# InverseForm: A Loss Function for Structured Boundary-Aware Segmentation

- **Author:** Shubhankar Borse et al.
- **Institution:** Qualcomm AI Research
- **Published:** CVPR 2021 (Oral)
- **Arxiv:** https://arxiv.org/abs/2104.02745

## Contribution

1. Proposed boundary distance-based measure, InverseForm.
    - Significantly more capable of capturing the spatial boundary transform than cross-entropy based beasure.
2. The scheme is agnostic to the backbone architecture choice.
    - Plug-and-play property.
    - Can fit into multi-task learning frameworks.
3. SOTA in both single-task (on NYU-Depth-v2), and multi-task settings (on PASCAL).

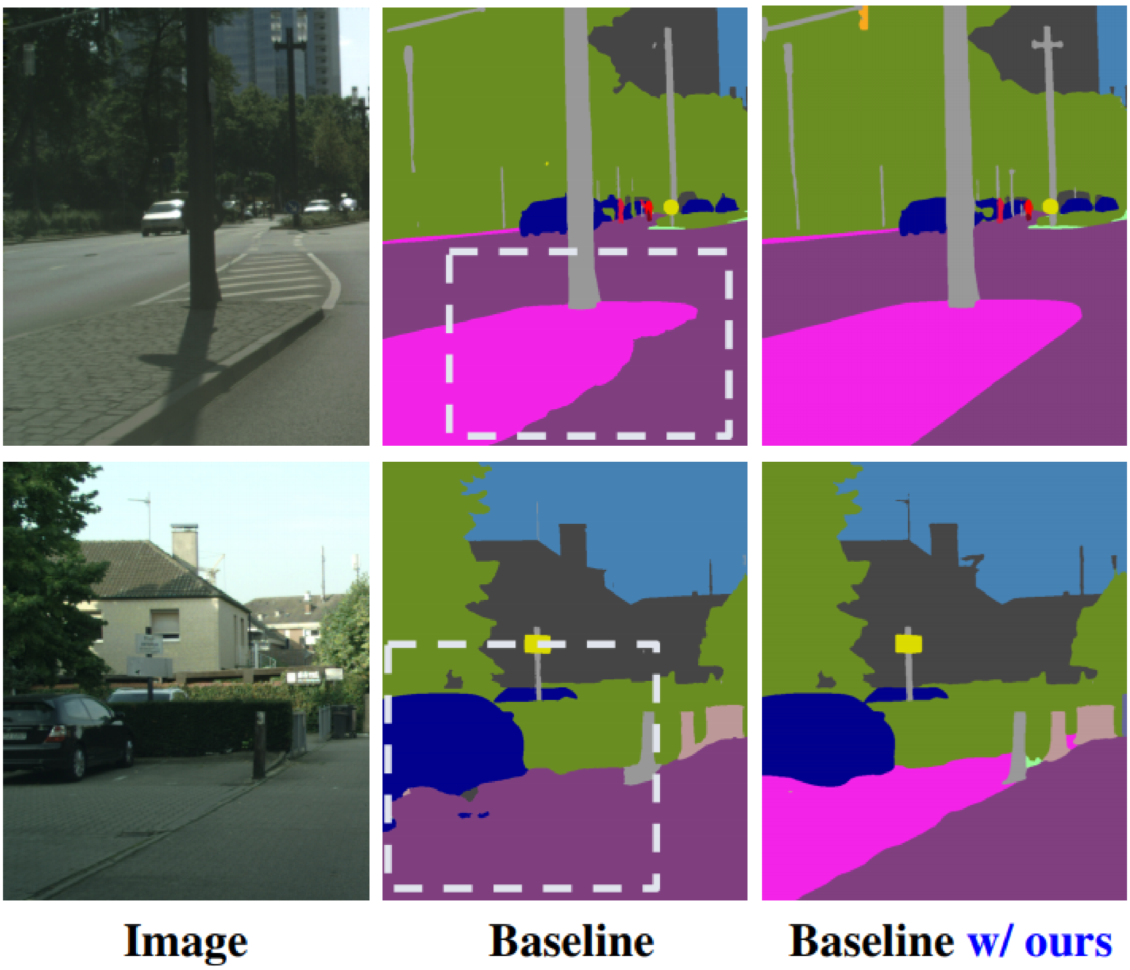



Figure 1. Left: Images from Cityscapes val benchmark. Middle: Segmented prediction for an HRNet-48-OCR baseline. Right: Same backbone trained using our InverseForm boundary loss.

## 3. Proposed Scheme: InverseForm

## 3.1. Motivation for distance-based metrics

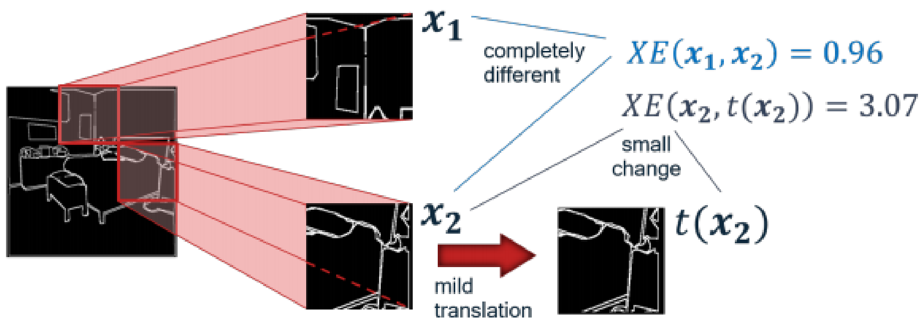
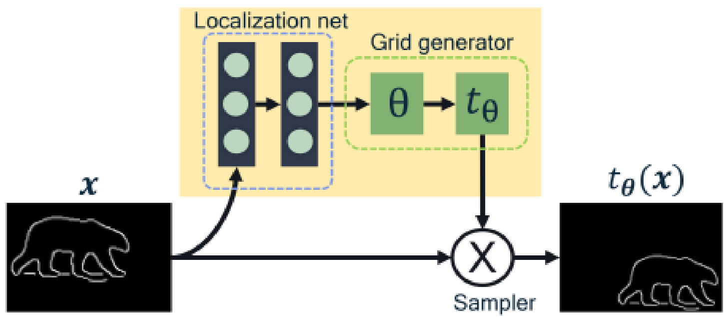


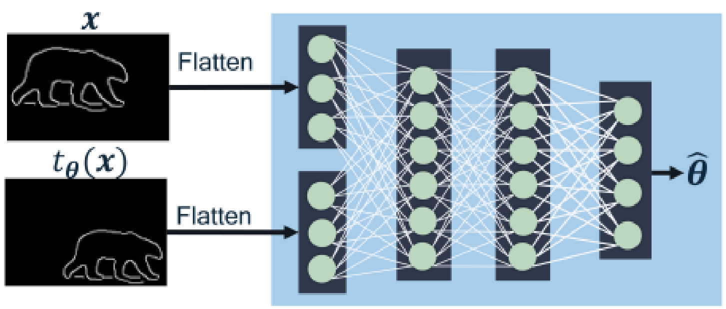

Figure 2. Cross-entropy(XE) based distance fails for spatial transformations of boundaries.

- Pixel-wise cross-entropy or balanced cross-entropy losses take into account the pixel-wise features (intensity, etc.)
- but not spatial distance between object boundaries and ground-truth boundaries.
- They are insufficient for imposing boundary alignment for segmentation. (Illustrated in Figure 2.)
- Accordingly, boundary detection networks trained with pixel-based losses produce thicker and distorted boundaries.
- Some works use Hausdorff distance to model this measure between boundaries,
- but this loss cannot be efficiently applied in a semantic segmentation settings.

## 3.2. Inverse transformation network




**(a)** Spatial transformer network




**(b)** Inverse transformation network

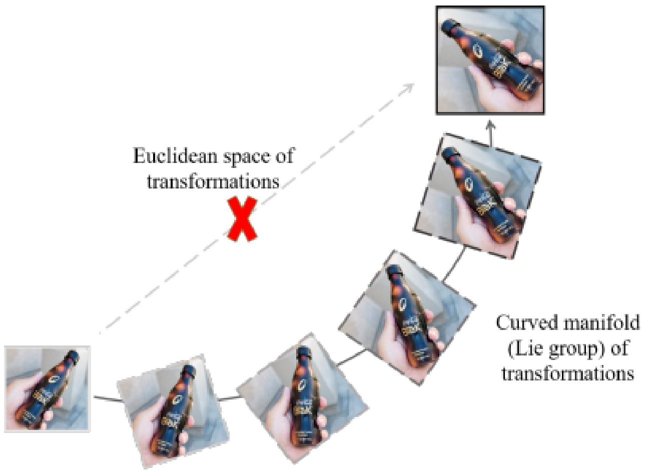
Figure 3. Spatial transformer (a) and our inversetransformation network (b).

- Assume that two boundary maps are related to each other through a homography transformation.
- Need to build Spatial Transformer Network. In this paper, $\theta$ is $3 \times 3$ matrix. (STN [21], Figure 3(a).)
- Create a network that inputs two boundary maps and predicts the "homography change" as its output. (Figure 3(b).)

- This network is called inverse transformation network, because it performs the inverse operation of STN, theoretically.
- The outputs of the inverse transformation network are the coefficients of the homography matrix.
- There are numerous methods to formulate a distance metric from these values. In this paper, two distance metrices are choosed.
- One may also attempt to directly regress on the distance instead of estimating the transformation coefficients.
- However, such an approach would not allow optimization of the boundary-aware segmentation network. **(?)**

## 3.3. Measuring distances from homography

- If there is a perfect match between input boundary maps, the network should estimate an identity matrix.
- **Euclidean distance:**
  - Train the inverse-transformation network by reducing $d_{if}(x, t_\theta(x)) = \left\|\hat{\theta} - \theta\right\|_F$.
  - At inference time, $d_{if}(x, t_\theta(x)) = \left\|\hat{\theta} - I_3\right\|_F$.
- **Geodesic distance:** *(not sure)*
  - Homography transformations reside on an analytical manifold instead of a flat Euclidean space.
  - Figure 1: The deviation between two transformations should be measured along the curved manifold (Lie group) of transformations rather than through the forbidden Euclidean space of transformations.




$$d_{if}(x, t_\theta(x)) = \left\|\frac{\mathrm{Log}(\theta^{-1}\hat{\theta})}{\mathrm{Log}(I_3)}\right\|_F$$

  - Above equation need Riemannian logarithm to calculate gradient, which does not have a closed-form solution.
  - In [27], project the homography Lie group onto a subgroup $SO(3)$ [41] where the calculation of geodesic distance does not need the Riemannian logarithm.
  - Now, the formulation is given by,

$$d_{if}(x, t_\theta(x)) = \arccos\left[\frac{\mathrm{Tr}(P) - 1}{2}\right] + \lambda \mathrm{Tr}(R_\pi^T R_\pi)$$

    - Weighting parameter: $\lambda = 0.1$.
    - Projection $P$ onto the rotation group $SO(3)$: $P = U diag\{1, 1, det(UV)^T\} V^T$
    - Projection residual: $R_\pi = \theta^{-1}\hat{\theta} - P$
  - During inference, $\theta = I_3$

## 3.4. Using InverseForm as a loss function

- At first, train the inverse-transformation network using boundary maps of images sampled from the target dataset.
- Apply the STN [21] to generate the transformed versions of boundary images. It leads greater realistic transformations than randomly sampling transformation parameters.
- Before feeding boundary maps to the network, images are split into smaller tiles.
- Ideally, the best tiling dimension should provide a balance between local and global contexts. (The effect of tiling dimension in the Appendix.)
- Assume the predicted boundary $b_{pred}$ is a transformed version of the ground truth boundary label $b_{gt}$. i.e. $b_{pred} = t_\theta(b_{gt})$.

$$L_{if}(b_{pred}, b_{gt}) = \sum_{j=1}^{N} d_{if}(b_{pred,j}, b_{gt,j})$$

## 3.5. Boundary-aware segmentation setup

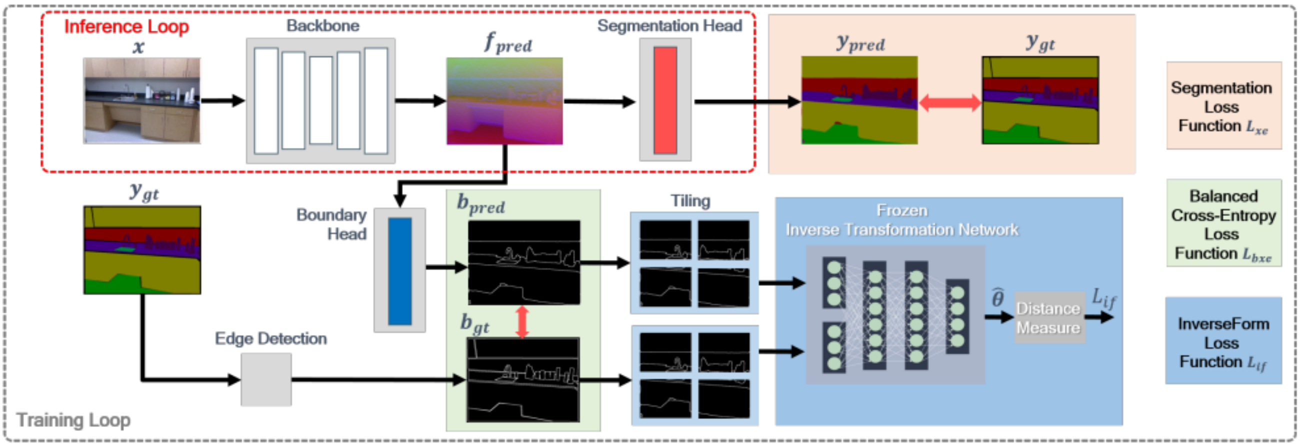



Figure 4. Overall framework for our proposed boundary-aware segmentation.

- Single-task architectures using InverseForm loss.
- Use a simple boundary-aware segmentation setup. (Figure 4.)
- This setup could be used over any backbone.

$$L_{total} = L_{xe}(y_{pred}, y_{gt}) + \beta L_{bxe}(b_{pred}, b_{gt}) + \gamma L_{if}(b_{pred}, b_{gt})$$

# 4. Experimental Results

## 4.1. Results on NYU-Depth-v2

| Network | Tasks | mIoU | mBA |
|---|---|---|---|
| HRNet-w18 | S | 33.18 | 37.46 |
| HRNet-w18 **w/ ours** | S+E | **34.79** | **40.72** |
| PAD-HRNet18 | S+D | 32.80 | 38.10 |
| PAD-HRNet18 | S+D+E+N | 33.10 | 42.69 |
| PAD-HRNet18 **w/ ours** | S+D+E+N | **34.70** | **43.24** |
| MTI-HRNet18 | S+D | 35.12 | 42.44 |
| MTI-HRNet18 | S+D+E+N | 37.49 | 43.26 |
| MTI-HRNet18 **w/ ours** | S+D+E+N | **38.71** | **45.59** |
| HRNet-w48 | S | 45.70 | 56.01 |
| HRNet-w48 **w/ ours** | S+E | **47.42** | **59.34** |

**Table 1:** Comparing baselines for NYU-Depth-v2 using HRNet-w18 and HRNet-w48 backbones in both single task and multi-task learning using our loss. In multi-task settings, S:Semantic segmentation, D:Depth, E:Edge detection, N:Surface normal estimation. Consistent improvement in segmentation mIoU and boundary mBA is visible.


| Network | Tasks | mIoU | mBA |
|---|---|---|---|
| HRNet-w18 | S | 33.18 | 37.46 |
| HRNet-w18 **w/ ours** | S+E | **34.79** | **40.72** |
| PAD-HRNet18 | S+D | 32.80 | 38.10 |
| PAD-HRNet18 | S+D+E+N | 33.10 | 42.69 |
| PAD-HRNet18 **w/ ours** | S+D+E+N | **34.70** | **43.24** |
| MTI-HRNet18 | S+D | 35.12 | 42.44 |
| MTI-HRNet18 | S+D+E+N | 37.49 | 43.26 |
| MTI-HRNet18 **w/ ours** | S+D+E+N | **38.71** | **45.59** |
| HRNet-w48 | S | 45.70 | 56.01 |
| HRNet-w48 **w/ ours** | S+E | **47.42** | **59.34** |

**Table 1:** Comparing baselines for NYU-Depth-v2 using HRNet-w18 and HRNet-w48 backbones in both single task and multi-task learning using our loss. In multi-task settings, S:Semantic segmentation, D:Depth, E:Edge detection, N:Surface normal estimation. Consistent improvement in segmentation mIoU and boundary mBA is visible.

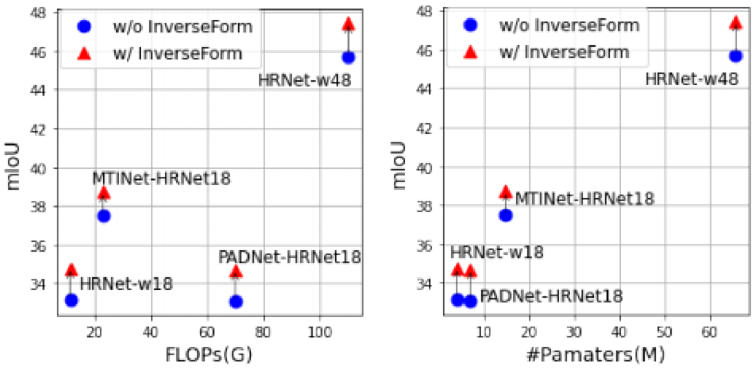




**Figure 5:** Comparison of mIoU v/s FLOPs and mIou v/s # params for different schemes on NYU-Depth-v2 dataset.

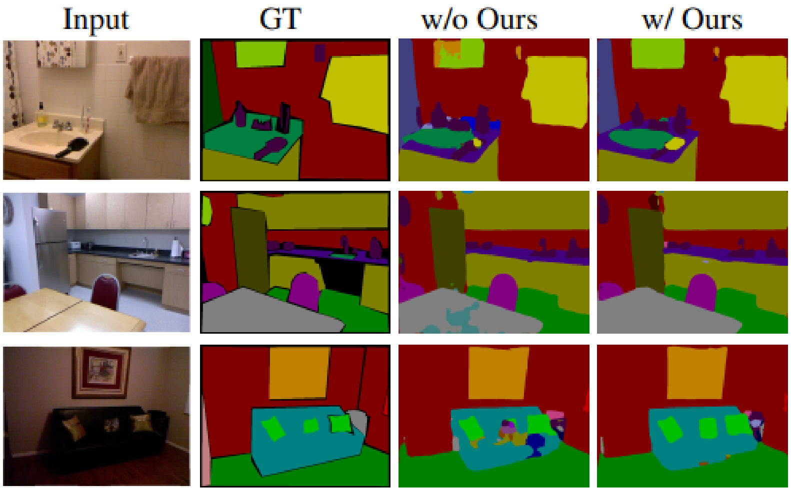



**Figure 7:** NYU-Depth-v2 results showing visual effect of training SA-Gates [35] baseline with InverseForm loss. Clear improvement is visible in the structure of predicted outputs due to boundary-aware segmentation.

## 4.2. Results on PASCAL

| Network | InverseForm | Seg (↑) | Edge (↑) | Saliency (↑) | Parts (↑) | Normals (↓) | $\Delta_m(\%)$ (↑) |
|---|---|---|---|---|---|---|---|
| SE-ResNet-26+ASTMT | | 64.61 | 71.0 | 64.70 | 57.25 | 15.00 | -0.11 |
| | ✓ | **65.13** | **71.4** | **65.29** | **57.93** | 15.07 | **0.49** |
| SE-ResNet-50+ASTMT | | 68.00 | 72.4 | 66.13 | 61.10 | 14.60 | -0.04 |
| | ✓ | **68.83** | **72.5** | **67.50** | **61.13** | **14.55** | **0.95** |
| SE-ResNet-101+ASTMT | | 68.51 | 73.5 | 67.72 | 63.41 | **14.37** | -0.6 |
| | ✓ | **70.14** | **73.7** | **68.70** | **64.76** | 14.55 | **0.39** |
| ResNet-18+MTI-Net | | 65.70 | 73.9 | 66.80 | 61.60 | 14.60 | 3.84 |
| | ✓ | **65.96** | **74.2** | **67.23** | **61.71** | **14.52** | **4.34** |
| HRNet-w18+MTI-Net | | 64.30 | 73.4 | 68.00 | 62.10 | 14.80 | 2.74 |
| | ✓ | **65.12** | **73.6** | **68.61** | **62.53** | **14.67** | **3.72** |

**Table 3:** Training state-of-the-art multi-task learning methods on PASCAL by adding InverseForm loss over boundary detection. HRNet-18 and SE-Resnet backbones are used in a multi-task setting and mIoU scores for segmentation, saliency, human parts and surface normal tasks as well as F-scores for boundary detection are compared with the original results. InverseForm loss consistently improves results barring a few cases.

## 4.3. Results on Cityscapes

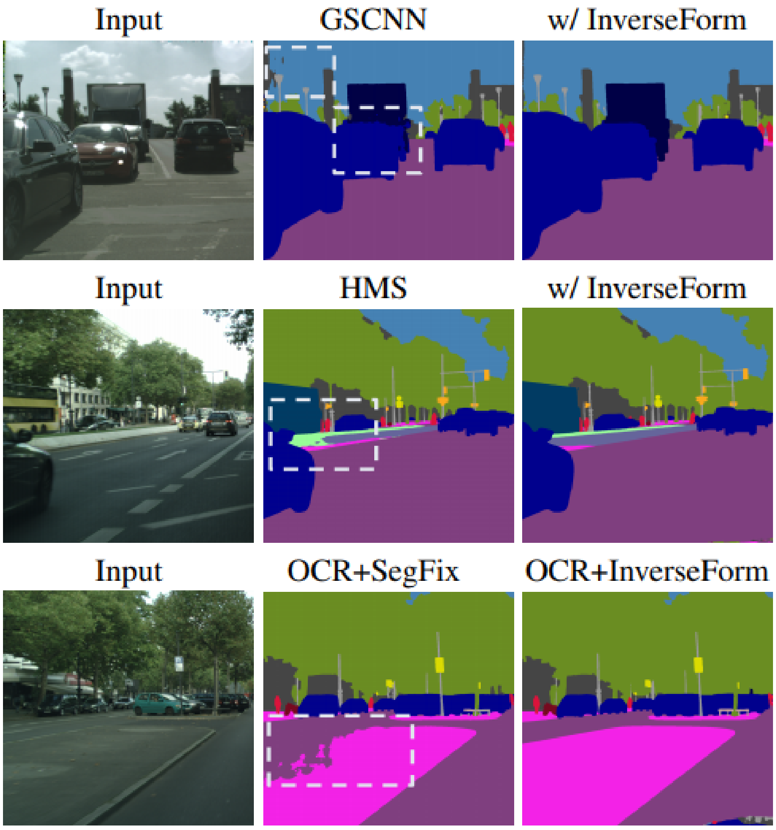



**Figure 6:** Cityscapes results showing visual effect of training different baselines with InverseForm loss. The structure of predicted outputs is improved in highlighted regions due to boundary-aware segmentation.

| Method | Backbone | Split | F | C | mIoU |
|---|---|---|---|---|---|
| Naive-student [4] | WRN41 | Test | ✓ | ✓ | 85.2 |
| GSCNN [39] | WRN38 | Test | ✓ | | 82.8 |
| HRNet-OCR [50] | HRNet48 | Test | ✓ | ✓ | 84.2 |
| OCR+SegFix [50] | HRNet48 | Test | ✓ | ✓ | 84.5 |
| OCR **w/ ours** | HRNet48 | Test | ✓ | ✓ | 84.8 |
| HMS [40] | HRNet48 | Test | ✓ | ✓ | 85.1 |
| HMS **w/ ours** | HRNet48 | Test | ✓ | ✓ | **85.6** |
| GSCNN [39] | WRN38 | Val | ✓ | | 81.0 |
| GSCNN+SegFix | WRN38 | Val | ✓ | | 81.5 |
| GSCNN **w/ ours** | WRN38 | Val | ✓ | | 82.6 |
| GSCNN **w/ ours** | WRN38 | Val | ✓ | ✓ | 84.0 |
| HMS | HRNet48 | Val | ✓ | ✓ | 86.7 |
| HMS **w/ ours** | HRNet48 | Val | ✓ | ✓ | **87.0** |

**Table 4:** Our method compared to various state-of-the-art algorithms on Cityscapes. The models reporting on test split are trained using training+validation data. F:Fine annotations, C:Coarse annotations

## 5. Ablation Studies

- Searching for the best inverse-transformation:

- Compared to the convolutional architecture used in AET [52]

| model | Loss net. | $L_{if,geo}$ | $L_{if,euc}$ | mIoU |
|---|---|---|---|---|
| HRNet-w48 | AET | ✓ | | 47.19 |
| | Ours | ✓ | | 47.28 |
| | AET | | ✓ | 47.03 |
| | Ours | | ✓ | **47.42** |
| HRNet-w18 | AET | ✓ | | 33.82 |
| | Ours | ✓ | | **34.84** |
| | AET | | ✓ | 33.97 |
| | Ours | | ✓ | 34.79 |

**Table 5:** Searching for the optimal InverseForm loss

- **Distance function:**
  - There is no clear winner.
  - Geodesic distance can lead to exploding gradients easily. This severely limits the search-space for hyperparameters.
  - Euclidean distance might not model perspective homography best, but has wider search-space and hence a more consistent improvement.