not require point-wise annotations. To detect noise points that have low spatial correlations to their neighboring points, we designed a point reconstruction task and utilized errors from the task as guidance for training our de-snowing network. As shown in Fig. 2, reconstruction target points are randomly selected in the range image for every iteration of training. The Point Reconstruction Network (PR-Net) then predicts depth values of the target points by aggregating information of their neighboring points. At the same time, the Reconstruction Difficulty Network (RD-Net) estimates the extent of errors of the points reconstructed by PR-Net, which we call ***reconstruction difficulty***.

The two deep neural networks, PR-Net and RD-Net, are jointly trained with a shared loss function. We design a loss function with two objectives. First, as seen in the left figure in Fig. 3, the loss function should guide RD-Net to produce high output for noise points on which PR-Net has high error. Second, when training the point reconstruction task, the loss function should attenuate the loss contribution of noise points because they are extremely difficult to be reconstructed. It is to make PR-Net concentrate on reconstructing non-noise points. Our baseline loss function is given as follows:

$$\mathcal{L}_{self} = \sum I \odot \left[\sqrt{2} \frac{\left| \theta(\widetilde{R}) - R \right|}{\exp(\phi(R))} + \phi(R) \right], \tag{1}$$

where R is a range image of LiDAR data and $\widetilde{R}$ is a randomly blanked range image. I is a binary mask that only selects loss from the blanked points. $\odot$ is an

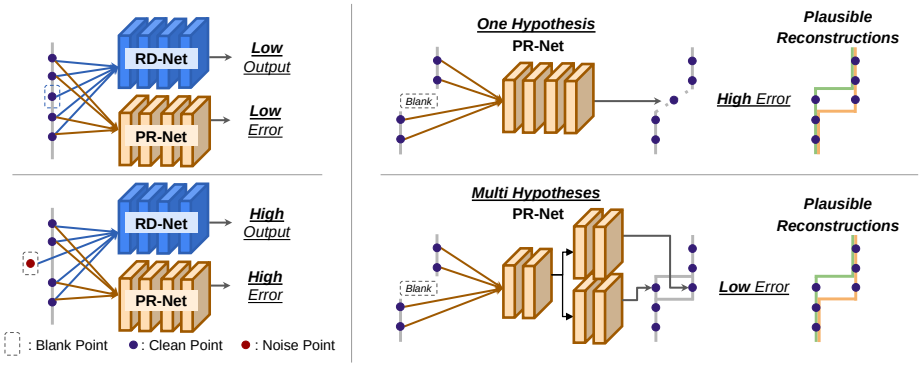


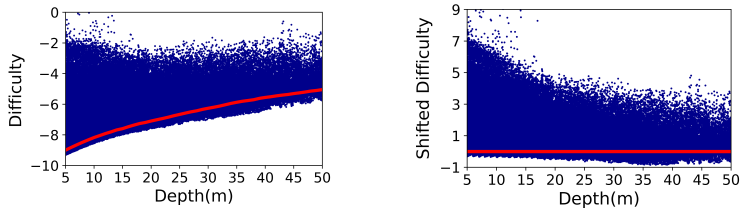
Fig. 3: The left image explains how PR-Net and RD-Net infer differently depending on whether a clean point or a noise point is blanked. The right image presents an example of ambiguity inherent in the point reconstruction task. PR-Net with multiple hypotheses can infer both of the multiple plausible reconstructions.

element-wise multiplication. $\theta(\cdot)$ and $\phi(\cdot)$ denote PR-Net and RD-Net, respectively. The blanked points are selected randomly for each iteration of training. The structure of Eq. (1) is inspired by the negative log-likelihood of a Laplacian distribution [21, 24].

PR-Net's reconstruction error guides RD-Net to learn to estimate the reconstruction difficulty of each point. PR-Net takes the blanked range image $\widetilde{R}$ as an input and predicts the depth value of the blanked points, and then we calculate $L1$ loss. Eq. (1) guides $\exp(\phi(R))$ to be high when the reconstruction error $L1$ is expected to be high, and to be low for the opposite. In other words, RD-Net is trained to predict the expected reconstruction quality of PR-Net for each point by taking the original range image R, which is not blanked, as an input. Consequently, we can identify the noise points by only utilizing the prediction of RD-Net. The regularization term $\phi(R)$ is added to prevent RD-Net from predicting an infinite value.

Eq. (1) also allows training of PR-Net to be robust to noise points. Trying to minimize the loss on noise points could decrease the reconstruction ability for clean points. Therefore, in Eq. (1), gradients from noise points are attenuated by $\exp(\phi(R))$ as a weighting factor for the loss given to PR-Net. Points with high output from RD-Net have a smaller effect on the loss of PR-Net.

At the test time, only RD-Net is used to detect noise points. It takes a range image R as an input and predicts how difficult it will be to reconstruct each point. If the output of RD-Net $\phi(R)$ is higher than a certain threshold, the point is classified as a noise point.



(a) Before post-processing (b) After post-processing

Fig. 4: RD-Net's output of each point in order of depth before (a) and after post-processing (b). Red lines indicate the 20^{th} percentile of each meter of depth.

3.3 Point Reconstruction with Multiple Hypotheses

PR-Net is trained to reconstruct randomly selected target points and their reconstruction errors are computed. However, the point reconstruction task has an inherent ambiguity, where some clean points may have multiple plausible answers for reconstruction. For example, in the right of Fig. 3, when a point on the object boundary is selected as a target, there can be two plausible answers: the object boundary or background. Since PR-Net with a single output cannot cover both answers precisely, the output is collapsed to the mean of them, which leads to an undesirable increase of the reconstruction error for clean points.

In order to distinguish the ambiguous clean points and noise, PR-Net is modified to have multiple outputs as shown in Fig. 2, which is inspired by multi-hypothesis learning. Following [30, 57], only predictions with a minimum error are used for loss calculation. Such prediction for each point is given as follows:

$$C_i = \min_k \left| \theta^k(\tilde{R}) - R \right|_i , \tag{2}$$

where $\theta^k(\cdot)$ is the k^{th} output, i.e., hypothesis, of PR-Net and C_i indicates the output of point i with a minimum error. Then, our reconstruction loss in Eq. (1) can be modified as follows:

$$\mathcal{L}_{self,mhl} = \sum I \odot \left[\sqrt{2} \frac{C}{\exp(\phi(R))} + \phi(R) \right] . \tag{3}$$

The modified loss guides PR-Net to have multiple plausible predictions rather than to be collapsed to the mean. If at least one of the multiple predictions is well-reconstructed, the reconstruction error will be low. However, since noise points are still difficult to reconstruct, albeit with a finite number of multiple predictions, reconstruction errors will be high.

3.4 Post-processing

As the sensing distance increases, the minimum output of RD-Net gradually increases, as shown in Fig. 4a. This can be seen as the effects of the decreased density of the point cloud at a far distance, which makes the reconstruction more

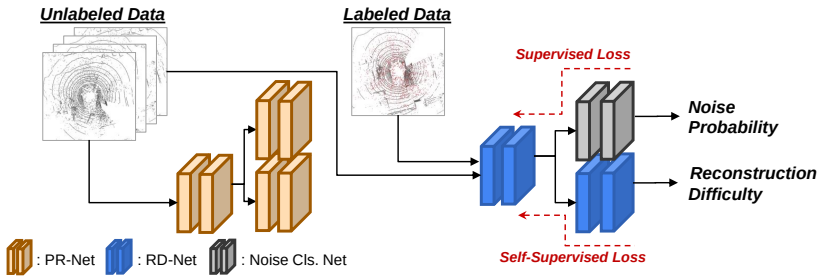
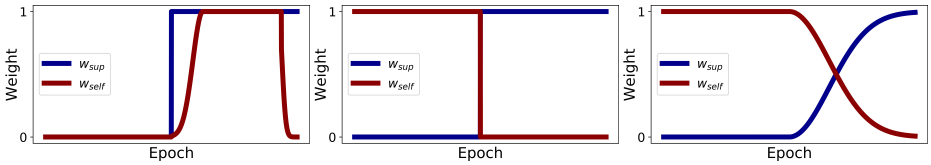


Fig. 5: Architecture design of our semi-supervised extension. The feature encoder of RD-Net and the noise classification network is shared.



(a) Ramp up/down (b) Pretrain (c) Smooth transfer

Fig. 6: Weighting functions for combining the supervised and self-supervised loss.

difficult. To compensate this effect, with partitioning the points by $1m$ depth interval, RD-Net's output is shifted downward in the amount of the shifting parameter for each depth interval. For example, in Fig. 4b, the shifting parameter is defined as the 20^{th} percentile of RD-Net's output value. It compensates the depth-dependent bias on reconstruction difficulty. We then detect the noise points based on a certain threshold that is empirically determined. The ablation studies on the shifting parameter are described in Section 4.1.

3.5 Semi-supervised Learning

While our self-supervised method is able to remove noise points without any labeled data, we found out that it can also be extended to a semi-supervised training scheme. Since RD-Net is trained to regress the reconstruction difficulty, which is one of the characteristics to distinguish noise points from others, we can expect that the backbone network of RD-Net extracts features suitable for noise classification. As shown in Fig. 5, the feature encoder of RD-Net is shared for the feature encoder of a noise classification network. All of the networks are jointly trained with the weighted sum of two loss functions as follows,

$$\mathcal{L} = w_{self} * \mathcal{L}_{self,mhl} + w_{sup} * \mathcal{L}_{sup}, \tag{4}$$

where L_{sup} is the cross entropy loss by following WeatherNet [16]. w_{self} and w_{sup} are the weights for the self-supervised loss and the supervised loss, respectively.

Three different weighting functions are proposed and evaluated. Fig. 6a indicates weighting functions that are widely used in consistency regularization methods for semi-supervised learning [20,26,36,53]. Weighting functions in Fig. 6b are inspired by self-supervised learning [6,8,12,13,39]. Our self-supervised method is regarded as a pretext task. The feature encoder of the noise classification network is initialized with the feature encoder of RD-Net. We further extend Fig. 6b into Fig. 6c. The pretrained features for estimating reconstruction difficulty are smoothly transferred into the noise classification network. The equations for the weighting functions are explained in the supplementary material.

4 Experimental Result

4.1 Point-wise Evaluation on Synthetic Data

Dataset A number of scene data with point-wise annotations should be provided for a fair quantitative evaluation, regardless of whether training is done in a supervised or an unsupervised manner. However, to the best of our knowledge, there is no published dataset that satisfies it. We collect real-snow noise points using our stationary data-capturing system and synthesize the collected noise points into various clean weather road scenes.

Since a background point cannot be detected if a noise point is on the same LiDAR ray, we can directly pick out noises by comparing range images of noisy point cloud sets (*Noise-Set*) and clean point cloud sets (*Clean-Set*) as follows:

$$L^N_{(u,v)} \leftarrow \begin{cases} \mathbf{N} \text{ (Noise)}, & \text{if } R^F_{(u,v)} \geq R^N_{(u,v)} + \tau \\ \mathbf{C} \text{ (Clean)}, & \text{else} \end{cases} \tag{5}$$

where R^F is the reference range image generated from *Clean-Set*, R^N is a range image of *Noise-Set*, L^N indicates a label map of R^N and τ is a margin for sensing errors of LiDAR. In the case of scanning an empty space (e.g., sky), since no valid background information is projected to $R^F_{(u,v)}$, we always assign the 'Noise' label to $L^N_{(u,v)}$. The collected noise points are then injected into road scene point cloud sets (*Base-Set*) taken in clean weather as follows:

$$R^S_{(u,v)} = \begin{cases} R^N_{(u,v)}, & \text{if } R^N_{(u,v)} \leq R^{max}_{(u,v)} \text{ and } L^N_{(u,v)} = \mathbf{N} \text{ (Noise)} \\ R^B_{(u,v)}, & \text{else} \end{cases} \tag{6}$$

where R^B is a range image in *Base-Set*, and R^S is a synthesized range image. R^{max} is the maximum detectable range which is introduced for a realistic synthesis [3, 16] by considering scene structures of R^B as follows:

$$R^{max}_{(u,v)} = \min\left(\frac{-ln(\frac{n}{I^B_{(u,v)}+g})}{2 * \beta}, R^B_{(u,v)}\right), \tag{7}$$

given the received laser intensity $I^B_{(u,v)}$, the adaptive laser gain g, the atmospheric extinction coefficient β, and the detectable noise floor n. In this paper, Nuscenes dataset [4] is used as *Base-Set* to synthesize snowy scenes. Please see the supplementary material for more details of the dataset generation process.

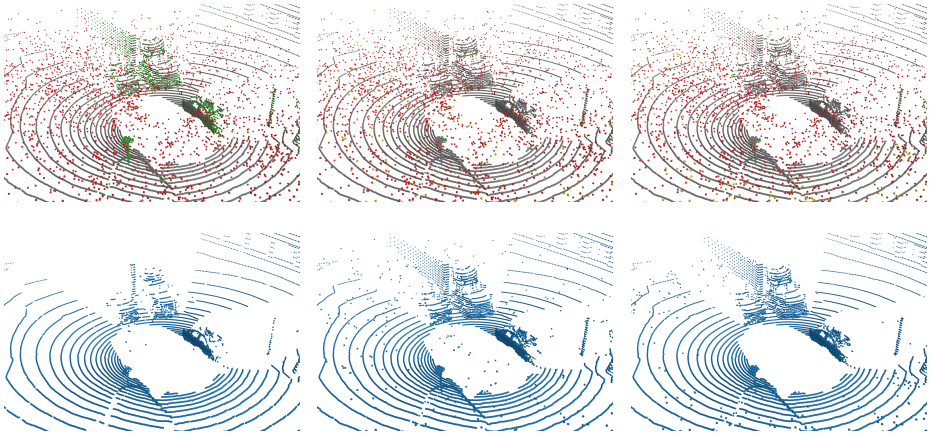
Table 1: Quantitative results on the synthesized snow noise data. *Labeled data* indicates the number of labeled training data used.

Method	Labeled Data	IoU	Precision	Recall
ROR [47]	0	17.82	17.86	98.65
DROR [5]	0	33.68	33.87	98.37
Ours w/o MHL	0	65.75	70.38	90.89
Ours	0	79.62	85.69	91.83
WeatherNet [16]	239	41.40	76.78	47.32
Ours(Semi.sup.)	239	82.44	96.39	85.07
WeatherNet [16]	23,908	84.04	97.48	85.90
Ours(Semi.sup.)	23,908	84.24	97.24	86.30

Implementation Details Our self-supervised model is trained with the synthesized snow noise dataset. We split a total of $34,139$ scans into training, validation, and test sets at a ratio of $70 : 15 : 15$. PR-Net and RD-Net consist of residual blocks [15] for the self-supervised model. For the semi-supervised model, PR-Net and RD-Net use the same backbone with WeatherNet to ensure a fair comparison. Separated layers in Fig. 5 consist of a *LiLaBlock* [41] and a convolution layer by following WeatherNet. Each training step encounters an independently selected random set of target points in order to learn various cases of the reconstruction. Horizontal flipping is randomly performed to augment training data. At the test time, points with an RD-Net output higher than the threshold are classified as noise points. Throughout experiments in this paper, the threshold is determined as 2.9, which achieves the highest performance for the validation dataset. Quantitative performance is evaluated using the Intersection-over-Union (IoU) metric, following WeatherNet [16].

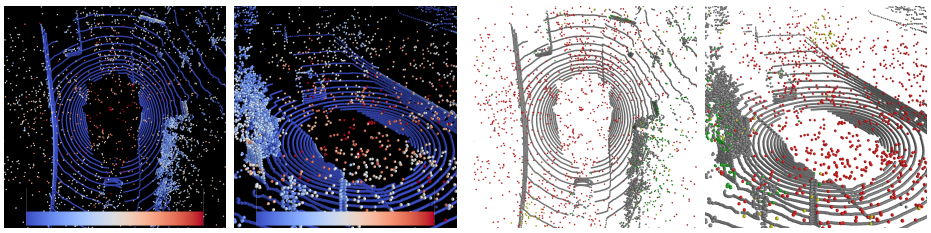
Quantitative Comparisons In Table 1, our proposed methods are compared with previous LiDAR de-noising approaches: ROR [47], DROR [5], and WeatherNet [16]. Table 1 shows that our proposed method yields a significantly higher IoU than the state-of-the-art label-free method, DROR. Notably, our approach achieves a comparable IoU to the supervised method without using any labeled data. Applying multi-hypothesis learning in Eq. (3) improved the baseline method in all metrics. For the case where only 1% of labeled data provided, the semi-supervised extension yields significantly higher performance than the supervised method, WeatherNet. Even when 100% labeled data provided, our method performs better than the supervised method without using any additional unlabeled data, which shows better exploitation of the same given data. *Ours (Semi-sup)* refers to the semi-supervised extension with *Smooth transfer* in Fig. 6, which is described in Section 3.5.

Table 2 shows an analysis on the performance changes of our method according to the noise level. The noise level is decided by following Canadian Adverse Driving Condition Dataset [42]. While WeatherNet and our semi-supervised



(a) DROR [5] (b) WeatherNet [16] (c) Ours

Fig. 7: Qualitative comparisons on the synthesized snow noise data. The first row shows all points with their prediction results (red: true positive, green: false positive, gray: true negative, yellow: false negative). Classification as noise corresponds to *positive*. The second row shows de-snowed point clouds. DROR misclassifies many sparsely distributed points at the side facade of the truck.



(a) Reconstruction difficulty (b) De-snowing result

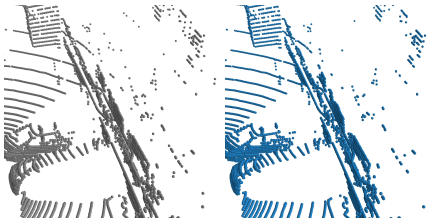
Fig. 8: Estimated reconstruction difficulty and its corresponding de-snowing result. In (a), the closer to the red, the higher reconstruction difficulty. In (b), the color of each points follows Fig. 7.

model generate relatively consistent performances when the noise level changes, all of the label-free methods including ours have lower IoU as the noise level decreases. Since unsupervised methods do not use direct supervision from point-wise labeled data, noise-like points among clean points can be misclassified as false positives. Fig. 9 depicts the case where noise-like clean points exist. Although only clean points are displayed, clean points floating in the air look very similar to noise points, and those are removed in Fig. 9b.

In Table 3, we evaluate three weighting functions proposed for our semi-supervised method in Fig. 6. All of the weighting functions have better performances than the supervised method when limited data are given. Compared to

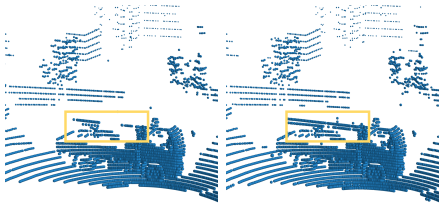
Table 2: Analyses on the performance changes according to the noise level.

Method	Metric	Noise Level			
		Light	Medium	Heavy	Extreme
DROR [5]	IoU	12.76	22.57	32.99	52.59
WeatherNet [16]	IoU	82.29	83.57	83.57	84.60
Ours(Semi-sup)	IoU	82.35	83.87	83.72	84.83
Ours	IoU	60.81	71.48	79.37	85.69
Ours	Precision	64.35	77.81	85.48	85.69
Ours	Recall	91.70	89.78	91.74	92.42



(a) Clean (b) De-snowed

Fig. 9: A de-snowing result of only-clean points scene. Some of floating clean points are similar to noises and eliminated by our method.



(a) w/o MHL (b) w/ MHL

Fig. 10: Comparison between the single hypothesis model and the multi hypotheses model.

the supervised method, *Ramp up/down*, which is widely used in semi-supervised methods, shows higher IoU when 1% and 10% labels are given but lower when sufficient labels are given, 100%. *Pretrain* yields better IoU than the supervised method even when 100% of labels are given. *Smooth transfer* shows the highest performance when 1% data are labeled and higher than the supervised method even when 100% of data are available for both of methods.

Ablation Studies In Table 4, we investigate performance changes of our self-supervised method according to variations in configurations. First, multi hypotheses learning yields significantly better performance than our baseline method that has a single hypothesis. Among the different number of hypotheses, the model predicting three hypotheses achieves the highest IoU. Second, we analyze the effects of the blank ratio for training. Our method yields the highest performance when 50% of points are blanked. Third, we look into the impact of different shifting parameters in the post-processing step. The experiment with the minimum RD-Net output value for each depth interval as the shifting parameter shows a slightly lower result while other settings achieve similar performance.

Qualitative Comparisons Fig. 7 demonstrates a qualitative analysis of our method with DROR and WeatherNet. First, DROR misclassifies clean points if

Table 3: De-snowing performances of the supervised method and our semi-supervised extension when limited labeled data are provided.

Method	IoU (1% labels)	IoU (10% labels)	IoU (100% labels)
WeatherNet [16]	41.40	78.75	84.04
Ramp up/down	79.39	81.73	82.57
Pretrain	62.46	82.87	84.86
Smooth transfer	82.44	83.70	84.24

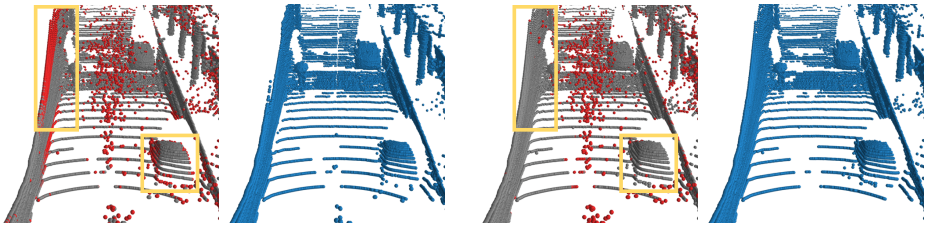
Table 4: Ablation experiments of the proposed self-supervised method.

# Hypotheses	1	2	3	4
IoU	65.75	76.17	79.62	76.52
Blank Ratio	10%	30%	50%	70%
IoU	76.64	78.93	79.62	76.57
Shifting Param.	min	10^{th}	20^{th}	30^{th}
IoU	72.63	79.36	79.62	79.22

they are sparsely distributed. Assuming the same distance, as an angle between a LiDAR ray and its hitting surface gets farther from perpendicular, a point density of the surface gets lower. As depicted in Fig. 7a, it leads to the failure case of DROR, which solely depends on sparsity for detecting noise. For example, in spite of strong semantic consistency of the side facade of the truck, DROR incorrectly removes the points on it based on sparsity. In contrast, our method preserves clean points if they can be reconstructed from neighboring points. Second, WeatherNet has the smallest number of false positives, which are marked as green points in Fig. 7. It still has remaining noise points around the LiDAR sensor. More results are in the supplementary material.

Fig. 8 visualizes the reconstruction difficulty estimated by RD-Net and its evaluation. Clean points have low reconstruction difficulty as they have high spatial correlations with neighbors. On the contrary, RD-Net assigns high difficulty to noise points. In Fig. 8a, points on trees have relatively high RD-Net output than the points on the road and cars. It shows that RD-Net successfully learns to reflect the reconstruction difficulty of each point. Although points on trees are inferred as more difficult points than other clean points, they still have lower difficulty than noise points and correctly classified as seen in Fig. 8b.

Fig. 10 demonstrates the effects of multi-hypothesis learning. When PR-Net infers a single hypothesis, many points on the upper side of a car are misclassified and removed, as shown in the yellow box in Fig. 10a. This is the case explained in Fig. 3. On the contrary, as seen in Fig. 10b, many of those points are preserved when multi-hypothesis outputs are inferred by PR-Net.



(a) DROR [5] (b) Ours

Fig. 11: Qualitative comparisons on the real-world snowy weather data (red: positive, gray and blue: negative).

4.2 Qualitative Evaluation on Real-world Data

We also evaluate our self-supervised method in real snowy weather scenarios captured by our mobility platform, equipped with a Velodyne 'VLS-128'. A total of 9,000 scans were captured in snowy weather. We qualitatively evaluate our method with DROR, which also does not require point-wise labels. Other experimental settings follow Section 4.1, except for the height of range images which is set to 128.

Fig. 11 shows a de-snowing result of DROR and ours. As 'VLS-128' LiDAR generates dense point clouds due to its high vertical resolution, the sparsity-based de-snowing method, DROR, generates fewer false positives than in Section 4.1. However, a number of clean points are still misclassified. For example, as shown in the yellow boxes in Fig. 11a, clean points on the car and the wall are filtered out when they have low spatial density. On the other hand, in Fig. 11b, since our method estimates point-wise reconstruction difficulty, clean points on the surface of objects are well preserved. More results are presented in the supplementary material.

5 Conclusion

In this work, we proposed a novel self-supervised method for de-snowing LiDAR point clouds in snowy weather conditions. By utilizing the characteristic of noise points that they have low spatial correlations with their neighboring points, our method is designed to detect noise points that are difficult to reconstruct from their neighboring points. Our proposed self-supervised approach outperforms the state-of-the-art label-free methods and yields comparable results to the supervised approach without using any annotation. Furthermore, we present that our self-supervised method can be exploited as a pretext task for the supervised training, which significantly improves the label-efficiency.

Acknowledgement This work was supported by the Agency for Defense Development (ADD) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1F1A1073505).

References

1. Batson, J., Royer, L.: Noise2self: Blind denoising by self-supervision. In: International Conference on Machine Learning. pp. 524–533 (2019)
2. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. Advances in Neural Information Processing Systems **32** (2019)
3. Bijelic, M., Gruber, T., Mannan, F., Kraus, F., Ritter, W., Dietmayer, K., Heide, F.: Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 11682–11692 (2020)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
5. Charron, N., Phillips, S., Waslander, S.L.: De-noising of lidar point clouds corrupted by snowfall. In: Conference on Computer and Robot Vision. pp. 254–261 (2018)
6. Chen, X., He, K.: Exploring simple siamese representation learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
8. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: IEEE International Conference on Computer Vision. pp. 1422–1430 (2015)
9. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. In: IEEE International Conference on Computer Vision. pp. 2918–2927 (2021)
10. Gao, B., Pan, Y., Li, C., Geng, S., Zhao, H.: Are we hungry for 3d lidar data for semantic segmentation? a survey of datasets and methods. IEEE Transactions on Intelligent Transportation Systems (2021)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (2012)
12. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
13. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33**, 21271–21284 (2020)
14. Gruber, T., Bijelic, M., Heide, F., Ritter, W., Dietmayer, K.: Pixel-accurate depth evaluation in realistic driving scenarios. In: International Conference on 3D Vision. pp. 95–105. IEEE (2019)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
16. Heinzler, R., Piewak, F., Schindler, P., Stork, W.: Cnn-based lidar point cloud denoising in adverse weather. IEEE Robotics and Automation Letters **5**(2), 2514–2521 (2020)

17. Hermosilla, P., Ritschel, T., Ropinski, T.: Total denoising: Unsupervised learning of 3d point cloud cleaning. In: IEEE International Conference on Computer Vision. pp. 52–60 (2019)
18. Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Label propagation for deep semi-supervised learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 5070–5079 (2019)
19. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: IEEE International Conference on Computer Vision. pp. 2146–2153. IEEE (2009)
20. Ke, Z., Wang, D., Yan, Q., Ren, J., Lau, R.W.: Dual student: Breaking the limits of the teacher in semi-supervised learning. In: IEEE International Conference on Computer Vision. pp. 6728–6736 (2019)
21. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems. pp. 5580–5590 (2017)
22. Kilic, V., Hegde, D., Sindagi, V., Cooper, A.B., Foster, M.A., Patel, V.M.: Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection. arXiv preprint arXiv:2107.07004 (2021)
23. Kim, K., Ye, J.C.: Noise2score: Tweedie's approach to self-supervised image denoising without clean images. Advances in Neural Information Processing Systems **34** (2021)
24. Klodt, M., Vedaldi, A.: Supervising the new with the old: learning sfm from sfm. In: European Conference on Computer Vision. pp. 698–713 (2018)
25. Krull, A., Buchholz, T.O., Jug, F.: Noise2void-learning denoising from single noisy images. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2129–2137 (2019)
26. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. arXiv preprint arXiv:1610.02242 (2016)
27. Landrieu, L., Simonovsky, M.: Large-scale point cloud semantic segmentation with superpoint graphs. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4558–4567 (2018)
28. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: International Conference on Acoustics, Speech and Signal Processing. pp. 8595–8598. IEEE (2013)
29. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3, p. 896 (2013)
30. Lee, S., Prakash, S.P.S., Cogswell, M., Ranjan, V., Crandall, D., Batra, D.: Stochastic multiple choice learning for training diverse deep ensembles. In: Advances in Neural Information Processing Systems. pp. 2119–2127 (2016)
31. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. In: International Conference on Machine Learning. pp. 2965–2974 (2018)
32. Luo, H., Wang, C., Wen, C., Cai, Z., Chen, Z., Wang, H., Yu, Y., Li, J.: Patch-based semantic labeling of road scene using colorized mobile lidar point clouds. IEEE Transactions on Intelligent Transportation Systems **17**(5), 1286–1297 (2015)
33. Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J.: Semantic labeling of mobile lidar point clouds via active learning and higher order mrf. IEEE Transactions on Geoscience and Remote Sensing **56**(7), 3631–3644 (2018)
34. Luo, S., Hu, W.: Differentiable manifold reconstruction for point cloud denoising. In: ACM International Conference on Multimedia. pp. 1330–1338 (2020)

35. Luo, S., Hu, W.: Score-based point cloud denoising. In: IEEE International Conference on Computer Vision. pp. 4583–4592 (2021)
36. Luo, Y., Zhu, J., Li, M., Ren, Y., Zhang, B.: Smooth neighbors on teacher graphs for semi-supervised learning. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 8896–8905 (2018)
37. Meyer, G.P., Laddha, A., Kee, E., Vallespi-Gonzalez, C., Wellington, C.K.: Lasernet: An efficient probabilistic 3d object detector for autonomous driving. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 12677–12686 (2019)
38. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4213–4220. IEEE (2019)
39. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision. pp. 69–84. Springer (2016)
40. Park, J.I., Park, J., Kim, K.S.: Fast and accurate desnowing algorithm for lidar point clouds. IEEE Access **8**, 160202–160212 (2020)
41. Piewak, F., Pinggera, P., Schafer, M., Peter, D., Schwarz, B., Schneider, N., Enzweiler, M., Pfeiffer, D., Zollner, M.: Boosting lidar-based semantic labeling by cross-modal training data generation. In: European Conference on Computer Vision Workshops. pp. 0–0 (2018)
42. Pitropov, M., Garcia, D.E., Rebello, J., Smart, M., Wang, C., Czarnecki, K., Waslander, S.: Canadian adverse driving conditions dataset. The International Journal of Robotics Research (2020)
43. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
44. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems **30** (2017)
45. Quan, Y., Chen, M., Pang, T., Ji, H.: Self2self with dropout: Learning self-supervised denoising from single image. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1890–1898 (2020)
46. Roriz, R., Campos, A., Pinto, S., Gomes, T.: Dior: A hardware-assisted weather denoising solution for lidar point clouds. IEEE Sensors Journal (2021)
47. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: IEEE International Conference on Robotics and Automation. pp. 1–4 (2011)
48. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. Advances in Neural Information Processing Systems **29** (2016)
49. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019)
50. Shim, I., Shin, S., Bok, Y., Joo, K., Choi, D.G., Lee, J.Y., Park, J., Oh, J.H., Kweon, I.S.: Vision system and depth processing for drc-hubo+. In: IEEE International Conference on Robotics and Automation. pp. 2456–2463 (2016)
51. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in Neural Information Processing Systems **33**, 596–608 (2020)

52. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)
53. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in Neural Information Processing Systems **30** (2017)
54. Van Engelen, J.E., Hoos, H.H.: A survey on semi-supervised learning. Machine Learning **109**(2), 373–440 (2020)
55. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 10687–10698 (2020)
56. Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O.: Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In: European Conference on Computer Vision. pp. 574–591. Springer (2020)
57. Yang, G., Hu, P., Ramanan, D.: Inferring distributions over depth from a single image. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 6090–6096 (2019)
58. Zhou, H., Chen, K., Zhang, W., Fang, H., Zhou, W., Yu, N.: Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1961–1970 (2019)
59. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)
60. Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. Advances in Neural Information Processing Systems **33**, 3833–3845 (2020)