1. Common Questions

Coarse-to-Fine Inference: During inference, the results from the backbone network can be shared so only the conic convolution layers need to be forwarded multiple times. Using the Nature Scene dataset as an example, we conduct 4 rounds of coarse-to-fine inference, in each of which we sample 64 vanishing points. So we forward the conic convolution part 256 times for each image during testing. We will clarify such implementation details in the revision.

Improving Efficiency: We note that the speed of academic implementation of LSD/J-Linkage is also 1 FPS, while Contour/J-Linkage takes more than two minutes per image. On the Nature Scene dataset, we have achieved 2.8 FPS with an RTX 2080 Ti by tweaking the block size of im2col and removing one round of inference. The median error increases from 1.10 to 1.11. We may further optimize the algorithm by decreasing the sample number but using more rounds in inference, and adding bottleneck architectures and group convolution to reduce the ops of 3×3 conic convolution.

2. Response to Reviewer #1

Implementation: Our conic convolution operator is implemented by modifying the "im2col + GEMM" function, which is used to implement ordinary convolution in Caffe, MxNet, and Tensorflow, etc. We change the sampling locations of im2col function according to Equation (3). A similar implementation is also used in [1].

Geometry Insight: Conic convolution makes edge detection easier and more accurate. An ordinary convolution may need hundreds of filters to recognize edge with different orientations, while conic convolution requires much less filters to recognize edges aligning with the candidate vanishing point because filters are firstly rotated towards the vanishing point. The strong/weak response (depends on the candidate is positive/negative) will then be aggregated by the subsequent fully-connected layer.

Guided Sampling Inference: We intentionally keep our method simple so that it can be fully trained in an end-to-end fashion. Leveraging intermediate results from traditional methods to the inference process could possibly speed up the computation, but it may also inherit bias and failure modes of such methods. This is nevertheless a valuable direction to study in the future.

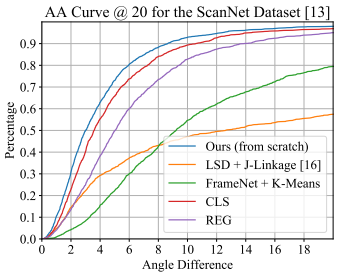

Figure I: More results on ScanNet.

Normal Clustering Baseline: We have conducted the suggested experiment that clusters the network-estimated surface normals into 3 principle directions using the pre-trained model from FrameNet [2], a state-of-the-art normal prediction network on ScanNet. The result is shown in Figure I. The normal clustering method under-performs other neural network baselines because it is not trained in an end-to-end fashion.

3. Response to Reviewer #2

Projecting Line Direction Vector to 2D: Line direction vector $\mathbf{d} \in \mathbb{R}^3$ are defined in the 3D camera space (line 117) and its 2D projection $\mathbf{v}$ to the image is the vanishing point (line 121). We can compute $\mathbf{v}$ by plugging $\mathbf{d}$ into Equation (1) with $\lambda \to \infty$, and then $\mathbf{v} = (p_x, p_y)$.

Metrics from Dataset Papers: We show the curves of consistency measure in Figure II, generated by the code from the authors of the Nature Scene dataset.

Baselines from Dataset Papers: On the Natural Scene dataset, we did compare our methods with the one from the dataset paper (line 265). On the SU3 dataset, our baseline network is the same as the one from the dataset paper except for the number of stacks and head networks (for fair comparison). In addition, LSD/J-Linkage is the de facto standard baseline for vanishing point detection.

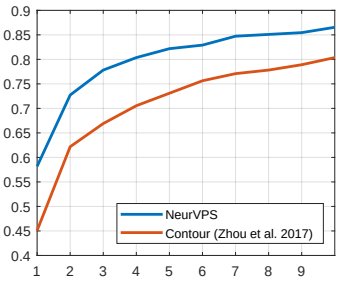

Figure II: Consistency measure on the Nature Scene dataset.

4. Response to Reviewer #3

Clarification: We will write down the loss function in the revision: $L(\hat{y}, y) = -(y \log(\hat{y}) + (1-y)\log(1-\hat{y}))$, where y is the classification label and $\hat{y}$ is the network prediction. Besides, we will clarify that δx and δy in Equation (3) are defined under the summation symbol, representing the offset relative to the convolution center.

Regarding Triplet Loss: Triplet loss is typically used to learn a similarity metric among instances whereas our task is a binary classification problem. It is difficult to pinpoint position of a vanishing point based only on similarity scores.

Regarding Metrics: We did report the mean angle error in the "mean" columns in Tables 1-3 of the paper. However, we find that this metric is unfair to traditional methods because outliers would dominate such errors. For example, in Table 1, the mean error of LSD is much higher than the error of the neural network baseline, but in general the neural network baseline is more inaccurate, according to Figure 5(a).

[1] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[2] Jingwei Huang, Yichao Zhou, Thomas Funkhouser, and Leonidas Guibas. FrameNet: Learning local canonical frames of 3D surfaces from a single RGB image. *arXiv preprint arXiv:1903.12305*, 2019.