A. Acknowledgements

We thank Anurag Beniwal, Benjamin Biggs, Kevin Chen, Mark Davenport, Peter Hallinan, Gerard Medioni, Vaishaal Shankar, Scott Sun, and Guanglei Xiong for their helpful comments, pointers, and feedback.

B. Experimental setup

B.1. Dataset details

In our experiments, we utilize the following datasets. We report the licenses for all datasets that publicly list them.

- CelebAMask-HQ [29]. License: non-commercial research and educational purposes.
- Car-Parts [36].
- DeepFashion-MultiModal [20, 33]. License: non-commercial research purposes.
- SHHQ [14]. License: CC0 and free for research use.
- Cityscapes [12]. License: on-commercial research and educational purposes.

We also utilize pre-trained StyleGAN2 and ReStyle models. In the face and car domain, these models were trained on the following datasets:

- FHHQ [24]. License: Creative Commons BY-NC-SA 4.0 license by NVIDIA Corporation.
- LSUN [44].
- Stanford Cars [27]. License: non-commercial research and educational purposes.

To make the DeepFashion-MultiModal segmentation masks compatible with StyleGAN-Human, we first used the segmentation mask to determine the background for each image and set the background to white. We then re-sized each image to the same size SHHQ images.

B.2. Segmentation mask class collapse

Consistent with prior works [47], we collapse the original labels in each dataset into a smaller number of labeled parts. For CelebAMask-HQ dataset, we remove any distinction between left/right in a number of parts (e.g., ears, eyes, eyebrows). Furthermore, we form one mouth part consisting of upper/lower lips and mouth. Finally, we collapse all accessories and clothing into background. See Tab. 3a for exact class collapse mapping. In long-tail experiments, we un-collapse the relevant long-tail classes (glasses and hats) and consider them separate classes.

For the Car-Parts dataset, we remove any distinction between left/right and front/back for parts such as doors, lights, bumpers, and mirrors. We also merge trunks and tailgates to be the same class. See Tab. 3b for exact class collapse mapping.

For DeepFashion-MultiModal, we consider two degrees of class collapse. In the first, we consider the following ten classes, with original classes included in parentheses: tops (tops and ties), outerwear, dresses (dresses, skirts, rompers), bottoms (pants, leggings, belts), face (face, glasses, earrings), skin (skin, neckwear, rings, wrist accessories, gloves, necklaces), footwear (shoes and socks), bags, and hair (hair and headwear). In the second, we further collapse the classes by including outerwear in tops and bags as background. See Tab. 3c and 3d for exact class collapse mappings.

For Cityscapes, we utilize the eight groups listed on the Cityscapes official website as our classes, with slight modifications. We consider parts labeled sidewalk, parking, and rail track as a part of the void class. See Tab. 3e for exact class collapse mapping.

B.3. Training setup

All experiments were run on V100 GPUs using Amazon Web Services (AWS) P3dn.24xlarge instances. Each MLP in the label generator ensemble was trained with the same parameters for all domains and tasks. Each MLP was trained for $\sim$ 4 epochs via the Adam optimizer [26] with learning rate 0.001 and batch size 64. For all results presented in Tab. 1 and 2, the labeled images used to train the label generator were chosen at random. For long-tail experiments (Sec. 4.5), images with the long-tail part were identified. Then, the labeled training images were selected at random from the identified images.

Prior to training the downstream network, we filter out the top 10% most uncertain synthetically generated images, except for the long-tail experiments. This is to ensure that images with long-tail parts, which are more likely to be "uncertain", are included in the training set for the downstream network. To train the downstream network, we again utilize the Adam optimizer [26] with learning rate 0.001 and batch size 64. We train ReStyle [4] on the set of labeled training images randomly selected from SHHQ [17, 33] and Cityscapes [12] for the full-body human poses and urban driving scene domains, respectively. We use default settings found in the ReStyle repository.

B.4. GAN inversion setup

For the full-body human poses and urban driving scenes domains, we train ReStyle with the candidate training examples. Our framework only uses GAN inversion to obtain latent codes for training the label generator. Training on the candidate training examples thus ensures that ReStyle optimally reconstructs these latent codes. For faces and cars, this procedure is not necessary because ReStyle optimally

Collapsed label (8)	CelebAMask-HQ original labels
Background	Background (0), hat (14), earring (15), necklace (16), neck (17), clothes (18)
Skin	Skin (1)
Nose	Nose (2)
Eyes	Left eye (3), right eye (4), glasses (5)
Eyebrows	Left eyebrow (6), right eyebrow (7)
Ears	Left ear (8), right ear (9)
Mouth	Mouth (10), upper lip (11), lower lip (12)
Hair	Hair (13)

(a)

Collapsed label (10)	Car-Parts original labels
Background	Background(0)
Bumper	Back bumper (1), front bumper (7)
Back window	Back glass (3)
Doors	Back left door (3), back right door (5), front left door (9), front right door (11)
Lights	Back left light (4), back right light (6), front left light (10), front right light (12)
Windshield	Front glass (8)
Hood	Hood (13)
Mirror	Left mirror (14), right mirror (15)
Trunk	Tailgate (16), trunk (17)
Wheel	Wheel (18)

(b)

Collapsed label (10)	DeepFashion-MM original labels
Background	Background(0)
Top	Top (1), tie (23)
Outerwear	Outerwear (2)
Dress	Skirt (3), dress (4), romper (21)
Bottoms	Pants (5), leggings (6), belt (10)
Face	Glasses (8), face (14), earring (22)
Skin	Neckwear (9), skin (15), ring (16), Wrist accessories (17), gloves (19), necklace (20)
Footwear	Footwear (11), socks (18)
Bags	Bags (12)
Hair	Headwear (7), hair (13)

(c)

Collapsed label (8)	DeepFashion-MM original labels
Background	Background(0), bags(12)
Top	Top (1), tie (23), outerwear (2)
Dress	Skirt (3), dress (4), romper (21)
Bottoms	Pants (5), leggings (6), belt (10)
Face	Glasses (8), face (14), earring (22)
Skin	Neckwear (9), skin (15), ring (16), Wrist accessories (17), gloves (19), necklace (20)
Footwear	Footwear (11), socks (18)
Hair	Headwear (7), hair (13)

(d)

Collapsed label (8)	Cityscapes (Fine annotations) original labels
Void	Unlabeled (0), ego vehicle (1), rectification border (2), out of ROI (3), static (4), dynamic (5), ground (6), sidewalk (8), parking (9), rail track (10)
Road	Road (7)
Construction	Building (11), wall (12), fence (13), guard rail (14), bridge (15), tunnel (16)
Object	pole (17), polegroup (18), traffic light (19), traffic sign (20)
Nature	Vegetation (21), terrain (22)
Sky	Sky (23)
Human	Person (24), rider (25)
Vehicle	UCar (26), truck (27), bus (28), caravan (29), trailer (30), train (31), motorcycle (32), bicycle (33), license plate (-1)

(e)

Table 3. Mapping from collapsed class label to original class label in faces (a), cars (b), full-body human poses (c), (d), and urban driving scenes (e) domains. Original class numbers provided for each original class label name in parentheses.

reconstructs the latent codes of training examples without training. For the optimization-based finetuning, we utilize $c_{reg} = 0.5$ and $\lambda_{\ell_2} = 0.1$ for all domains. We run 300 optimization steps for the car domain, 500 iterations for the face and urban driving scenes domains, and 2,000 iterations for the human full-body poses domain. See Appendix C for ablations on GAN inversion optimization steps.

B.5. Label generator architecture

For all experiments, we utilize an ensemble of two layer MLPs with ReLU activations and batch normalizations for our label generator. We sweep the combination of layer widths and report the performance associated with the best performing combination for each domain and number of labeled training images. See Appendix C for ablations on layer widths. Below, we report the combination of label generator sizes that produced the best performance. (x, y) indicates that a network with first hidden layer of width x and second hidden layer of width y was used.

Faces For segmentation, we utilize layer sizes of (256, 32) for 50 training images and (512, 64) for 16 training images. For keypoints, we utilize (512, 32) for PCK-0.1, PCK-0.05, and PCK-0.02 with 50 training images. For 16 training images, we utilize (512, 64) for PCK-0.1 and (512, 32) for PCK-0.05 and PCK-0.02.

Cars For segmentation, we utilize (512, 256) for both 50 training images and 16 training images.

Full-body human poses For segmentation, we utilize (1024, 32) and (2048, 64) for 50 training images in the 8 class and 10 class settings and (2048, 64) and (2048, 128) for 16 training images in the 8 class and 10 class settings. For keypoints, we utilize (512, 128), (256, 128), and (128, 64) for PCK-0.1, PCK-0.05, and PCK-0.02 with 50 training images. For 16 training images, we utilize (512, 256) for all three PCK thresholds.

Urban driving scenes For segmentation, we utilize (512, 64) for both 50 and 16 training images. For depth maps, we utilize (512, 256) for both 50 and 16 training images.

B.6. Keypoint heatmap regression

For keypoint detection experiments, we utilize a heatmap regression setup. Given an image (of size $H \times W$) and a corresponding list of K keypoints, we form a corresponding pixel-wise label for the image as follows. For each of the K keypoints, we create a $H \times W$ sized heatmap. The values of the heatmap are the values of the density of a standard two-dimensional Gaussian centered at the location of the keypoint with variance σ. We further scale the values of the heatmap by 10, so that the maximum value of the heatmap is 10. We find through hyperamater tuning that $\sigma = 25$ works well for full body while $\sigma = 5$ works well for faces. With faces, we use $\sigma = 5$ for the original sized CelebA images and then resize the mask to be of CelebAMask-HQ resolution.

The label generator and downstream task are tasked with predicting a vector of K values for each pixel. At test time, after predicting K heatmaps corresponding to the K keypoints, we take the location of the maximum element of each heatmap as the location of the keypoint. When computing the PCK metric, we only compute if a keypoint was correctly detected for visible keypoints. Information on if a particular keypoint is visible or not is provided in DeepFashion-MM, but not for CelebA.

C. Ablation studies

In this section, we present ablation studies that shed insights on various hyperparameters.

Hypercolumn dimension We experiment with keeping only a subset of the channels from the style block intermediate outputs from the lower resolution layers. In the StyleGAN2 generator, the first 10 style block outputs (which range from 4×4 to 128×128 resolutions) each contain 512 channels, comprising 5120 of the 6080 total channels. We quantify the effect of keeping zero or the first 64, 128, and 256 channels on the downstream task performance in the face domain. As shown in Tab. 6a, in the face domain, while utilizing only higher resolution layers degrades performance considerably, we can remove 256 of the 512 channels for the first 10 style blocks with very minimal loss in performance. This results in a hypercolumn dimension 3520, which is a 42% reduction compared to the original dimension of 6080. In our experiments, we utilize the full hypercolumn dimension, but note that due to memory considerations, utilizing a subset of the dimensions is feasible from a performance trade-off perspective.

Number of MLPs in label generator ensemble We experiment with the number of MLPs in the ensemble. We train 1, 3, 5, 7, and 10 MLPs to generate labels. As seen in Fig. 6b, in the face domain, using only 1 network results in a performance drop, but using anywhere from 3 to 7 MLPs results in performance meeting or even exceeding the performance of using all 10 MLPs. In our experiments, we utilize 10 networks to provide for more robustness in more difficult domains, such as full-body humans and urban driving scenes.

Size of MLPs in label generator ensemble We investigate whether network layer widths impact downstream per-

formance. The original DatasetGAN framework utilizes 3-layer MLPs with intermediate dimensions of 128 and 32. We explore 7 additional combinations of layer widths: (256, 32), (256, 64), (256, 128), (512, 32), (512, 64), (512, 128), and (512, 256). As seen in Fig. 5, in the face domain, for the face domain, downstream performance does not necessarily increase with increasing network widths, but remains relatively stable.

Number of labeled training images We characterize the effects of the number of labeled training images has on downstream task performance in the car domain. As emphasized throughout the paper, a notable benefit HandsOff has over comparable frameworks is the ability for practitioners to increase the number of labeled training images without incurring costs of manual annotations. As observed in Fig. 6c, in the car domain, the downstream performance generally increases as the number of training images is increased, but this increase is not non-decreasing. One explanation for why is that the *composition* of the training data may have a larger impact on downstream performance than simply the number of images. This fact is explored in the long-tail experiments of the main paper. In our experiments, we report the performance with 16 labeled training images, which is the same number of training images in comparable baselines. We also report the performance of 50 labeled training images to highlight our framework's ability to accommodate more than a $3\times$ increase in training data.

Reconstruction quality We examine the effects of GAN inversion reconstruction quality on downstream performance. Specifically, we vary the number of optimization refinement steps on the ReStyle-produced latent code. To quantitatively assess reconstruction quality, we use the value of the loss in the refinement step. As seen in Tab. 6, in the car domain, as the number of optimization iterations increases, the downstream performance generally increases. However, this increase does not scale directly with reconstruction loss.

Size of generated dataset We characterize the effects of the size of the generated dataset on downstream performance. For each generated dataset size, we filter out the top 10% uncertain images. As seen in Fig. 6d, in the car domain, as the size of the dataset grows, the downstream performance generally increases. However, the performance improvement has diminishing returns, as performance improvement is most notable moving from 5,000 to 10,000 generated image-label pairs. As a result, in our experiments, we utilize dataset sizes of 10,000 to strike a balance between performance and time and computation needed to generate larger datasets.

Percent of generated dataset filtered We experiment with the percent of the dataset that is filtered out. To do so, we generate a dataset of size $10,000$ and then filter out varying percentages. As seen in Fig. 6e, in the car domain, employing filtering results in relatively similar performances. Therefore, in our experiments, we utilize a filtering percentage of 10% to strike a balance between removing highly uncertain labels and the number of image-label pairs that are used to train the downstream model.

Cityscapes downstream network finetuning. We report the effects of finetuning the trained downstream model with the original 16 or 50 labeled images used to train the label generator. As seen in Tab. 7, finetuning results in increases in performance, indicating that finetuning overcomes the difficulty in producing high quality in-distribution images with a GAN.

Transfer learning pretrain dataset choice. We report the performance of the transfer learning baseline in the face and car domain when pretrained on ImageNet versus pretrained on ImageNet and COCO. As seen in Tab. 8, pretraining on COCO in addition to ImageNet results in mild performance gains.

D. Additional results

D.1. Reconstructed image alignment

An underlying assumption of the HandsOff framework is that the reconstructed images resulting from GAN inversion align well semantically with the original labels. In this section, we present visual examples of reconstructed image alignment with original labels.

In the face domain, we utilize ReStyle for the encoder initialization and use 500 steps of optimization to refine the images. As seen in Fig. 7a, the reconstructions align very well with the semantic segmentation masks from CelebAMask-HQ.

In the car domain, we utilize ReStyle for the encoder initialization and use 300 steps of optimization to refine the images. As seen in Fig. 7b, the output of the ReStyle captures the overall scene very well, but struggles in preserving fine details, as shown in red circles. By utilizing the optimization based refinement step, we are able to correct for these small details. These refined images align much better with the original segmentation masks, as shown in Fig 7b.

D.2. Face domain few-shot segmentation results

In this section, we compare the downstream few-shot segmentation performance of HandsOff against self-supervised approaches and diffusion-model based approaches. Namely, we compare against DDPM-Segment [6], DatasetDDPM [6], MAE [19], and SwAV [9].

DatasetDDPM and DDPM-Segment both utilize denoising diffusion probabilistic models (DDPMs). DDPM-Segment extracts intermediate network outputs from various time steps of the denoising process to form pixel-level image representations, akin to the hypercolumn representations formed from StyleGAN2 in HandsOff. Then, an ensemble of linear classifiers is trained to output a pixel-level label. DDPM-Segment is different from HandsOff in that it does not generate synthetic datasets. Instead, at inference time, the ensemble of linear classifiers is applied to the pixel-level representation of an image. DatasetDDPM simply replaces the GAN in DatasetGAN with a DDPM, forming pixel-level representations in the same manner as DDPM-Segment. For MAE and SwAV, we utilize the approach of [6] and extract intermediate layer outputs to form image representations of real images. We then train a segmenter to map from these representations to label outputs.

	# labeled images	CelebAMask-HQ 8 classes
DDPM-Segment	16	0.772
DatasetDDPM	20	0.739
MAE	16	0.772
SwAV	16	0.725
HandsOff	16	**0.781**
	# labeled images	CelebAMask-HQ 19 classes
DDPM-Segment	20	**0.599**
MAE	20	0.578
SwAV	20	0.524
HandsOff	20	0.583

Table 4. Segmentation task performance in face domain, reported in mIOU (↑). Top half: experiments performed on our splits with 8 classes. Bottom half: experiments performed on [6] splits with 19 classes. Results for DDPM-Segment, MAE, and SwAV are those as reported in Table 2 in [6].

In Tab. 4, we report the performance on our train/test splits with 8 classes and the train/test splits found in [6] with 19 classes. With our splits and 8 segmentation classes, HandsOff outperforms all baselines, including diffusion model-based approaches DDPM-Segment and DatasetDDPM. This is likely due to two reasons: 1. DDPM-Segment does not leverage the inherent ability of generative models to produce more samples whereas HandsOff produces a large dataset on which the downstream segmenter is trained. The volume of downstream training data compensates for the advantage that diffusion models have over GANs. 2. Unlike DatasetDDPM, HandsOff trains on annotations of real images and avoids hand annotating synthetic images, which as found by [6], when used in training, generally result in poorer performance. With the train/test splits found in [6] and 19 classes, DDPM-Segment performs slightly worse than DDPM-Segment, but outperforms the strongest self-supervised baselines (MAE [19] and SwAV [9]), as reported in [6]. We utilize the implementation of [6] to train DDPM-Segment end-to-end on our train/test splits. Furthermore, we utilize the publicly released synthetically generated datasets from DatasetDDPM to train a downstream network and evaluate on our train/test splits, as the labeled DDPM-generated images used to train DatasetDDPM were not publicly available.

D.3. Additional examples of generated labels

In this section, we present additional visual examples of generated images and their labels as well as examples of segmentation mask improvements in the long-tail segmentation setting.

1. In Fig. 8, we present examples in the face domain. We include examples of the predicted aggregated keypoint heatmaps used to generate the predicted keypoints. To produce the aggregated heatmap, we sum across all of the individual keypoint heatmaps.

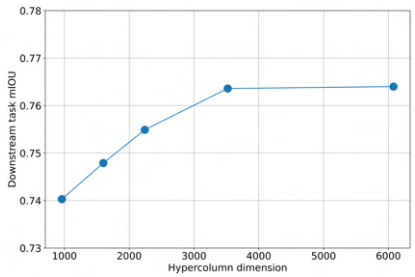
2. In Fig. 9, we present examples in the car domain.

3. In Fig. 10, we present examples in the full-body human pose domain. We again include examples of aggregated predicted heatmaps used to generate the predicted keypoints. To produce the aggregated heatmap, we sum across all of the individual keypoint heatmaps.

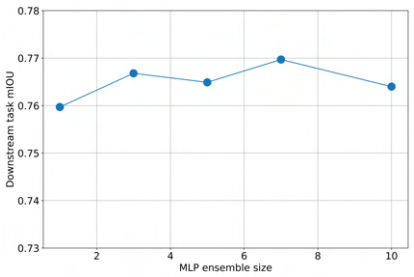
4. In Fig. 11, we present examples in the urban driving scene domain.

D.4. Additional examples of long-tail visualizations

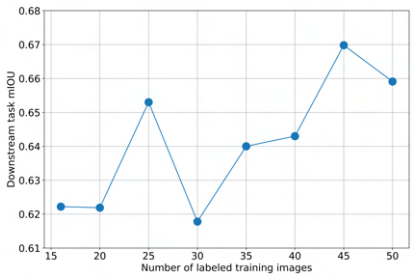
In Fig. 12a and 12b, we present examples of long-tail segmentation mask progressions and pixel-wise uncertainty measurements with glasses and hats, respectively. Uncertainty is measured by Jensen-Shannon divergence (See Sec. 3.3).



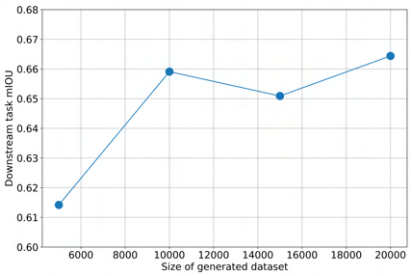
(a) Ablation for hypercolumn dimension in the face domain.



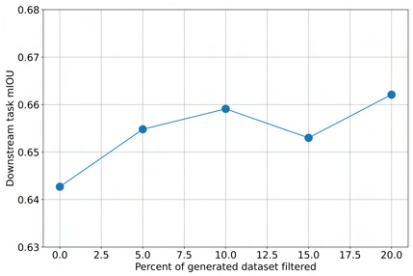
(b) Ablation for ensemble size in the face domain.



(c) Ablation for number of labeled training images in the car domain.



(d) Ablation for the size of generated dataset in the car domain.



(e) Ablation for the percent of generated dataset that is filtered in the car domain.

MLP layer widths	(128, 32)	(256, 32)	(256, 64)	(256, 128)	(512, 32)	(512, 64)	(512, 128)	(512, 256)
mIOU	0.7740	**0.7859**	0.7813	0.7807	0.7828	0.7818	0.7817	0.7850

Table 5. Ablation for MLP hidden layer widths in the face domain

Optimization loss	3.333	2.292	2.185	2.140	2.108	2.089
Optimization iterations	0	100	200	300	400	500
mIOU	0.5735	0.6278	0.6301	**0.6679**	0.6426	0.6591

Table 6. Ablation for GAN inversion quality in the car domain.

# labeled images	No finetuning	Finetuning
16	0.5206	0.5510
50	0.5492	0.6047

Table 7. Ablation for Cityscapes downstream network finetuning.

Domain	# labeled images	ImageNet pretrain	COCO + ImageNet pretrain
Faces	16	0.4575	0.4896
Faces	50	0.6197	0.6295
Cars	16	0.3232	0.3313
Cars	50	0.4802	0.5026

Table 8. Ablation for choice of pretraining dataset for transfer learning baseline.

(a)

(b)

Figure 7. (a) Alignment of reconstructed images with original segmentation masks in the face domain. Semantic features align almost perfectly with segmentation masks. (b) Visualization of fine detail improvement after optimization refinement in car domain. Areas of vast improvement circled in red.

Figure 8. Examples of HandsOff generated labels (segmentation masks, keypoint heatmaps, and keypoints) in the face domain. Last row of examples represent typical failure cases. Hats, a rare class, are occasionally mis-classified as hair or clothing. Additionally, when the image includes GAN generated artifacts, segmentation mask quality is typically lower, while keypoint locations remain accurate.

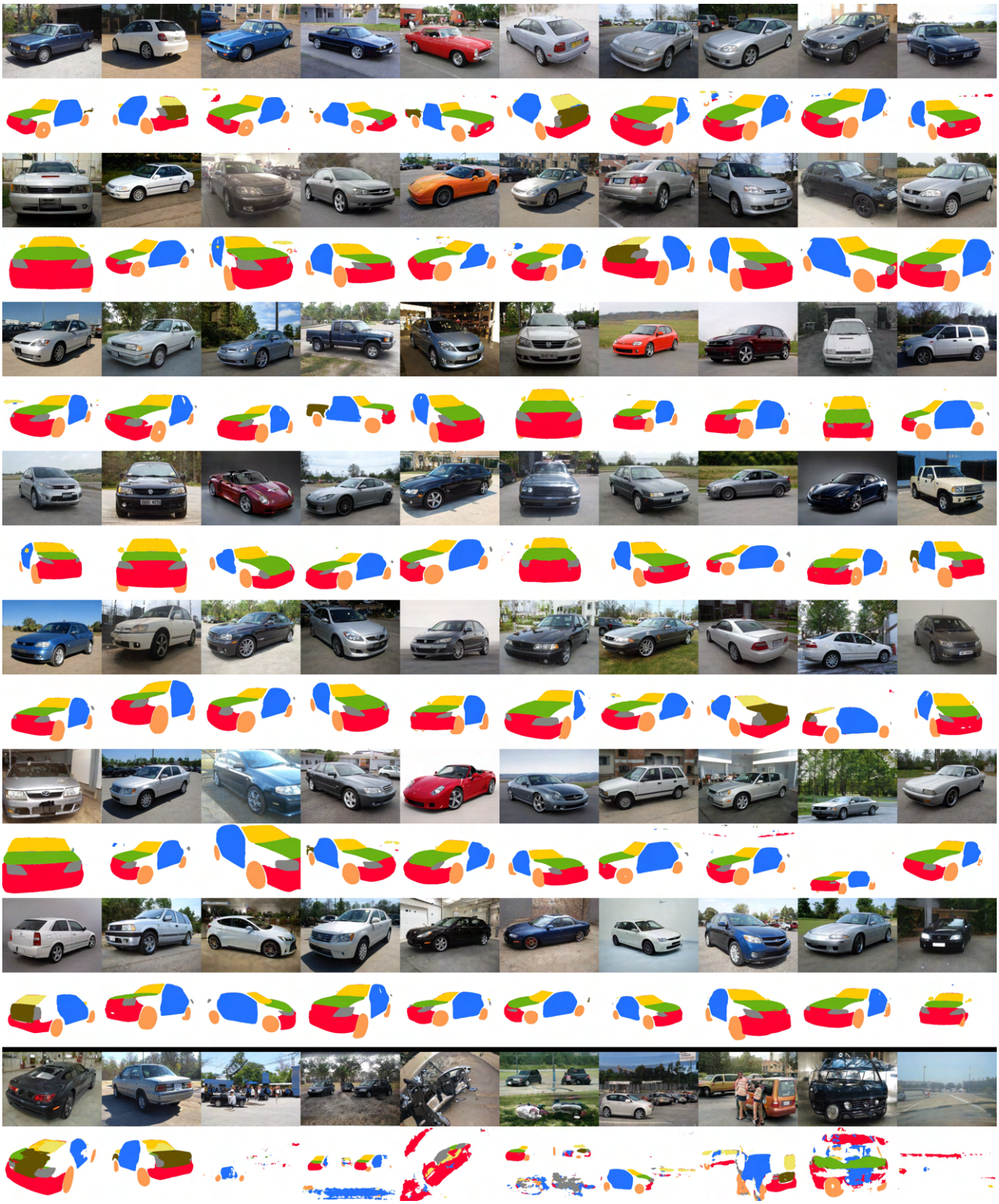


Figure 9. Examples of HandsOff generated segmentation masks in the car domain. Last row of examples represent typical failure cases. Similar classes, such as back trunk and front hood or front glass and back glass are confounded. Additionally, segmentation performance is typically lower when GAN generated images are out of domain or incoherent.

Figure 10. Examples of HandsOff generated labels (segmentation masks, keypoint heatmaps, and keypoints) in the full-body human poses domain. Last row of examples represent typical failure cases. Similar classes, tops, outerwear, and dresses are confounded. Furthermore, patterned pieces of clothing seem to result in mixed segmentation performance. Keypoint locations remain accurate even when segmentation masks are of lower quality.

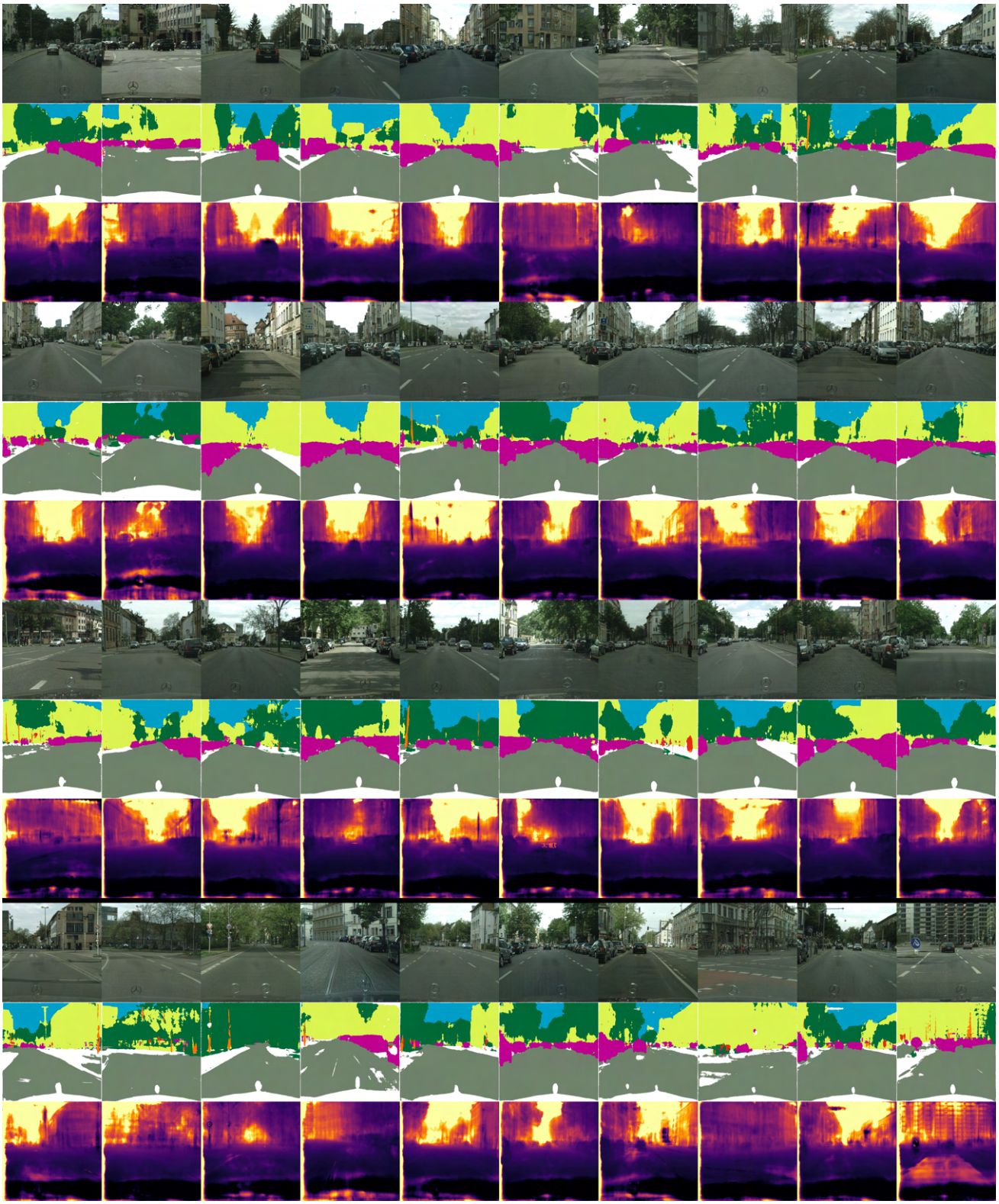
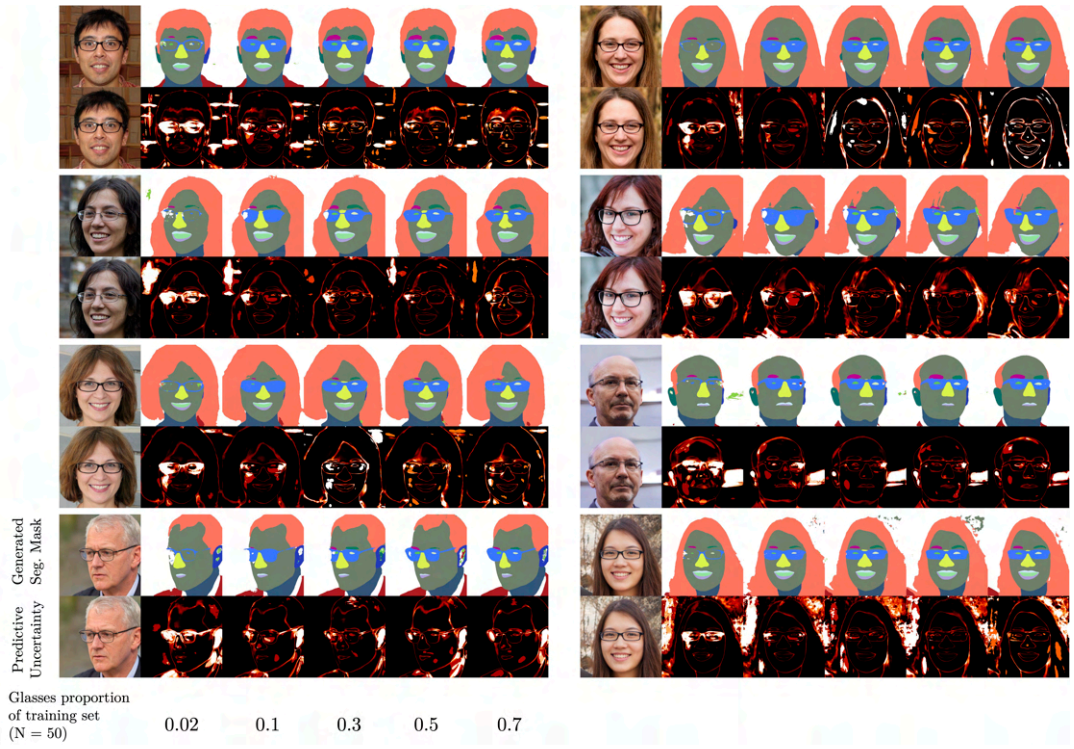
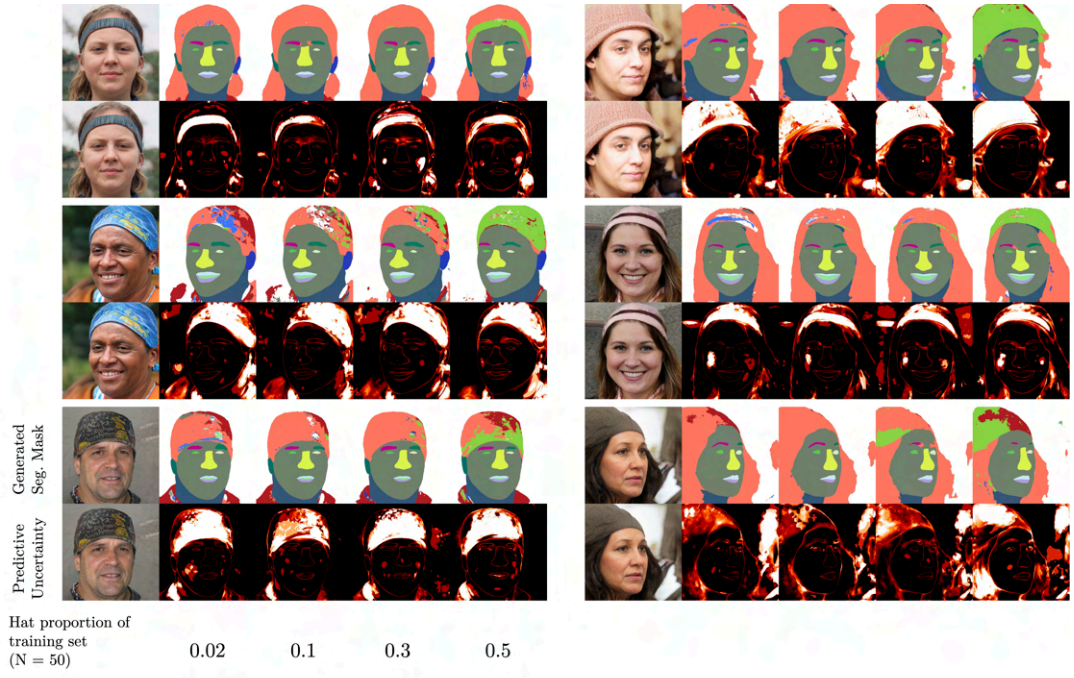


Figure 11. Examples of HandsOff generated labels (segmentation masks and depth maps) in the urban driving scenes domain. Last row of examples represent typical failure cases. Visually small objects such as light poles and street signs are often confounded as background classes or not labeled. In cases of background buildings with many vertical lines, such lines can be mistaken as street sign poles (last image in last row). Depth maps remain relatively accurate even when segmentation masks are of lower quality.



(a)



(b)

Figure 12. Visualization of generated segmentation mask and pixel-wise label generator uncertainty. (a) Not only do we see qualitative improvement in the generated label for glasses, we also see that the classifier is less uncertain when generating the correct label. (b) Hats are a particularly challenging class to characterize, so while the quality of the masks improves drastically, the classifier uncertainty remains relatively high. The last row of examples shows typical failure cases, where the hat is classified as semantically similar classes, such as hair or clothing.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2StyleGAN++: How to Edit the Embedded Images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[3] Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Labels4Free: Unsupervised segmentation using StyleGAN. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[4] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A Residual-Based StyleGAN Encoder via Iterative Refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 5, 11

[5] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. HyperStyle: StyleGAN Inversion with HyperNetworks for Real Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[6] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-Efficient Semantic Segmentation with Diffusion Models. In *International Conference on Learning Representations (ICLR)*, 2022. 14, 15

[7] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The Power of Ensembles for Active Learning in Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 14, 15

[10] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised Monocular Depth and Ego-motion Learning with Structure and Semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019. 7

[11] Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. Editing in Style: Uncovering the Local Semantics of GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 11

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming Transformers for High-Resolution Image Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[14] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen-Change Loy, Wayne Wu, and Ziwei Liu. StyleGAN-Human: A Data-Centric Odyssey of Human Generation. 2022. 5, 11

[15] Raghudeep Gadde, Qianli Feng, and Aleix M Martinez. Detail Me More: Improving GAN's photo-realism of complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 5

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. 2014. 2

[17] Gilles Hacheme and Noureini Sayouti. Neural Fashion Image Captioning: Accounting for Data Diversity. In *arXiv preprint arXiv:2106.12154*, 2021. 11

[18] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANSpace: Discovering Interpretable GAN Controls. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 14, 15

[20] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2Human: Text-Driven Controllable Human Image Generation. *ACM Transactions on Graphics (TOG)*, 2022. 5, 11

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[22] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training Generative Adversarial Networks with Limited Data. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[23] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-Free Generative Adversarial Networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[24] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 11

[25] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 11

[27] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 11

[28] Weicheng Kuo, Christian Häne, Esther Yuh, Pratik Mukherjee, and Jitendra Malik. Cost-Sensitive Active Learning for Intracranial Hemorrhage Detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018. 5

[29] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 11

[30] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, and Antonio Torralba. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[31] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[32] Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. EditGAN: High-Precision Semantic Image Editing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3

[33] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5, 11

[34] Prem Melville, Stewart M Yang, Maytal Saar-Tsechansky, and Raymond Mooney. Active Learning for Probability Estimation Using Jensen-Shannon Divergence. In *Proceedings of the European Conference on Machine Learning*, 2005. 5

[35] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[36] Kitsuchart Pasupa, Phongsathorn Kittiworapanya, Napasin Hongngern, and Kuntpong Woraratpanya. Evaluation of deep learning algorithms for semantic segmentation of car parts. *Complex & Intelligent Systems*, 2021. 5, 11

[37] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[38] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal Tuning for Latent-based Editing of Real Images. *ACM Transactions on Graphics (TOG)*, 2022. 2, 3

[39] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the Latent Space of GANs for Semantic Face Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[40] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an Encoder for StyleGAN Image Manipulation. *ACM Transactions on Graphics (TOG)*, 2021. 2

[41] Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, and Qifeng Chen. High-Fidelity GAN Inversion for Image Attribute Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[42] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 7

[43] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN Inversion: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[44] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv preprint arXiv:1506.03365*, 2015. 11

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[46] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep Long-Tailed Learning: A Survey. *arXiv preprint arXiv:2110.04596*, 2021. 1

[47] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. DatasetGAN: Efficient Labeled Data Factory with Minimal Human Effort. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 3, 5, 11

[48] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative Visual Manipulation on the Natural Image Manifold. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3