Robustness Verification for Perception Models against Camera Motion Perturbations

Hanjiang Hu [1] Changliu Liu [1] Ding Zhao [1]

Abstract

Robust perception is still challenging due to the internal vulnerability of DNNs to adversarial examples as well as the external uncertainty of sensing data, e.g. sensor placement and motion perturbation. Recent work can only give provable robustness guarantees in a probabilistic way which is not enough for safety-critical scenarios due to false positive certificates. To this end, we propose the first deterministic provable defense framework against camera motion by extending the verification of neural networks (VNN) method from ℓ_p bounded perturbation to parameterized camera motion space for robotics applications. Through the dense partitions of image projection from 3D dense point cloud to fully cover all the pixels, all the pixel values can be bounded by linear relaxations using linear programming, which makes the camera motion perturbation verifiable and compatible with current incomplete and complete formal VNN methods given DNN models. Extensive experiments are conducted on the Metaroom dataset for the dense image projection and our sound and complete method is more computationally efficient than the randomized smoothing based method at small perturbation radii.

1. Introduction

With the remarkable advancement of deep neural networks (DNNs) in computer vision, visual perception has been well studied and widely used in robotics and autonomous driving applications. Many previous works show that DNNs can be easily fooled to make wrong predictions by adding imperceptible ℓ_p norm bounded noise over pixel level (Goodfellow et al., 2014; Szegedy et al., 2013; Xiao et al., 2018) or semantic transformations like image rotation, translation, scaling, etc. (Pei et al., 2017; Dreossi et al., 2018; Hosseini & Poovendran, 2018; Engstrom et al., 2019; Hendrycks & Dietterich, 2018; Kanbak et al., 2018; Liu et al., 2018), which is also called model internal vulnerability (Hu et al., 2022b). However, in the sense of robust perception for the safety of robotics and autonomous vehicles, there are many other external sensing uncertainty caused by sensor placement or movement (Hu et al., 2022a), motion blurring/corruptions (Sayed & Brostow, 2021; Mintun et al., 2021), adversary environmental conditions (Sun et al., 2022; Hu et al., 2021), etc., which are of great importance in the real-wrold safety-critical scenarios.

To deal with the internal vulnerability of DNNs, many empirical defense methods (Madry et al., 2018; Tramèr et al., 2018; Ma et al., 2018; Tramer et al., 2020) are proposed against ℓ_p-bounded pixel-wise perturbations as well as provable robustness guarantees (Cohen et al., 2019; Tjeng et al., 2018; Zhang et al., 2018; Dathathri et al., 2020). For the robustness certification for semantic transformation, probabilistic certification methods (Fischer et al., 2020; Li et al., 2021; Alfarra et al., 2021; Hao et al., 2022) are recently proposed with good scalability, while deterministic verifications (Balunović et al., 2019; Mohapatra et al., 2020; Ruoss et al., 2021; Yang et al., 2022) are much more important for safety-critical cases with zero tolerance for failure without constructing stochastic smoothed models.

As a commonly-seen external sensing uncertainty for robotics applications, image projection from 3D points to 2D pixels is fundamental in geometric computer vision and computational imaging. As shown in previous work (Hu et al., 2022b), deep learning based visual perception models are not robust against the camera motion perturbations along all six translation and rotation axes, which will influence the safety of the whole robotics system as the downstream tasks of perception. Although randomized smoothing based probabilistic robustness certification is shown in (Hu et al., 2022b), there is still a chance that some non-robust example is wrongly certified to be robust. Hence, it is inadequate for safety-critical applications like autonomous driving. Besides, smoothing-based method (Hu et al., 2022b) is computationally expensive due to Monte Carlo sampling in camera motion space, which is a severe limitation in practical usage.

[1] Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Hanjiang Hu <hanjianghu@cmu.edu>.

Presented at the 2^{nd} Workshop on Formal Verification of Machine Learning, co-located with the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA., 2023. Copyright 2023 by the author(s).

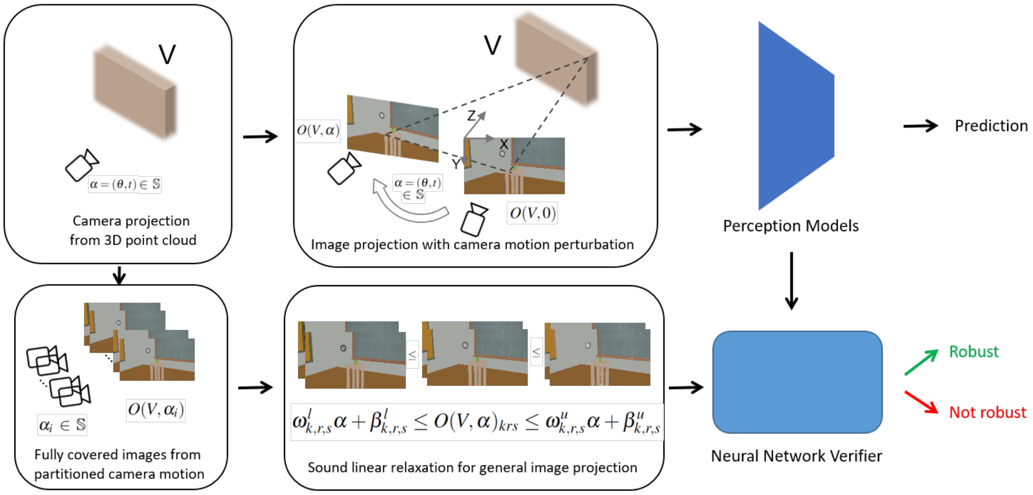
V
$\alpha = (\theta, t) \in \mathbb{S}$
Camera projection from 3D point cloud
V
$O(V, \alpha)$
Z
X
Y
$\alpha = (\theta, t) \in \mathbb{S}$
$O(V, 0)$
Image projection with camera motion perturbation
Perception Models
Prediction
$\alpha_i \in \mathbb{S}$
$O(V, \alpha_i)$
Fully covered images from partitioned camera motion
≤
≤
$\omega^l_{k,r,s}\alpha + \beta^l_{k,r,s} \leq O(V, \alpha)_{krs} \leq \omega^u_{k,r,s}\alpha + \beta^u_{k,r,s}$
Sound linear relaxation for general image projection
Neural Network Verifier
Robust
Not robust

Figure 1. Overview of robustness verification against camera motion perturbation.

To this end, we introduce the formal verification for the robustness of the perception models against the camera motion perturbation. Specifically, by explicitly formulating the image projection from 3D point cloud to 2D pixels, all pixels can be expressed in terms of parameterized camera motion. Then using the technique of uniform partitioning, partitioned images can fully cover all the potential pixels for any camera motion within the perturbation radius. We solve the linear programming using the partitioned images to find the linear lower bound and upper bound for each pixel, making the formal verification possible given the pretrained DNN models and agnostic to off-the-shelf VNN methods (Zhang et al., 2018; Weng et al., 2018; Lyu et al., 2020; 2021; Singh et al., 2019; Xu et al., 2020a). Experiments are conducted on Metaroom dataset (Hu et al., 2022b) and our proposed results are more efficient than the randomized smoothing baseline at small perturbation radii. Our contributions are summarized as follows:

- We propose the first formal verification method for the robustness of vision models against camera motion perturbation.
- We introduce the linear relaxation of image projection through linear programming based on the fully-covered partitions of image capturing.
- Extensive experiments validate that our method is compatible with current VNN methods and achieves sound and complete camera motion perturbation verification efficiently compared to randomized smoothing based certification.

2. Methodology

2.1. Uniform Partitions in Camera Motion Space

Following the literature (Hu et al., 2022b), Then k color channels for each pixel can be defined given colored point cloud $\mathcal{V} : \mathbb{P} \times [0, 1]^k$ via the 3D-2D projective transformation $O : \mathcal{V} \times \mathbb{R}^6 \to \mathbb{R}^m, m = krs$. For the one-axis rotation or translation perturbation in $\mathbb{S}$, $\alpha \in \mathbb{R}^6$ can be degraded to $\alpha \in \mathbb{S} \subset \mathbb{R}^1$ by setting other dimensions to be 0. Detailed definitions are shown in Def. B.1 and B.2 in Appendix.

Given the dense normalized colored point cloud $V : \mathbb{P} \times [0, 1]^k$ for the image projection, the projection function $O(V, \alpha)$ at each pixel $(r, s) \in \mathbb{Z}^2$ is a piecewise constant function w.r.t α, as shown in Figure 2. For all the intervals $\Delta_{r,s}$ along α axis, the projected pixel value $O(V, \alpha)_{r,s}$ is constant for any $\alpha \in \Delta_{r,s}$. Therefore, we present Proposition 2.1 showing given pixel (r, s) that there exists $\Delta_{r,s}$ such that for any camera motion within $\Delta_{r,s}$, all the 3D-2D projections fall into the projections of the endpoints of $\Delta_{r,s}$. In this case, we call the projection function $O(V, \alpha)_{r,s}$ is *fully-covered* by such consistent interval upper bound $\Delta_{r,s}$, as shown in Figure 2. The proof of Proposition 2.1 can be found in Appendix Sec. C.

Proposition 2.1 (Camera motion interval to fully cover each pixel)**.** *Given the projection from dense 3D points $V : \mathbb{P} \times [0, 1]^k$, for each pixel (r, s) there exists an interval $\Delta_{r,s}$ such that $\forall u \in \mathbb{S}, 0 \leq \Delta^* \leq \Delta_{r,s}$, it holds that,*

$$O(V, u + \Delta^*)_{r,s} = O(V, u)_{r,s} \text{ or } O(V, u + \Delta_{r,s})_{r,s}$$

According to the upper bound of consistent camera motion interval in Lemma 2.1, we propose to adopt uniform partitions in the camera motion space $\mathbb{S}$ with partition interval Δ to obtain n partitioned images, $O(V, \alpha_i), \alpha_i \in \mathbb{S}, i = 1, 2, \ldots, n$, where

$$n = \lfloor \frac{\max \mathbb{S} - \min \mathbb{S}}{\Delta} \rfloor + 1, \Delta \leq \min_{(r,s) \in \mathbb{Z}^2} \Delta_{r,s} \quad (1)$$

Therefore, for any pixel over the image grid, all potential values can be captured through these partitioned images, resulting in sound linear constraints introduced in the next

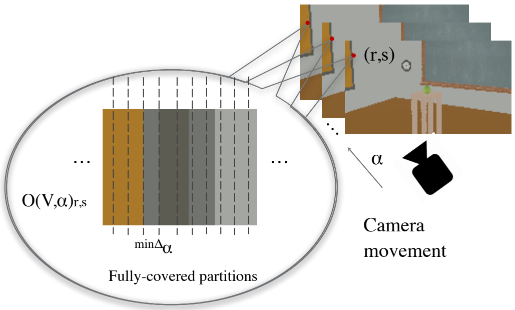

Figure 2. Illustration of uniform partitions in camera motion space to fully cover all the pixel values.

section. In addition to uniform partitions, we leave the adaptive partitioning method in practice as future work.

2.2. Sound Linear Constraints via Optimization

In this section, we aim to find the linear relaxation w.r.t camera motion $\alpha \in \mathbb{S}$ for the lower and upper bounds of the projection function $O(V, \alpha)$, i.e., find the linear weight parameters $\omega^l_{k,r,s}, \beta^l_{k,r,s}, \omega^u_{k,r,s}, \beta^u_{k,r,s}$ for all k 0-1-normalized channels and pixels (r, s) satisfying the following constraints $\forall \alpha \in \mathbb{S}$,

$$\omega^l_{k,r,s}\alpha + \beta^l_{k,r,s} \leq O(V, \alpha)_{krs} \leq \omega^u_{k,r,s}\alpha + \beta^u_{k,r,s} \tag{2}$$

Based on the n uniformly partitioned images from Equation (1), $O(V, \alpha_i), \alpha_i \in \mathbb{S}, i = 1, 2, \ldots, n$, all the potential pixel values for any camera motion $\alpha \in \mathbb{S}$ are captured and sampled. Therefore, the following linear constraint is sound and equivalent to (2), which is different from the unsound constraint in (Balunović et al., 2019).

$$\omega^l_{k,r,s}\alpha_i + \beta^l_{k,r,s} \leq O(V, \alpha_i)_{krs} \leq \omega^u_{k,r,s}\alpha_i + \beta^u_{k,r,s} \tag{3}$$

Furthermore, we formulate the linear relaxation problem in (3) using uniformly partitioned image $O(V, \alpha_i)$ as well as partitioned camera motion α_i as the optimization problems below,

$$\begin{aligned} \min_{\omega^l_{k,r,s}, \beta^l_{k,r,s}} \quad & \frac{1}{n}\sum_{i=1}^{n}[O(V, \alpha_i)_{krs} - \omega^l_{k,r,s}\alpha_i - \beta^l_{k,r,s}] \\ \text{such that} \quad & \omega^l_{k,r,s}\alpha_i + \beta^l_{k,r,s} \leq O(V, \alpha_i)_{krs} \\ & \omega^l_{k,r,s}\alpha_i + \beta^l_{k,r,s} \geq 0, \quad \forall i = 1, 2, \ldots, n \end{aligned} \tag{4}$$

$$\begin{aligned} \min_{\omega^u_{k,r,s}, \beta^u_{k,r,s}} \quad & \frac{1}{n}\sum_{i=1}^{n}[\omega^u_{k,r,s}\alpha_i + \beta^u_{k,r,s} - O(V, \alpha_i)_{krs}] \\ \text{such that} \quad & O(V, \alpha_i)_{krs} \leq \omega^u_{k,r,s}\alpha_i + \beta^u_{k,r,s} \\ & \omega^u_{k,r,s}\alpha_i + \beta^u_{k,r,s} \leq 1, \quad \forall i = 1, 2, \ldots, n \end{aligned} \tag{5}$$

which can be solved directly through linear programming for each channel k and pixel (r, s). The results of linear bounds of image projection can be seen in Figure 7.

2.3. Image Projection in LiRPA

Denote the solved linear weight parameters for all channels and pixels as flatted vectors $\omega^l, \beta^l, \omega^u, \beta^u \in \mathbb{R}^m, m = k \times r \times s$, so $\forall \alpha \in \mathbb{S}$, (2) can be reorganized as below,

$$\text{diag}(\omega^l)\begin{bmatrix}\alpha \\ \vdots \\ \alpha\end{bmatrix} + \beta^l \leq O(V, \alpha) \leq \text{diag}(\omega^u)\begin{bmatrix}\alpha \\ \vdots \\ \alpha\end{bmatrix} + \beta^u \tag{6}$$

where diag$(\cdot)$ is the diagonal matrix constructed from a vector and vector $[\alpha, \ldots, \alpha]^\top$ is with dimension of $m = k \times r \times s$. Note that the form of linear bounds of (6) is the same as Unary Nonlinear Functions (Zhang et al., 2018; Shi et al., 2020; Xu et al., 2020a), so the forward and backward propagation can be obtained based on LiRPA. Details about forward and backward oracle functions in LiRPA can be found in Appendix Sec. D.

3. Experiments

In this section, we will answer two questions: is the sound linear relaxation for the image projection compatible with different off-the-shelf VNN algorithms under different camera motion perturbation radii? How do the formal verification performance and efficiency vary under different model complexities and different translation or rotation axes? To answer these questions, we first introduce the dataset, NN models, metrics and VNN methods as the setup. The code and implementation details are in `https://github.com/HanjiangHu/metaroom_vnn_comp2023`.

3.1. Experimental Setup

Dataset and NN models. We conduct our experiment based on the realistic indoor dataset MetaRoom (Hu et al., 2022b) with camera poses and dense point cloud for image projection. We choose different perturbation radii along z-axis translation and y-axis rotation, as shown in Figure 6. Note that the perturbation radii can be larger if the models to be verified are with smoother decision boundaries. Following the literature of formal verification (Zhang et al., 2018; 2019), we change the camera intrinsic matrix to obtain down-scaled projected images and choose the 4-layer and 6-layer feed-forward convolutional neural networks with ReLU as the perception models to verify the robustness against camera motion perturbation. We remark that our method can be extended to any feed-forward NN (Xu et al., 2020a) through LiRPA with enough computational resources.

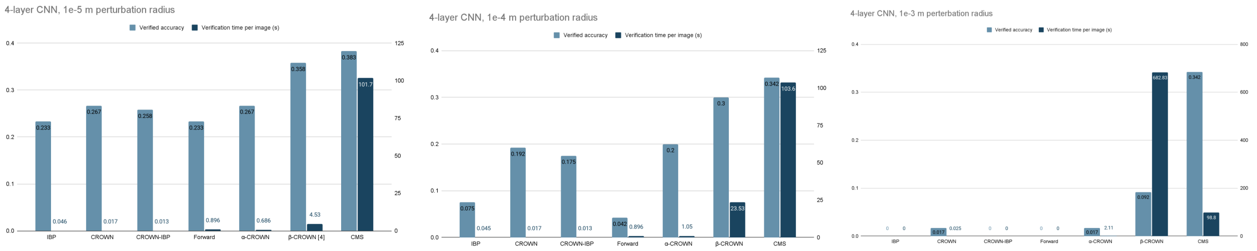

Figure 3. Certified accuracy and certification time per image at different perturbation radii along z-axis translation.

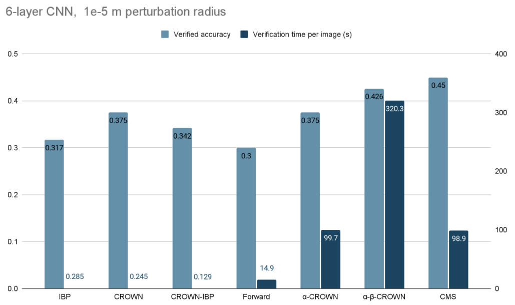

Figure 4. Performance with 6-layer CNN under 1×10^{-5}m z-axis translation perturbation.

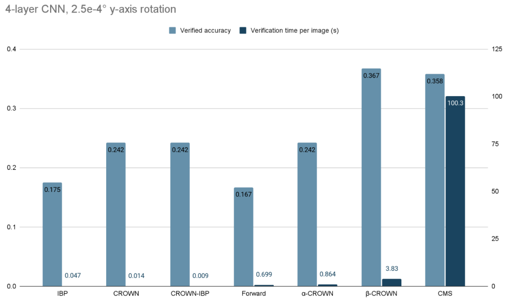

Figure 5. Performance with 4-layer CNN under $2.5° \times 10^{-4}$ y-axis rotation perturbation.

Evaluation metrics and VNN methods. We adopt the certified accuracy, which is the ratio of the test images where the predicted labels are consistent with the ground truth labels and the lower bound of the predicted score for the ground truth label is higher than the upper bounds of scores for all the other non-ground-truth labels. It is also consistent with the robustness certification work (Li et al., 2021; Chu et al., 2022; Hu et al., 2022b) to show the effectiveness of certification approaches. Besides, to compare the time efficiency of certification, we report the average certification time per image for different sound deterministic VNN methods IBP (Gowal et al., 2018), CROWN (Zhang et al., 2018), CROWN-IBP (Zhang et al., 2019), Forward (Xu et al., 2020a), α-CROWN (Xu et al., 2020b), β-CROWN (Wang et al., 2021) and the unsound probabilistic certification method CMS (Hu et al., 2022b) with the confidence of 99%. Note that β-CROWN (Wang et al., 2021) is the complete verification method while others are incomplete.

3.2. Performance associated with different VNN methods at different perturbation radii

From Figure 3, it can be seen that if the perturbation radius is relatively small, all the VNN methods empowered by the sound linear relaxation of image projection work well and the verification time per image is much less than randomized smoothing based one (Hu et al., 2022b) even for complete VNN method (Wang et al., 2021), which answers the first question that our linear relaxation for image projection can help verify the robustness of NN against camera motion perturbation at small radius.

3.3. Influence of different model complexity and different axes of perturbation

We can see from Figure 4 that when the neural network goes deeper, the verification time per image increase dramatically compared to 4-layer CNN, especially for the complete β-CROWN method. But the certified accuracy of all VNN methods shows that our linear relaxation has the potential to scale up to larger models in terms of effectiveness. From Figure 5, we can see that for the rotation along the y-axis, β-CROWN has even better performance than randomized smoothing based CMS (Hu et al., 2022b), showing that the VNN method with the proposed linear relaxed image projection can own the advantage of soundness, effectiveness, and efficiency simultaneously due to tighter linear relaxation along y-axis rotation.

4. Conclusion

In this work, we propose the first deterministic verification method against camera motion for robotics applications. Through the technique of uniform partitioning for image projection, the image projection can be linearly lower-bounded and upper-bounded through the optimization of linear programming, which is agnostic and valid to different LiRPA-based VNN methods. Experiments show that the sound VNN methods empowered by our relaxation can perform better than smoothing-based baseline with much fewer computational resources at small perturbation radii.

References

Alfarra, M., Bibi, A., Khan, N., Torr, P. H., and Ghanem, B. Deformrs: Certifying input deformations with randomized smoothing. *arXiv preprint arXiv:2107.00996*, 2021.

Balunović, M., Baader, M., Singh, G., Gehr, T., and Vechev, M. Certifying geometric robustness of neural networks. *Advances in Neural Information Processing Systems 32*, 2019.

Chu, W., Li, L., and Li, B. Tpc: Transformation-specific smoothing for point cloud models. In *International Conference on Machine Learning*. PMLR, 2022.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R. R., Shankar, S., Steinhardt, J., Goodfellow, I., Liang, P. S., and Kohli, P. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5318–5331, 2020.

Dreossi, T., Jha, S., and Seshia, S. A. Semantic adversarial deep learning. In *International Conference on Computer Aided Verification*, pp. 3–26. Springer, 2018.

Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pp. 269–286. Springer, 2017.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. Exploring the landscape of spatial robustness. In *International conference on machine learning*, pp. 1802–1811. PMLR, 2019.

Fischer, M., Baader, M., and Vechev, M. T. Certified defense to image transformations via randomized smoothing. In *NeurIPS*, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Hao, Z., Ying, C., Dong, Y., Su, H., Song, J., and Zhu, J. Gsmooth: Certified robustness against semantic transformations via generalized randomized smoothing. In *International Conference on Machine Learning*, pp. 8465–8483. PMLR, 2022.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Hosseini, H. and Poovendran, R. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1614–1619, 2018.

Hu, H., Yang, B., Qiao, Z., Zhao, D., and Wang, H. Seasondepth: Cross-season monocular depth prediction dataset and benchmark under multiple environments. *arXiv preprint arXiv:2011.04408*, 2021.

Hu, H., Liu, Z., Chitlangia, S., Agnihotri, A., and Zhao, D. Investigating the impact of multi-lidar placement on object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2550–2559, 2022a.

Hu, H., Liu, Z., Li, L., Zhu, J., and Zhao, D. Robustness certification of visual perception models via camera motion smoothing. In *6th Annual Conference on Robot Learning*, 2022b.

Jeong, J. and Shin, J. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020.

Kanbak, C., Moosavi-Dezfooli, S.-M., and Frossard, P. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4441–4449, 2018.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pp. 97–117. Springer, 2017.

Li, L., Xie, T., and Li, B. Sok: Certified robustness for deep neural networks. *arXiv preprint arXiv:2009.04131*, 2020.

Li, L., Weber, M., Xu, X., Rimanic, L., Kailkhura, B., Xie, T., Zhang, C., and Li, B. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 535–557, 2021.

Liu, C., Arnon, T., Lazarus, C., Barrett, C., and Kochenderfer, M. J. Algorithms for verifying deep neural networks. *arXiv preprint arXiv:1903.06758*, 2019.

Liu, H.-T. D., Tao, M., Li, C.-L., Nowrouzezahrai, D., and Jacobson, A. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *International Conference on Learning Representations*, 2018.

Lorenz, T., Ruoss, A., Balunović, M., Singh, G., and Vechev, M. Robustness certification for point cloud models. *arXiv preprint arXiv:2103.16652*, 2021.

Lyu, Z., Ko, C.-Y., Kong, Z., Wong, N., Lin, D., and Daniel, L. Fastened crown: Tightened neural network robustness certificates. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5037–5044, 2020.

Lyu, Z., Guo, M., Wu, T., Xu, G., Zhang, K., and Lin, D. Towards evaluating and training verifiably robust neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4308–4317, 2021.

Ma, X., Li, B., Wang, Y., Erfani, S. M., Wijewickrema, S., Schoenebeck, G., Song, D., Houle, M. E., and Bailey, J. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Mintun, E., Kirillov, A., and Xie, S. On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34:3571–3583, 2021.

Mohapatra, J., Weng, T.-W., Chen, P.-Y., Liu, S., and Daniel, L. Towards verifying robustness of neural networks against a family of semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–252, 2020.

Müller, M. N., Makarchuk, G., Singh, G., Püschel, M., and Vechev, M. T. Prima: general and precise neural network certification via scalable convex hull approximations. *Proc. ACM Program. Lang.*, 6(POPL):1–33, 2022.

Pang, T., Lin, M., Yang, X., Zhu, J., and Yan, S. Robustness and accuracy could be reconcilable by (Proper) definition. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17258–17277. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/pang22a.html.

Pei, K., Cao, Y., Yang, J., and Jana, S. Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*, 2017.

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018a.

Raghunathan, A., Steinhardt, J., and Liang, P. S. Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, 31, 2018b.

Ruoss, A., Baader, M., Balunović, M., and Vechev, M. Efficient certification of spatial robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 2504–2513, 2021.

Salman, H., Li, J., Razenshteyn, I., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019a.

Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019b.

Samangouei, P., Kabkab, M., and Chellappa, R. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Sayed, M. and Brostow, G. Improved handling of motion blur in online object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1706–1716, 2021.

Shi, Z., Zhang, H., Chang, K.-W., Huang, M., and Hsieh, C.-J. Robustness verification for transformers. *arXiv preprint arXiv:2002.06622*, 2020.

Singh, G., Gehr, T., Püschel, M., and Vechev, M. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41, 2019.

Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., and Yu, F. Shift: A synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21371–21382, 2022.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tjeng, V., Xiao, K. Y., and Tedrake, R. Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*, 2018.

Tramèr, F., Boneh, D., Kurakin, A., Goodfellow, I., Papernot, N., and McDaniel, P. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, 2018.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *Advances in Neural Information Processing Systems*, 33:1633–1645, 2020.

Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.

Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Daniel, L., Boning, D., and Dhillon, I. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Xiao, C., Li, B., Zhu, J. Y., He, W., Liu, M., and Song, D. Generating adversarial examples with adversarial networks. In *27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pp. 3905–3911. International Joint Conferences on Artificial Intelligence, 2018.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020a.

Xu, K., Zhang, H., Wang, S., Wang, Y., Jana, S., Lin, X., and Hsieh, C.-J. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. *arXiv preprint arXiv:2011.13824*, 2020b.

Yang, R., Laurel, J., Misailovic, S., and Singh, G. Provable defense against geometric transformations. *arXiv preprint arXiv:2207.11177*, 2022.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient neural network robustness certification with general activation functions. In *Advances in neural information processing systems*, pp. 4939–4948, 2018.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

Zhang, H., Wang, S., Xu, K., Li, L., Li, B., Jana, S., Hsieh, C.-J., and Kolter, J. Z. General cutting planes for bound-propagation-based neural network verification. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022a.

Zhang, J., Li, L., Zhang, C., and Li, B. Care: Certifiably robust learning with reasoning via variational inference. *arXiv preprint arXiv:2209.05055*, 2022b.

A. Related Work

A.1. Provable Defenses against ℓ_p-bounded Attacks

Empirical defense approaches (Madry et al., 2018; Tramèr et al., 2018; Ma et al., 2018; Samangouei et al., 2018; Tramer et al., 2020; Pang et al., 2022) are well studied to train robust models against specific adversarial perturbations or attacks. In contrast, defense with provable guarantees aims to guarantee the accuracy for all perturbations within some ℓ_p-bounded attack radius (Li et al., 2020; Liu et al., 2019), which is called robustness certification or verification of deep neural networks. The complete verification problem (Katz et al., 2017; Ehlers, 2017) is the NP-complete problem for deep neural networks (Li et al., 2020; Zhang et al., 2022b), although they can guarantee to find such attacks if they exist. Incomplete verification methods are more relaxed for the trade-off of efficiency of verification, which can be categorized into deterministic and probabilistic ones (Tjeng et al., 2018; Wong & Kolter, 2018; Singh et al., 2019; Dathathri et al., 2020; Müller et al., 2022; Zhang et al., 2022a). Although probabilistic certifications (Cohen et al., 2019) based on randomized smoothing present impressive scalability and advantages using adversarial training (Salman et al., 2019a) and consistency regularization (Jeong & Shin, 2020), they are usually not sound, i.e., there exist false positive cases for some non-robust samples. Deterministic certifications are always sound, using linear programming (Salman et al., 2019b; Zhang et al., 2018) or semi-definite programming (Raghunathan et al., 2018a;b) with a sacrifice of scalibility to large-scale datasets.

A.2. Semantic Transformation Robustness Verification

Beyond the ℓ_p bounded perturbation, it is of great interest to study the robustness of deep neural networks against semantic transformations in recent years, e.g. geometric transformation over 2D images or 3D point cloud data. The empirical robustness of adversarial attacks in the semantic transformation (Pei et al., 2017; Dreossi et al., 2018; Hosseini & Poovendran, 2018; Engstrom et al., 2019; Hendrycks & Dietterich, 2018; Kanbak et al., 2018; Liu et al., 2018) is challenging because the landscape of optimization in the parameterized semantic space (e.g. translation, rotation, etc.) is highly non-convex, although these perturbations are closer to the real world than norm-based bounded ones. Recent literature aims to provide the robustness guarantee against 2D images semantic transformations (Hao et al., 2022; Li et al., 2021; Ruoss et al., 2021; Alfarra et al., 2021; Balunović et al., 2019), with either function relaxations-based deterministic guarantees (Balunović et al., 2019; Mohapatra et al., 2020; Lorenz et al., 2021; Ruoss et al., 2021; Yang et al., 2022) or random smoothing based high-confident probabilistic guarantees (Fischer et al., 2020; Li et al., 2021; Alfarra et al., 2021; Chu et al., 2022; Hao et al., 2022). However, the robustness against projective transformation induced by camera movement is rarely studied in the literature, while we believe it is commonly seen in practical autonomous driving and robot applications. Recent work (Hu et al., 2022b) proposes a probabilistic framework to certify such robustness via camera motion smoothing (CMS) in the camera motion space. However, the first weakness of CMS (Hu et al., 2022b) is that it is not sound and the false positive certificate will become a huge concern in safety-critical scenarios. Besides, it is computationally expensive to shake the camera over 10k to 100k times for Monte Carlo sampling with image projections. The above limitations motivate us to provide a sound and efficient formal verification method for DNNs against the camera motion perturbation.

B. Background of Image Projection and LiRPA

In this section, we first detail the image projection from dense 3D point cloud given camera poses (Hu et al., 2022b). Then we list the forward and backward modes in the linear relaxation based perturbation analysis (Xu et al., 2020a) for NN verification.

B.1. Image Projection from 3D Point Cloud

Following the literature (Hu et al., 2022b), image projection can be obtained through the intrinsic matrix K of the camera and the extrinsic matrix of camera pose $\alpha = (R, t) \in \mathbb{R}^6$ given 3D point cloud $\mathbb{P} \in \mathbb{R}^3$. In this way, each 3D point $P \in \mathbb{P}$ corresponds to a 2D position over the pixel grid, which is denoted as the 3D-2D position projection oracle function $\rho : \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^2$, as defined in Definition B.1.

Definition B.1 (3D-2D position projection oracle, Definition 1 from (Hu et al., 2022b))**.** For any 3D point $P = (X, Y, Z) \in \mathbb{R}^3$ under the camera coordinate with the camera intrinsic matrix K, based on the camera motion of $\alpha = (\theta, t) \in \mathbb{R}^6$ with rotation matrix $R = \exp(\theta^\wedge) \in SO(3)$ and translation vector $t \in \mathbb{R}^3$, define the projection function $\rho : \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^2$ and

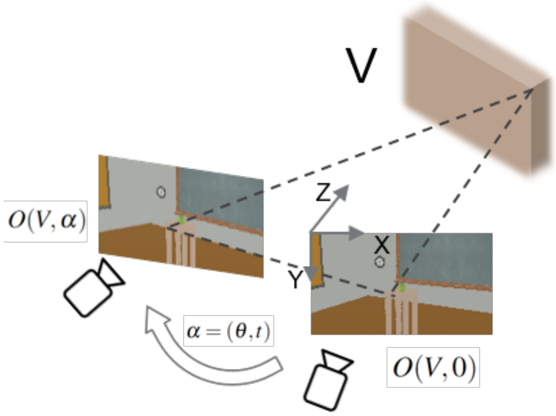

Figure 6. Illustration of image projection from 3D point cloud

the depth function $D:\mathbb{R}^3\times\mathbb{R}^6\to\mathbb{R}$ as an oracle over P

$$[\rho(P,\alpha),1]^\top = \frac{1}{D(P,\alpha)}KR^{-1}(P-t) \tag{7}$$

$$D(P,\alpha) = [0,0,1]R^{-1}(P-t) \tag{8}$$

Based on the position oracle above, the k color channels for each pixel can be defined given colored point cloud $\mathcal{V}:\mathbb{P}\times[0,1]^k$ via the 3D-2D projective transformation $O:\mathcal{V}\times\mathbb{R}^6\to\mathbb{R}^m, m=krs.$ in Definition B.2. The illustration of image projection is shown in Figure 6.

Definition B.2 (Colored projection for each pixel, Definition 2 from (Hu et al., 2022b))**.** Given the oracle of projection function $\rho:\mathbb{R}^3\times\mathbb{R}^6\to\mathbb{R}^2$ and the depth function $D:\mathbb{R}^3\times\mathbb{R}^6\to\mathbb{R}$ with k-channel colored 3D point cloud $V\in\mathcal{V}:\mathbb{P}\times[0,1]^k$ under the camera coordinate frame, define the colored projection for pixels on image gird $\mathbb{Z}^2$ with k channels as $O:\mathcal{V}\times\mathbb{R}^6\to\mathbb{R}^m, m=krs, x=O(V,\alpha)$ parameterized with camera motion $\alpha\in\mathbb{R}^6$ using Floor function $\lfloor\cdot\rfloor$,

$$x_{k,r,s} = O(V,\alpha)_{k,r,s} = V_{P^*_\alpha,k} \tag{9}$$

$$\text{where } P^*_\alpha = \mathop{\arg\min}_{\{P\in\mathbb{P}|\lfloor\rho(P,\alpha)\rfloor=(r,s)\}} D(P,\alpha) \tag{10}$$

Remark B.3. In our work, we focus on the one-axis rotation or translation perturbation in $\mathbb{S}$ where $\alpha\in\mathbb{R}^6$ can be degraded to $\alpha\in\mathbb{S}\subset\mathbb{R}^1$ by setting other dimensions to be 0. We remark that Definition B.1 and B.2 can apply to general rotation and translation in $SE(3)$.

B.2. Linear Relaxation Based Perturbation Analysis (LiRPA)

Following Linear Relaxation Based Perturbation Analysis (LiRPA) (Xu et al., 2020a), on the general computation graph, we study the node i where node j is one of its predecessor nodes $j\in u(i)$. In our case, the input of the graph is the one-axis camera motion $\alpha\in\mathbb{R}$ and the perturbation is within $\mathbb{S}\subset\mathbb{R}$. For node i, the computed value is denoted as $h_i(\alpha)$. In addition to all the nodes of model weights, given pojection function O with 3D point cloud V, the projection input node is node $proj$ with the value of $h_{proj}(\alpha)=O(V,\alpha)$. The output node on the graph is node o with the value of $h_o(\alpha)$.

Goal of LiRPA. The final verification goal against camera motion perturbation $\alpha\in\mathbb{S}$ is to find the linear lower and upper bounds of output node o w.r.t α, i.e.,

$$\underline{\mathbf{W}}_o\alpha+\underline{\mathbf{b}}_o \le h_o(\alpha) \le \overline{\mathbf{W}}_o\alpha+\overline{\mathbf{b}}_o \quad \forall\alpha\in\mathbb{S}, \tag{11}$$

where $\underline{\mathbf{W}}_o, \underline{\mathbf{b}}_o, \overline{\mathbf{W}}_o, \overline{\mathbf{b}}_o$ are the weight parameters to be computed in the following forward and backward modes.

Forward mode of LiRPA. Forward mode propagates bounds from parent nodes to children nodes to achieve the goal (11), e.g., for node i

$$\underline{\mathbf{W}}_i\alpha + \underline{\mathbf{b}}_i \leq h_i(\alpha) \leq \overline{\mathbf{W}}_i\alpha + \overline{\mathbf{b}}_i \quad \forall \alpha \in \mathbb{S}. \tag{12}$$

Initially, for the $k \times r \times s$ dimensional vector of camera motion, it holds that $\mathbf{I}[\alpha, \dots, \alpha]^\top \leq [\alpha, \dots, \alpha]^\top \leq \mathbf{I}[\alpha, \dots, \alpha]^\top$. And for node $i \neq 0$ with parent nodes $u(i)$, given the forward LiRPA oracle function G_i, we have can find the weight parameters $\mathbf{W}, \mathbf{b}$ from parent nodes $u(i)$ to node i,

$$(\underline{\mathbf{W}}_i, \underline{\mathbf{b}}_i, \overline{\mathbf{W}}_i, \overline{\mathbf{b}}_i) = G_i(\{(\underline{\mathbf{W}}_j, \underline{\mathbf{b}}_j, \overline{\mathbf{W}}_j, \overline{\mathbf{b}}_j) | j \in u(i)\}) \tag{13}$$

It is easy to apply (13) recursively to achieve goal (11).

Backward mode of LiRPA. Backward mode propagates bounds from children nodes to parent nodes to achieve the goal (11), i.e., for any node i in $\mathbf{V}$,

$$\sum_{i \in \mathbf{V}} \underline{\mathbf{A}}_i h_i(\alpha) + \underline{\mathbf{d}} \leq h_o(\alpha) \leq \sum_{i \in \mathbf{V}} \overline{\mathbf{A}}_i h_i(\alpha) + \overline{\mathbf{d}} \quad \forall \alpha \in \mathbb{S}, \tag{14}$$

Initially, we have $\underline{\mathbf{A}}_o = \overline{\mathbf{A}}_o = \mathbf{I}$, $\underline{\mathbf{A}}_i = \overline{\mathbf{A}}_i = \mathbf{0}(i \neq o)$, $\underline{\mathbf{d}} = \overline{\mathbf{d}} = \mathbf{0}$ to make (14) hold. And for node $i \neq 0$ with parent nodes $u(i)$, given the backward LiRPA oracle function F_i, we have can find the weight parameters $\mathbf{A}, \mathbf{d}$ from node i to parent nodes $u(i)$,

$$\begin{aligned} &(\underline{\mathbf{\Lambda}}_{u_1(i)}, \overline{\mathbf{\Lambda}}_{u_1(i)}, \cdots, \underline{\mathbf{\Lambda}}_{u_{m(i)}(i)}, \overline{\mathbf{\Lambda}}_{u_{m(i)}(i)}, \underline{\mathbf{\Delta}}, \overline{\mathbf{\Delta}}) = F_i(\underline{\mathbf{A}}_i, \overline{\mathbf{A}}_i), \\ \text{s.t.} \quad &\sum\nolimits_{j \in u(i)} \underline{\mathbf{\Lambda}}_j h_j(\alpha) + \underline{\mathbf{\Delta}} \leq \underline{\mathbf{A}}_i h_i(\alpha) \\ &\overline{\mathbf{A}}_i h_i(\alpha) \leq \sum\nolimits_{j \in u(i)} \overline{\mathbf{\Lambda}}_j h_j(\alpha) + \overline{\mathbf{\Delta}}. \end{aligned} \tag{15}$$

Equation (15) can be solved through a BFS to achieve goal (11) (Xu et al., 2020a). However, the remaining yet challenging part of verifying the robustness against the camera motion is to find the forward and backward oracle functions G_{proj}, F_{proj} in (13) and (15) for the projection input node $proj$, which is the main focus of our work.

C. Proofs in Fully-covered Camera Motion Intervals

Given the dense normalized colored point cloud $V : \mathbb{P} \times [0, 1]^k$ for the image projection, based on the 3D-2D projection in Definition B.1 ad B.2, the projection function $O(V, \alpha)$ at each pixel $(r, s) \in \mathbb{Z}^2$ is a piecewise constant function w.r.t α, as shown in Figure 2. For all the intervals $\Delta_{r,s}$ along α axis, the projected pixel value $O(V, \alpha)_{r,s}$ is constant for any $\alpha \in \Delta_{r,s}$.

We first notice that the projected pixel value is determined by the target 3D point $P^* \in \mathbb{P}$, which has the least projected depth on the pixel (r, s) under any camera motion within the motion interval $\mathbb{U}_{P^*,r,s}$, which is defined as *consistent camera motion interval* in the formal Definition below.

Definition C.1 (Consistent camera motion interval)**.** Given the 3D points $\mathbb{P} \subset \mathbb{R}^3$, the position projection function $\rho : \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}^2$ and the depth function $D : \mathbb{R}^3 \times \mathbb{R}^6 \to \mathbb{R}$, for any $P^* \in \mathbb{P}$ projected on (r, s) with the least depth value, define the consistent camera motion set $\mathbb{U}_{P^*,r,s}$ as the consistent camera motion interval, where

$$\mathbb{U}_{P^*,r,s} = \{\alpha \mid P^* = \operatorname*{arg\,min}_{\{P \in \mathbb{P} | \lfloor \rho(P, \alpha) \rfloor = (r,s)\}} D(P, \alpha)\}$$

Based on the consistent motion interval $\mathbb{U}_{P,r,s}$ for any pixel (r, s) and any 3D point P, intuitively it is easy to find the intervals $\Delta_{r,s}$ along α axis, within which the 3D-2D projection function $O(V, \alpha)$ at each pixel (r, s) has the consistent value. Specifically, all the intervals of the piecewise constant function $O(V, \alpha)_{r,s}$ correspond to different 3D points projected to pixel (r, s) as the camera motion α varies within perturbation $\mathbb{S}$.

Besides, we present the following Lemma C.2 showing given pixel (r, s) that there exists $\Delta_{r,s}$ as the upper bound of the *consistent camera motion interval* $\mathbb{U}_{P,r,s}$ for any $P \in \mathbb{P}$, such that for any camera motion within $\Delta_{r,s}$, all the 3D-2D projections fall into the projections of the endpoints of $\Delta_{r,s}$. In this case, we call the projection function $O(V, \alpha)_{r,s}$ is *fully-covered* by such consistent interval upper bound $\Delta_{r,s}$, as shown in Figure 2.

Lemma C.2 (Upper bound of fully-covered motion interval)**.** *Given the projection from entire 3D point $V \in \mathcal{V} : \mathbb{P} \times [0,1]^K$ along one-axis translation or rotation and the consistent camera motion interval $\mathbb{U}_{P,r,s}$ for any $P \in \mathbb{P}$ projected on (r, s), define the interval $\Delta^{r,s}$ as,*

$$\Delta^{r,s} = \min_{P \in \mathbb{P}} \{\sup \mathbb{U}_{P,r,s} - \inf \mathbb{U}_{P,r,s}\} \tag{16}$$

then for any projection on (r, s) under camera motion $u \in \bigcup_{P \in \mathbb{P}} \mathbb{U}_{P,r,s}$, we have $\forall\ 0 \leq \Delta^* \leq \Delta^{r,s}$

$$O(V, u + \Delta^*)_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s}\} \tag{17}$$

Proof. Considering the projection function ρ on r, s with any 3D point $P : \lfloor \rho(P, \alpha) \rfloor = (r, s)$ at camera motion $\alpha \in \mathbb{S}$ with the least depth, $\mathbb{U}_{P,r,s} \neq \emptyset$. With

$$\Delta^{r,s} = \min_{P \in \mathbb{P}} \{\sup \mathbb{U}_{P,r,s} - \inf \mathbb{U}_{P,r,s}\} = \min_{P \in \mathbb{P} | \lfloor \rho(P,\alpha) \rfloor = (r,s), \alpha \in \mathbb{U}_{P,r,s}} \sup \mathbb{U}_{P,r,s} - \inf \mathbb{U}_{P,r,s} \tag{18}$$

For the 3D projective oracle O on pixel (r, s), based on the definition of $O(V, u), O(V, u + \Delta^{r,s})$, we have $O(V, u)_{r,s} = V_{P_u}, O(V, u + \Delta^{r,s})_{r,s} = V_{P_{u+\Delta^{r,s}}}$, where

$$P_u = \operatorname*{arg\,min}_{\{P \in \mathbb{P} | \lfloor \rho(P,u) \rfloor = (r,s)\}} D(P, u), P_{u+\Delta^{r,s}} = \operatorname*{arg\,min}_{\{P \in \mathbb{P} | \lfloor \rho(P,u+\Delta^{r,s}) \rfloor = (r,s)\}} D(P, u + \Delta^{r,s})$$

Suppose there exists $\Delta^* \in [0, \Delta^{r,s}]$ such that

$$O(V, u + \Delta^*)_{r,s} \neq O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s} \neq O(V, u + \Delta^{r,s})_{r,s}$$

i.e., there exists $O(V, u + \Delta^*)_{r,s} = V_{P_{u+\Delta^*}}$ such that $P_u \neq P_{u+\Delta^*}, P_{u+\Delta^{r,s}} \neq P_{u+\Delta^*}$. In this case, according to the definition of $\mathbb{U}_{P_{u+\Delta^*},r,s}$, it holds that

$$\sup \mathbb{U}_{P_{u+\Delta^*},r,s} - \inf \mathbb{U}_{P_{u+\Delta^*},r,s} < \Delta^{r,s}$$

which contradicts with (18). Therefore, such $O(V, u + \Delta^*)_{r,s}$ does not exist and for any $0 \leq \Delta^* \leq \Delta^{r,s}$,

$$O(V, u + \Delta^*)_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s}\}$$

which concludes the proof. □

Proposition C.3 (Camera motion interval to fully cover each pixel, **restated** of Proposition 2.1)**.** *Given the projection from dense 3D points $V : \mathbb{P} \times [0,1]^k$, for each pixel (r, s) there exists an interval $\Delta_{r,s}$ such that $\forall u \in \mathbb{S}, 0 \leq \Delta^* \leq \Delta_{r,s}$, it holds that,*

$$O(V, u + \Delta^*)_{r,s} = O(V, u)_{r,s} \text{ or } O(V, u + \Delta_{r,s})_{r,s}$$

Proof. Given the projection from dense 3D points $V : \mathbb{P} \times [0,1]^k$, we can find the consistent camera motion interval $\mathbb{U}_{P,r,s}$ according to C.1. Then for the one-axis camera translation or rotation $\mathbb{S} = \bigcup_{P \in \mathbb{P}} \mathbb{U}_{P,r,s}$, by applying Lemma C.2, we can find the upper bound of fully-covered camera motion interval as

$$\Delta^{r,s} = \min_{P \in \mathbb{P}} \{\sup \mathbb{U}_{P,r,s} - \inf \mathbb{U}_{P,r,s}\} \tag{19}$$

where $\forall\ u \in \mathbb{S}, 0 \leq \Delta^* \leq \Delta^{r,s}$

$$O(V, u + \Delta^*)_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s}\} \tag{20}$$

Therefore, simply let any $\Delta_{r,s} \leq \Delta^{r,s}$, we have $\forall u \in \mathbb{S}, 0 \leq \Delta^* \leq \Delta_{r,s} \leq \Delta^{r,s}$, then by Lemma C.2 it holds that,

$$O(V, u + \Delta^*)_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s}\} \tag{21}$$

Besides, with $\Delta_{r,s} \leq \Delta^{r,s}$, we have $\forall u \in \mathbb{S}$, then by Lemma C.2 it holds that,

$$O(V, u + \Delta_{r,s})_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta^{r,s})_{r,s}\} \tag{22}$$

By combining (21) and (22), we have $\forall u \in \mathbb{S}, 0 \leq \Delta^* \leq \Delta_{r,s}$, it holds that

$$O(V, u + \Delta^*)_{r,s} \in \{O(V, u)_{r,s}, O(V, u + \Delta_{r,s})_{r,s}\}, \text{ i.e. } O(V, u + \Delta^*)_{r,s} = O(V, u)_{r,s} \text{ or } O(V, u + \Delta_{r,s})_{r,s}$$

which concludes the proof. □

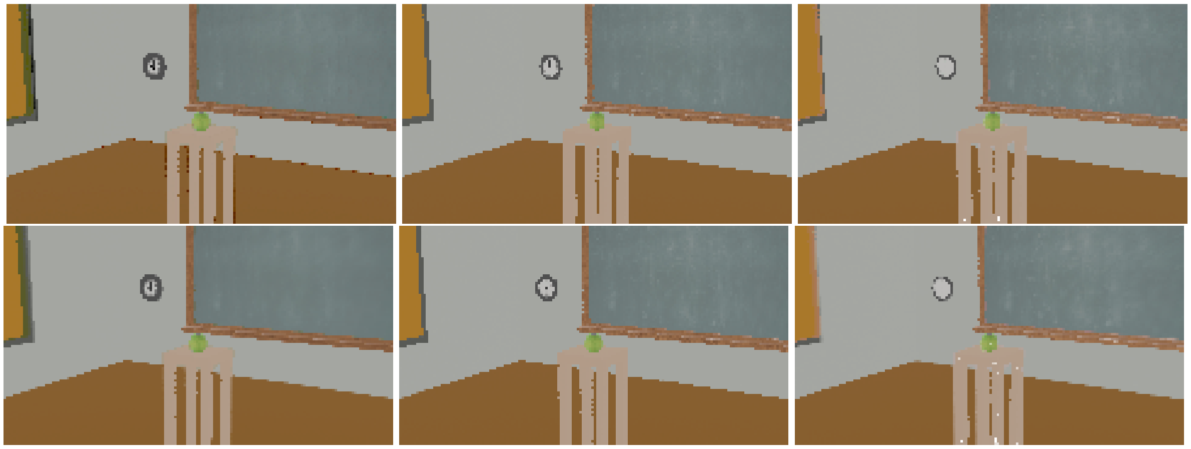

Figure 7. Visualization of sound linear relaxation of the projected images in the middle column. The left column shows the linear lower bound (darker pixels) while the right column shows the linear upper bound (lighter pixels). The camera in the second row moves 0.1m closer to the object along the z-axis compared to the first row.

D. Theorems of Forward and Backward Function for Projection

Theorem D.1 (Forward oracle function for the projection node)**.** *Given the 3D-2D projection function O with 3D point cloud V, construct the projection node* proj *as $\alpha \rightarrow O(V, \alpha)$ followed by neural networks as the general computational graph. Given $\mathbf{I}[\alpha, \ldots, \alpha]^\top \le [\alpha, \ldots, \alpha]^\top \le \mathbf{I}[\alpha, \ldots, \alpha]^\top$, the forward oracle function $G_{proj} : (\mathbf{I}, 0) \rightarrow (\underline{\mathbf{W}}_{proj}, \underline{\mathbf{b}}_{proj}, \overline{\mathbf{W}}_{proj}, \overline{\mathbf{b}}_{proj})$ can be calculated as*

$$\underline{\mathbf{W}}_{proj} = \text{diag}(\omega^l), \underline{\mathbf{b}}_{proj} = \beta^l, \overline{\mathbf{W}}_{proj} = \text{diag}(\omega^u), \overline{\mathbf{b}}_{proj} = \beta^u \tag{23}$$

such that (12) *can be updated with $\forall \alpha \in \mathbb{S}$,*

$$\underline{\mathbf{W}}_{proj} \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} + \underline{\mathbf{b}}_{proj} \le O(V, \alpha) \le \overline{\mathbf{W}}_{proj} \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} + \overline{\mathbf{b}}_{proj}.$$

Theorem D.2 (Backward oracle function for the projection node)**.** *Given the 3D-2D projection function O with 3D point cloud V, construct the projection node $proj$ as $\alpha \rightarrow O(V, \alpha)$ followed by neural networks as the general computational graph. Given the backward relaxation with image node* img *in* (14) *where $\underline{\mathbf{A}}_{img} h_{img}(\alpha) = \underline{\mathbf{A}}_{img} O(V, \alpha), \overline{\mathbf{A}}_{img} h_{img}(\alpha) = \overline{\mathbf{A}}_{img} O(V, \alpha)$, the backward oracle function $F_{proj} : (\underline{\mathbf{A}}_{img}, \overline{\mathbf{A}}_{img}) \rightarrow (\underline{\boldsymbol{\Lambda}}_{proj}, \overline{\boldsymbol{\Lambda}}_{proj}, \underline{\boldsymbol{\Delta}}, \overline{\boldsymbol{\Delta}})$ can be calculated as*

$$\begin{aligned}
\underline{\boldsymbol{\Lambda}}_{proj} &= \max\{\underline{\mathbf{A}}_{img}, 0\}\text{diag}(\omega^l) + \min\{\underline{\mathbf{A}}_{img}, 0\}\text{diag}(\omega^u) \\
\overline{\boldsymbol{\Lambda}}_{proj} &= \max\{\overline{\mathbf{A}}_{img}, 0\}\text{diag}(\omega^u) + \min\{\overline{\mathbf{A}}_{img}, 0\}\text{diag}(\omega^l) \\
\underline{\boldsymbol{\Delta}} &= \max\{\underline{\mathbf{A}}_{img}, 0\}\beta^l + \min\{\underline{\mathbf{A}}_{img}, 0\}\beta^u \\
\overline{\boldsymbol{\Delta}} &= \max\{\overline{\mathbf{A}}_{img}, 0\}\beta^u + \min\{\overline{\mathbf{A}}_{img}, 0\}\beta^l
\end{aligned}$$

such that (14) *can be updated with $\forall \alpha \in \mathbb{S}$,*

$$\underline{\boldsymbol{\Lambda}}_{proj} \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} + \underline{\boldsymbol{\Delta}} \le \underline{\mathbf{A}}_{img} O(V, \alpha), \overline{\mathbf{A}}_{img} O(V, \alpha) \le \overline{\boldsymbol{\Lambda}}_{proj} \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} + \overline{\boldsymbol{\Delta}}.$$

We direct the readers to Section 3.2 of (Shi et al., 2020) and Section A.1 of (Xu et al., 2020a) for the proof of Theorem D.1 and D.2 as a special case of unary nonlinear functions.