Towards Point Cloud Counterfactual Explanations

Nicolas Vercheval and Aleksandra Pižurica

Department of Telecommunications and Information Processing, TELIN-GAIM

Ghent University, Belgium

***Abstract*—Counterfactual explanations, engineered alterations of classifier inputs that change predictions, are widely used for interpreting classifier decisions. In this work, we extend counterfactual generation to point cloud data, addressing the challenges posed by its unstructured nature. We introduce hierarchical modeling to enhance counterfactual learning, specifically the ability to identify and manipulate key semantic traits influencing classifier predictions. Through visual and quantitative evaluations, we demonstrate the effectiveness of our method in generating counterfactuals that successfully revert classifier predictions. Our implementation is publicly available on GitHub.**

***Index Terms*—counterfactuals, point cloud, VQVAE, XAI**

I. Introduction

Counterfactual explanations provide insights into a classifier's decision by applying the principles of counterfactual reasoning [1], [2]. They fabricate altered input examples to elicit a different classification outcome from the target classifier. Comparing original and altered inputs, particularly in visually interpretable image data, reveals which features most influenced the evaluation, offering the transparency that explainable AI (XAI) demands. This demand has driven the development of counterfactual methods for many 2D and 3D image data types [3]–[5], with point clouds being a notable exception.

Counterfactual explanation methods for images commonly rely on generative models to ensure realistic input alterations. For instance, VAEX [6] trains a variational autoencoder (VAE) [7] conditioned on classifier evaluations, enabling smooth transitions between target predictions via latent space manipulation. Although applicable to any data type, its application is often limited by the generative model's ability to represent the data. Point cloud data exemplifies this limitation, presenting a significant challenge for generative models due to the difficulty in capturing shape information from unstructured data points [8]. Developing counterfactual models for point clouds remains a significant research gap, especially given the growing reliance on multimodal data in the automotive industry, where point cloud data from LIDAR sensors is used for object detection and classification. Integrating counterfactual explanations could identify potential bias in pedestrian classification and other critical vulnerabilities before deployment; however, no such model has been proposed.

This paper introduces the first counterfactual model for point clouds, designed to fill this gap. We achieve this by developing a conditional generative model capable of effectively handling point clouds and integrating it into the VAEX framework, replacing the original conditional variational autoencoder, which is ineffective with this data. Furthermore, we propose a novel method for integrating counterfactual generation into generative model training. This results in realistic counterfactuals that gradually change a classifier's decision toward the target evaluation.

Unlike previous work in the VAEX framework [6], [9], which used classifier evaluations as direct input, we introduce an intermediate latent variable that captures the classification-related features, creating a hierarchical structure. This allows greater flexibility in feature representation and introduces noise during training, preventing over-reliance on input evaluations. During inference, we condition this latent variable with the target evaluation, enabling robust counterfactual explanations.

This strategy is more effective with a small latent space, as the model needs to rely on the conditional input for reconstruction and generation. For this purpose, we utilize PCGen [10], a recent state-of-the-art generative model that employs a double autoencoder architecture to compress point cloud data into a compact latent representation. Another advantage of PCGen is that the inner encoder's loss is defined on a discrete latent space, providing an additional semantic layer, rather than on the output space. Consequently, any deviation from the reconstruction due to the sampled conditional input is less penalized, allowing for more semantically meaningful counterfactual generation.

We evaluate our approach on ModelNet [11], a standard dataset for point cloud classification. We isolate two groups of labels that are frequently misclassified as each other and create two smaller datasets with only the selected classes. For each dataset, we train a counterfactual model that consistently changes the classifier's evaluation to the desired target. We perform quantitative experiments to assess the model's success in reverting classifier decisions. Additionally, we provide visual examples demonstrating the gradual change in point cloud representations as we shift the target evaluation. The contributions of this work can be summarized as follows:

- We introduce the first point cloud counterfactual method,

This work has been funded by the Fonds Wetenschappelijk Onderzoek (FWO) projects 1S89420N and G094122N and by the Flanders AI Research Programme grant no. 174K02325. Corresponding author: nicolas.vercheval@ugent.

to our knowledge.

- We propose hierarchical conditional generation for improved robustness and flexibility.
- We provide quantitative and visual evidence demonstrating the efficacy of the proposed method on a standard benchmark dataset.

The paper proceeds as follows: Section II briefly reviews relevant prior work, including concurrent explainability methods for point clouds. Section III details the technical novelties of our generative model and their importance for counterfactual generation. Section IV presents and discusses experimental results on a standard benchmark dataset. Finally, Section V concludes the paper, summarizing key findings and outlining future research directions.

II. Related Works

Explainability methods for black-box point cloud classifiers constitute a growing research area, with the development of numerous methods tailored to the specific characteristics of point cloud data. However, much concurrent research has focused on feature-importance methods. For instance, [12] highlights important points by measuring feature norms, [13] proposes a local surrogate for LIME [14], ablating information via point removal, while [15] builds on Grad-CAM [16] by incorporating local relations between the features. These methods, while valuable, typically provide a heatmap indicating the most influential points, but do not generate realistic input alterations, which offer a more complete explanation.

CausalPC [17] also models a causal framework for classification, but with a different scope, focusing on input-level robustness by modeling point cloud structure and mitigating hidden confounders like adversarial noise. In contrast, our approach models semantic features associated with classification decisions. While [18] is perhaps the closest work to counterfactual explanations, generating point clouds that maximize target classifier evaluations, their method searches for a global maximum of target activation through iterative input modifications. On the other hand, counterfactual methods are instead local, returning an alteration semantically related to the input.

III. Generating counterfactual point clouds

Counterfactual generation involves creating samples under alternative scenarios, where the data differs from the actual input. The challenge lies in the absence of ground truth for direct model training. Our solution consists of inferring latent features and dividing them into relevant and irrelevant to the classification, creating a meaningful latent space for manipulation.

Building upon VAEX [6], we exploit PCGen's high semantic compression and double semantic layer to allow for greater exploration by reducing reliance on explicit ground truth. Thanks to this advantage, we introduce a hierarchical structure to the VAEX counterfactual generation method, incorporating the conditional target into the probabilistic model and thereby improving the robustness of the counterfactuals.

The following sections expand on the modeling approach and detail how PCGen facilitates it.

A. Hierarchical Modeling

To understand how classification changes with sample alterations, counterfactual methods aim to generate altered samples highlighting the key features driving these changes. We achieve this by inverting cause and effect, modeling the conditional distribution $p(X|\text{class}(X))$, where $\text{class}(X)$ is the classifier evaluation, specifically the softmaxed output. We introduce two latent variables z_1, for semantic features unrelated to classification, and z_2, for features that are related to it and learn a generative model $p(X|z_1, \text{class}(X) = y)$, where y is the calculated evaluation. We propose the following probabilistic modeling:

$$p(X|\text{class}(X) = y) = p(X|z_1, z_2)p(z_2|\text{class}(X) = y). \quad (1)$$

This offers two main advantages. First, unlike previous work [6], [9], the model learns a probabilistic dependence on the classifier evaluation by sampling z_2, enhancing robustness towards the conditional input. Second, it allows more flexibility in the features related to the classifier inputs, which typically vary from sample to sample. This ultimately benefits the counterfactual production, which replaces the actual evaluation with a target one.

Specifically, we create counterfactuals as follows. First, we train a variational autoencoder to learn the posterior distribution $p(z_1|X)$ and $p(z_2|X, \text{class}(X) = y)$ through variational inference. Second, we change the original conditional input to the target evaluation $\hat{y}$ and calculate $p(z_1|X)$ and $p(z_2|X|\text{class}(X) = \hat{y})$, as well as their respective expectations, μ_1 and μ_2. This passage is crucial as it allows us to preserve the specific characteristics of a sample. In particular, μ_1 retains all the semantics unrelated to classification, while μ_2 encodes the semantics the given sample X would have if it had a different classification. Finally, we use the learned conditional distribution to create the counterfactual:

$$\hat{X} = p(X|\mu_1, \mu_2) \quad (2)$$

To implement the hierarchical modeling, we train a network to generate a Gaussian $p(z_2|\text{class}(X))$ and minimize the Kullback-Leibler divergence:

$$\mathcal{D}_{KL}(z_2|\text{class}(X), z_2|X, \text{class}(X)), \quad (3)$$

where the inferred $p(z_2|X, \text{class}(X))$ is also a Gaussian distribution. Note that while there we do not force the independence between z_1 and $\text{class}(X)$, in practice, this independence arises naturally from the constrained latent space, which discourages redundant information. Testing a specific loss to enforce their independence yielded no noticeable effect on the model.

This approach alone is insufficient, as classifiers often produce extreme evaluation values, limiting the exploration of more balanced values. Previous work [9] centralized the probabilities but did not clarify how to remap to multiple targets. Instead, we propose using the original logits of the classifier and dividing them by a temperature parameter before applying the softmax operation.

This modeling strategy is applicable to any data. At the same time, we find it particularly beneficial when the variational autoencoder encodes a preprocessed representation rather than raw data, as detailed in the following section.

B. Point cloud generation

Generating a distribution of points from a latent variable is not an easy task, especially when only low-sampled, raw point clouds are available. PCGen shows that VQVAE [19] can effectively encode a point cloud as a discrete latent w.

We leave the VQVAE model as is and consider its discrete latent space as building blocks for generating a point cloud, that can be assembled similarly for both reconstruction and counterfactual production. Instead, we focus on the inner VAE model, trained separately to compress w even further into a continuous variable z. From a probabilistic modeling point of view, this can be expressed with:

$$p(X|w, \text{class}(X) = y) = p(X|w)p(w|\text{class}(X) = y). \quad (4)$$

To learn the conditional model, we focus on the inner encoder. PCGen uses VampPrior [20] to provide a more flexible prior for the latent space at the cost of increased complexity. Instead, we introduce the z_2 variable as in (1) and use a standard Gaussian as a prior of z_1. We train the second autoencoder separately, using the classifier evaluations. All together we obtain:

$$p(X|\text{class}(X) = y) = \\ p(X|w)p(w|z_1, z_2)p(z_2|\text{class}(X) = y). \quad (5)$$

This approach takes advantage of the inner model's reconstruction loss, which is defined on the discrete latent space. This allows the VAE to focus on reconstructing a description of the point cloud, rather than the final shape, and it is less penalized when incorrectly reconstructing the specific shape, provided that the general semantics remain correct.

To enable gradual transitions, we consider target evaluations with varying intensities, denoted as τ, or counterfactual intensities. Specifically, we linearly interpolate between the original tempered probabilities, y, and the one-hot encoding of the target class $\hat{y}$, using τ as the interpolation factor:

$$\hat{y}_\tau = (1 - \tau)y + \tau\hat{y}, \quad \tau \in [0, 1]. \quad (6)$$

This interpolated vector serves as the conditional input for the inner decoder, generating a discrete latent variable that the outer decoder transforms into the counterfactual image.

IV. Experimental results

We test our method on the ModelNet [11] dataset, a classic benchmark for point cloud classification. We use a version where 2048 points have been uniformly sampled from the CAD models and use the same data to train the generative model. In particular, we do not use the original CAD models as ground truth for reconstruction, to better simulate the challenges of realistic point cloud datasets. We chose the popular DGCNN [21] architecture for the classifier and trained it on the full dataset. We observe that "desk" and "table" are frequently misclassified, as are "bottle", "bowl", "cup", and "vase". We extract these label groups, creating two smaller datasets: ModelNet-DT and Modelnet-BBCV. We retrain the DGCNN on the two smaller datasets to focus on the critical classes and apply the proposed method for the two new classifiers.

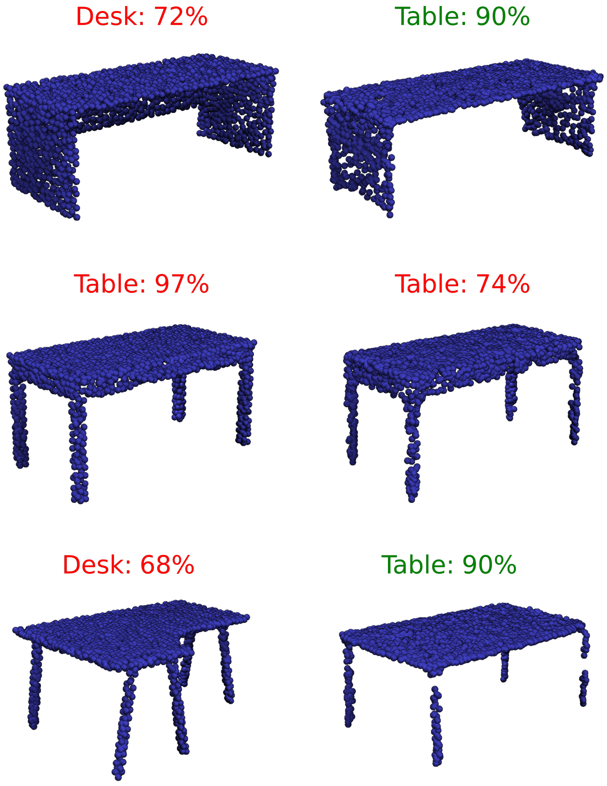

Fig. 1. Counterfactuals of misclassified examples from ModelNet-DT, with the predicted class and the associated classifier probability. Left: the original sample. Right: its counterfactual. Incorrect predictions are in red, correct predictions are in green.

Fig. 1 displays counterfactuals for misclassified examples from the ModelNet-DT test dataset, targeting the correct class. To observe subtle changes, we use a counterfactual value of $\tau = 0.5$. This often results in point clouds similar to the input, but with key modifications, as illustrated in the first row. The input sample, arguably mislabeled, is predicted as a desk instead of a table. Its counterfactual highlights potential reasons for this misclassification: the presence of a back panel, a reduced height, and an increased thickness of the horizontal panel. In the second row, the counterfactual remains misclassified as a desk. However, it now resembles a desk more closely, exhibiting smoother corners and curled legs. This highlights a potential blind spot in the model. Another blind spot is evident in the third row's counterfactual sample. Here, the shape of the counterfactual changes drastically as the target shifts towards the class "table", strongly suggesting the absence of corner tables in the training dataset.

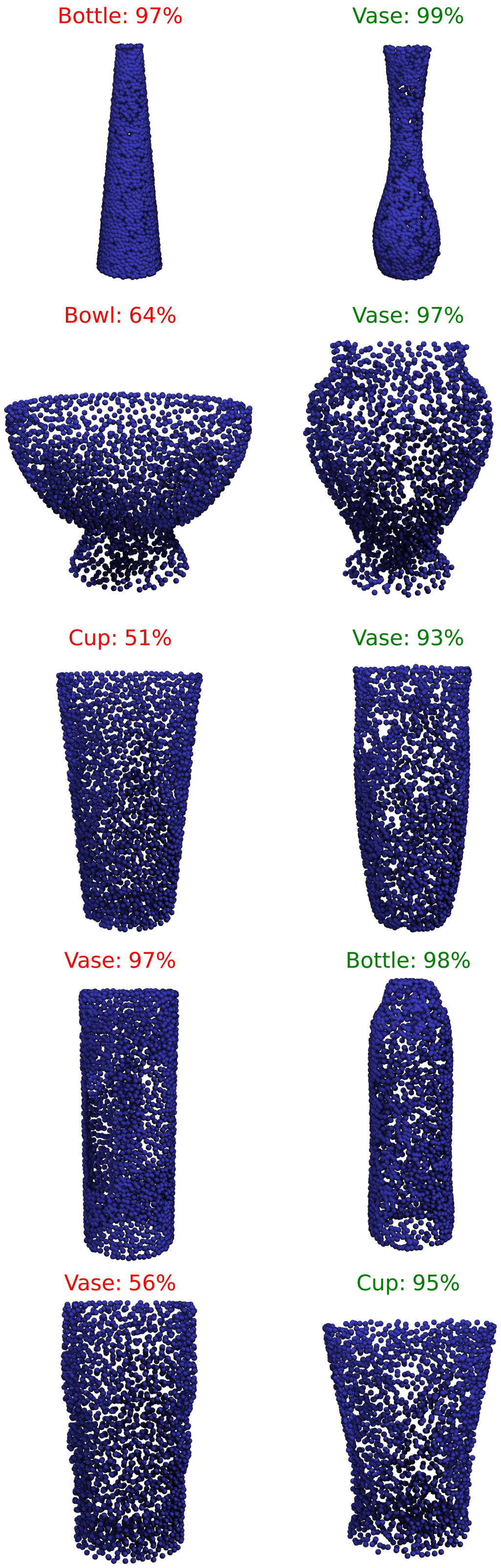

Fig. 2. Counterfactuals of misclassified examples from ModelNet-BBCV, with the predicted class and the associated classifier probability. Left: the original sample. Right: its counterfactual. Incorrect predictions are in red, correct predictions are in green.

TABLE I
COUNTERFACTUAL SUCCESS IN REVERTING CLASSIFIER DECISION

Dataset	$\tau = 0.5$	$\tau = 1$
ModelNet-DT	80%	99%
ModelNet-BBCV	86%	100%
ModelNet-DT (misclassified)	87%	97%
ModelNet-BBCV (misclassified)	95%	100%

Similarly, Fig. IV presents counterfactuals for misclassified examples from the ModelNet-BBCV test dataset, targeting the correct class. Focusing on the problematic 'vase' class, we include three examples where samples with other labels were misclassified as vases, and two misclassified vase examples (no vase was misclassified as a bowl). The classifier expects vases to be rounder with a wide opening. Bottles are detected from their bottlenecks, and cups are associated with a V-shape. Our model's ability to interpolate through different targets, while keeping recognizable traits, is demonstrated in Fig. 3.

Table I shows the counterfactual success rate, defined as the percentage of generated counterfactuals that successfully revert the classifier's decision to the target class. We also report statistics for counterfactuals of misclassified samples, which are crucial for analyzing classifier performance. We evaluate counterfactual intensities of $\tau = 0.5$ and $\tau = 1$. Our method is effective even with the lower intensity, while the higher intensity consistently reverts the decision. Notably, even when the lower intensity does not change the prediction, it still impacts associated probabilities, revealing potential model weaknesses while maintaining close similarity to the original sample. Conversely, an intensity of $\tau = 1$ may alter the sample excessively (Fig. 4). Table II presents the model's accuracy on the original dataset and on counterfactuals generated with a target $\bar{y}$ of equal probabilities across all classes. While our method doesn't completely eliminate classification-associated traits, which would lead to 50% accuracy for ModelNet-DT and to 25% for ModelNet-BBCV, it significantly reduces the original accuracy.

V. CONCLUSIONS AND FUTURE WORK

In this work, we presented the first method for generating counterfactual explanations for point cloud data. Our approach effectively alters classifier evaluations, typically reversing predictions, while preserving characteristics unrelated to the classification of the original samples. We demonstrated the utility of these counterfactuals in revealing classifier weaknesses, such as underrepresented data and challenges in capturing complex shapes. Future research will extend this method to more realistic datasets, particularly those from critical appli-

TABLE II
CLASSIFIER ACCURACY AFTER REMOVING CLASSIFICATION SEMANTICS

Dataset	Original	$\bar{y}$
ModelNet-DT	87%	73%
ModelNet-BBCV	89%	43%

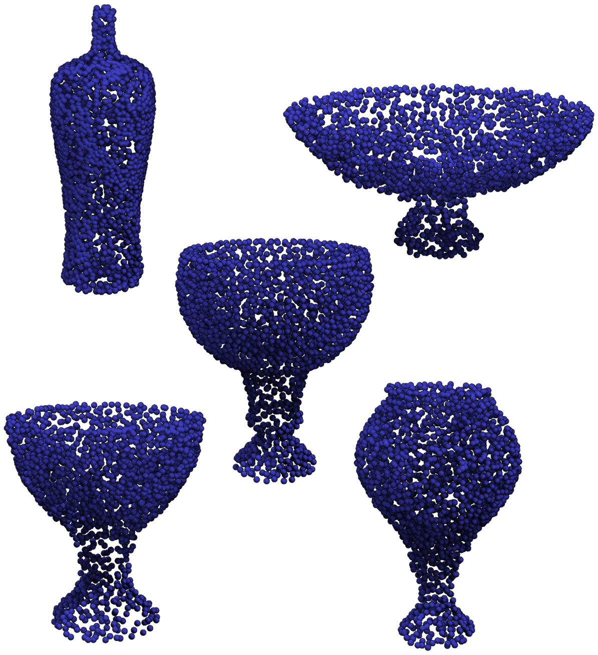

Fig. 3. Center: original sample (cup). Top left: counterfactual to bottle. Top right: counterfactual to bowl. Bottom left: accentuated cup. Bottom right: counterfactual to vase.

cations, and explore the generation of diverse counterfactuals to isolate specific causes of classifier evaluations.

References

[1] J. Pearl, *Causality*, 2nd ed. Cambridge University Press, 2009.

[2] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, and J. Jorge, "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications," *Information Fusion*, vol. 81, pp. 59–83, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521002281

[3] S. Mertes, T. Huber, K. Weitz, A. Heimerl, and E. André, "GANterfactual: Counterfactual explanations for medical non-experts using generative adversarial learning," *Frontiers in Artificial Intelligence*, vol. 5, p. 825565, Apr 8 2022.

[4] S. Singla, M. Eslami, B. Pollack, S. Wallace, and K. Batmanghelich, "Explaining the black-box smoothly—a counterfactual approach," *Medical Image Analysis*, vol. 84, p. 102721, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841522003498

[5] G. Guo, L. Deng, A. Tandon, A. Endert, and B. C. Kwon, "MiMICRI: Towards domain-centered counterfactual explanations of cardiovascular image classification models," in *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1861–1874. [Online]. Available: https://doi.org/10.1145/3630106.3659011

[6] N. Vercheval and A. Pizurica, "Hierarchical variational autoencoders for visual counterfactuals," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2513–2517. [Online]. Available: http://doi.org/10.1109/icip42928.2021.9506780

[7] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint*, vol. arXiv:1312.6114, 2013, [Online]. Available: https://arxiv.org/abs/1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114

[8] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 40–49. [Online]. Available: https://proceedings.mlr.press/v80/achlioptas18a.html

[9] N. Vercheval, M. Benčević, D. Muževič, I. Galić, and A. Pizurica, "Counterfactual functional connectomes for neurological classifier selection," in *2023 31st European Signal Processing Conference (EUSIPCO)*. IEEE, 2023, pp. 1050–1054. [Online]. Available: http://doi.org/10.23919/EUSIPCO58844.2023.10289859

[10] N. Vercheval, R. Royen, A. Munteanu, and A. Pizurica, "PCGen : a fully parallelizable point cloud generative model," *SENSORS*, vol. 24, no. 5, p. 30, 2024. [Online]. Available: http://doi.org/10.3390/s24051414

[11] Z. Wu, S. Song, A. Khosla, F. Yu, L.-l. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, oral Presentation. 3D Deep Learning Project Webpage.

[12] M. Y. Levi and G. Gilboa, "Fast and simple explainability for point cloud networks," *arXiv preprint*, vol. arXiv:2403.07706, 2024, available: https://arxiv.org/abs/2403.07706. [Online]. Available: https://arxiv.org/abs/2403.07706

[13] H. Tan and H. Kotthaus, "Surrogate model-based explainability methods for point cloud NNs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 2239–2248.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," *arXiv preprint*, vol. arXiv:1602.04938, 2016, [Online]. Available: https://arxiv.org/abs/1602.04938. [Online]. Available: https://arxiv.org/abs/1602.04938

[15] F. Matrone, M. Paolanti, A. Felicetti, M. Martini, and R. Pierdicca, "Bubblex: An explainable deep learning framework for point-cloud classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 6571–6587, 2022.

[16] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[17] Y. Huang, M. Zhang, D. Ding, E. Jiang, Z. Wang, and M. Yang, " CausalPC: Improving the Robustness of Point Cloud Classification by Causal Effect Identification ," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2024, pp. 19 779–19 789. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01870

[18] H. Tan, "Visualizing global explanations of point cloud DNNs," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023, pp. 4741–4750.

[19] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6309–6318.

[20] J. Tomczak and M. Welling, "VAE with a VampPrior," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Storkey and F. Perez-Cruz, Eds., vol. 84. PMLR, 09–11 Apr 2018, pp. 1214–1223. [Online]. Available: https://proceedings.mlr.press/v84/tomczak18a.html

[21] A. V. Phan, M. L. Nguyen, Y. L. H. Nguyen, and L. T. Bui, "Dgcnn: A convolutional neural network over large-scale labeled graphs," *Neural Networks*, vol. 108, pp. 533–543, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608018302636

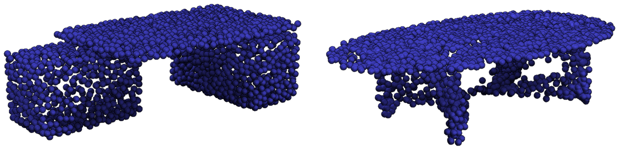

Fig. 4. Left: original sample. Right: counterfactual with intensity of 1.