# SHIC: Shape-Image Correspondences with no Keypoint Supervision

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi

Visual Geometry Group, University of Oxford
{suny, chrisr, vedaldi}@robots.ox.ac.uk
robots.ox.ac.uk/vgg/research/shic/

# Appendix

In this Appendix, we first discuss the limitations of our approach (Sec. 1). Then we discuss implementation details (Sec. 2) and provide additional ablations (Sec. 3). Finally, we show more qualitative examples (Sec. 4) and show a failure mode we observe (Sec. 5).

## 1 Limitations

Our method has several limitations. First, it relies on having a few hundred images per category, which might not always be possible for low-resource classes. However, this is still a significant step forward from prior works, which need much more data and/or human annotations.

Next, the symmetry equivariance loss we propose assumes the shape is symmetric. While this is true for all shapes we consider, there could be several instances where this assumption does not hold: (i) if the shape is not symmetric by design, *e.g.*, it an animal that misses a leg; (ii) if the shape is articulated and thus not symmetric. In that instance, the loss $\mathcal{L}_{eq}$ should not be used, which would lead to a small drop in performance.

Finally, our model only predicts image-to-vertex matching, whereas prior methods such as CSE [1] also predict segmentation masks. However, prior methods do *not* evaluate segmentation performance, as they are not competitive, and this is not the main point of the methods. Furthermore, they use masks as an *additional form of supervision*, as the model is additionally trained to predict masks, whereas we only use masks to sample points used during training (as not to try matching background points to the shape).

## 2 Additional implementation details

### 2.1 Symmetry equivariance loss

We automatically discover the plane of symmetry of the shape. We assume the shape's plane of symmetry is either one of the $(x, y, z)$ planes. In practice, this is most often true. We test each of the $(x, y, z)$ planes as follows. First, we center

**Fig. 1: Background images.** To generate synthetic images, we sample from these, do a random crop, and predict depth.

the mesh. Then, for every plane, we mirror all vertices along that plane. For every vertex, we find its nearest neighbour mirrored vertex. We sum the Euclidean distances between all vertices and their mirrored nearest neighbours. Intuitively, the correct plane of symmetry corresponds to the smallest sum of distances. Finally, for every vertex $x$, we obtain its symmetric one $x_F$ by finding its nearest neighbour when we mirror the shape along the selected plane of symmetry.

### 2.2 Training

During training, we perform data augmentations: random crops, rotations, and colour jitters. We perform these on both the natural and synthetic (generated with a depth-to-image model) images. We train using the Adam optimizer for 40 epochs, using $lr = 0.001$, which is decreased $10\times$ after 20 epochs.

### 2.3 Synthetically generated ground-truth

As discussed in the paper, to generate each synthetic image, we sample a viewpoint and a background.In practice, we sample from 4 background images (Fig. 1), which we randomly crop before computing depth.We find that we can obtain diverse backgrounds with a small number of background templates by using different random seeds. We sample viewpoints only from the side and front.We found that when we sample an image from the back, Stable Diffusion still tries to place a face on the back of the head, leading to unnatural-looking images. We show more examples of generated images in Fig. 2.

## 3 Additional ablations

### 3.1 Pseudo-ground truth

We perform additional ablations on the features used to construct the pseudo-ground-truth $\Sigma$ in Tab. 1. First, we render shaded surfaces instead of surface normals and find that leads to a small drop in performance. Next, we exclude the SD features from SD-DINO [2], and only use DINO features for matching.This

**Fig. 2: Synthetically generated images.**

makes computing the pseudo-ground-truth $\Sigma$ faster, as SD features are more expensive.As expected, we see decreased performance when only using DINO features.

### 3.2 Number of training images

We train our method using a different number of natural images in Tab. 2. We train on $\{50, 200, 500, \text{and } 2k+\}$ images, where $2k+$ is the number of images of the particular class in the dataset, falling between 2k and 3k. We exclude the classes "bear" and "sheep" as they contain under $2k$ images.We see that with as few as 500 images, we achieve comparable performance to our full models.

## 4 Qualitative examples

In Fig. 3 we show similarity heatmaps of the visual feature with the CSE embeddings over the shape. We show further qualitative examples of texture remapping in Figs. 4 to 6.

| Ablation | DensePose-LVIS |
|---|---|
| Ours | **24.9** |
| Renders `shaded` (instead of `normals`) | 25.2 |
| Features w/o SD | 25.4 |

**Table 1: Data ablations.** First, we ablate using shaded renders of the template shape instead of surface normals. Next, we train models using *only* DINO features (w/o SD), as they are quicker to compute. We evaluate using geodesic distance (lower is better).

| No# images | DensePose-LVIS |
|---|---|
| 50 | 34.7 |
| 200 | 29.8 |
| 500 | 24.9 |
| 2k+ | 22.3 |

**Table 2: Ablation of the number of training images.** We ablate the number of training images used for each class. For this ablation, we exclude the "bear" and "sheep" classes as they have under 2k images, the other classes have between 2k and 3k images. We evaluate using geodesic distance (lower is better).

## 5 Failure case

We observe a failure case, where the model predicts wrong patches (Fig. 7). We notice that these patches correspond to the same semantic part, but on opposite sides (*e.g.*, a patch of "left belly" is predicted where there should be "right belly").

**Fig. 3: Similarity heatmaps.** We show similarity heatmaps between the visual feature sampled at the annotated location in red with the CSE embeddings learnt over the shape. We color every vertex according to that similarity and annotate the most similar vertex in red.

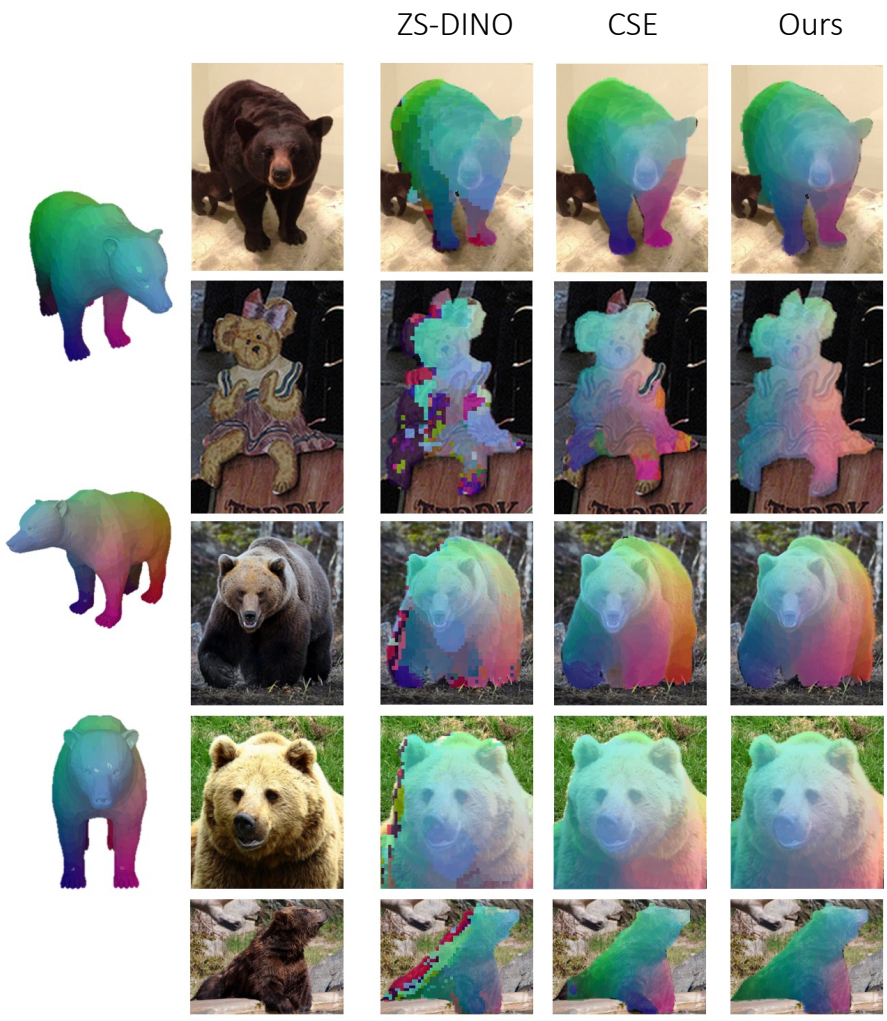


**Fig. 4: Qualitative results.**

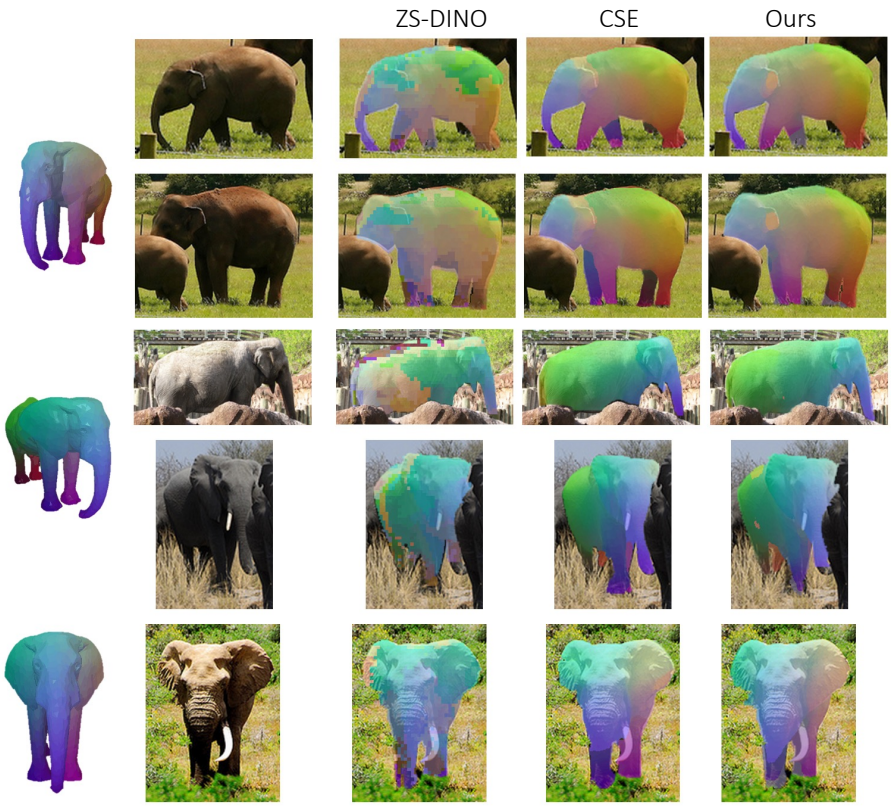


**Fig. 5: Qualitative results.**


**Fig. 6: Qualitative results.**

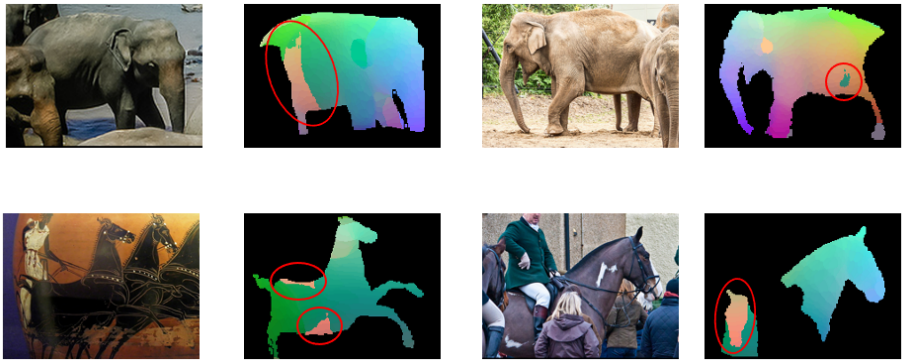

**Fig. 7: Failure cases.** We annotated failure cases in red, where the model predicts wrong patches.

## References

1. Neverova, N., Sanakoyeu, A., Labatut, P., Novotny, D., Vedaldi, A.: Discovering relationships between object categories via universal canonical maps. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 404–413 (2021) 1
2. Zhang, J., Herrmann, C., Hur, J., Cabrera, L.P., Jampani, V., Sun, D., Yang, M.H.: A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence. arXiv preprint arxiv:2305.15347 (2023) 2