MIMIC: Masked Image Modeling with Image Correspondences

Kalyani Marathe[1,2*] **Mahtab Bigverdi**[1,2*] **Nishat Khan**[1] **Tuhin Kundu**
Patrick Howe **Sharan Ranjit S** [1] **Anand Bhattad** [3] **Aniruddha Kembhavi**[2]
Linda G. Shapiro[1] **Ranjay Krishna**[1,2]
[1]University of Washington, [2]Allen Institute for Artificial Intelligence,
[3]Toyota Technological Institute at Chicago
{kmarathe,mahtab,nkhan51,shapiro,ranjay}@cs.washington.edu,
anik@allenai.org, tuhinkundu@outlook.com, {pdh, sharanrs}@uw.edu

Appendix

1. Dataset, Resources, Assets

1.1. Dataset usage

The code and instructions to download, access, and use MIMIC-3M can be found here. The primary use case of this dataset is to train a 3D-aware ViT in a self-supervised manner.

1.2. Compute Resources

As mentioned in Section 4.1 (Pretraining) we train CroCo [20] for 200 epochs, each epoch taking about 1 hour 40 minutes using 8 NVIDIA RTX A6000 GPUs. The cost for one training run is about 111 GPU days.

1.3. Assets

We provide the details of the dataset and code licenses used in our study in Table1. We bear all responsibility in case of violation of rights. Our code is primarily based on MAE [9], Multi-MAE [2] and CroCo [20] and our work is licensed under CC BY-NC-SA 4.0.

Table 1. List of the assets and licenses

Asset	License
Pretraining datasets	
HM3D [15]	[link]
Gibson [21]	[link]
3DStreetView [23]	[link]
CO3D [16]	[link]
Mannequin [11]	[link]
ArkitScenes [3]	[link]
Objectron [1]	[link]
ScanNet [5]	[link]
Matterport [4]	[link]
DeMoN [19]	[link]
Downstream datasets	
ImageNet-1K [6]	[link]
NYUv2 [14]	[link]
ADE20K [25]	[link]
Taskonomy [24]	[link]
MSCOCO [12]	[link]
Code/Pretrained models	
MAE [10]	[link]
CroCo [20]	[link]
MultiMAE [2]	[link]

2. Data curation details

2.1. Details on mining potential pairs

We utilized different data types within our datasets, including videos, 3D scenes, and street views. Consequently, the process of mining potential pairs for each data type varied. For street views [23], we adopted a strategy where we grouped images based on their target id (images that have the same target id in their name, show the same physical point in their center). Subsequently, among all possible combinations of images in a group, we selected the pair with minimal overlap ranging from 50% to 70%.

When dealing with video data, a practical approach involved creating a list of frames at reg-

[*] The authors contribute equally to this work.

ular time intervals, determined by the speed of the video. Then, we generated pairs of consecutive frames from this list. In cases where substantial overlap between consecutive frames was observed, we specifically chose the second consecutive frame and evaluated its overlap with the preceding frame. We implemented this step to ensure that the selected frame pair exhibits an appropriate level of dissimilarity and minimized redundancy.

To tackle the challenges associated with handling 3D scenes, we employed the habitat simulator [17] to sample locations within the navigable area of the scene. We initialized an agent with a random sensor height and rotated it eight times at $45°$ intervals, capturing a comprehensive view of the surroundings to form the first list of eight images. Subsequently, we sampled a random rotation degree from multiples of $60°$ (excluding $180°$ and $360°$), and rotated the agent accordingly before moving in the current direction for a random step ranging from 0.5 to 1 meter. We repeated the process of rotating eight times at $45°$ intervals, capturing the second list of eight images. Likewise, we randomly rotated and moved the agent to generate the third list of eight images. From these lists, we selected an optimal pair (img_1, img_2) from a pool of 8×16 potential pairs. img_1 belonged to the first list, while img_2 was chosen from the combined pool of the second and third lists, with a minimal overlap ranging from 50% to 70%, if applicable.

The selection of a $45°$ rotation aimed to capture a comprehensive view of the environment while minimizing redundancy. Furthermore, the choice of rotation degrees as multiples of $60°$ prevented capturing images in directions already covered by those obtained with the $45°$ rotation, effectively avoiding the capture of zoomed-in versions of previously acquired images.

2.2. Details on measuring the overlap

Given a pair of images or views from a scene (we call it a potential pair), we checked whether these two are sufficiently overlapped during the six steps. If they had enough overlap, we saved this pair along with other metadata for the next phase, which was the model pretraining. The six steps are listed below:

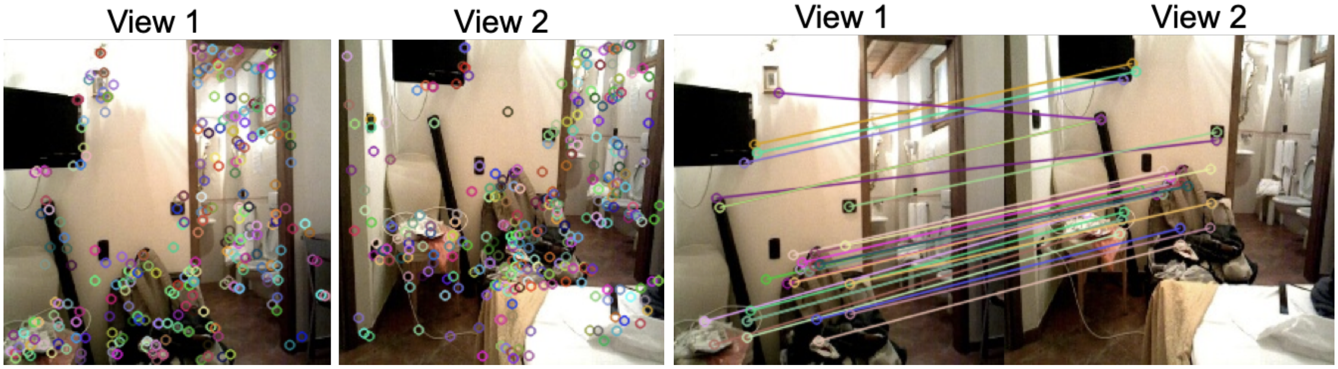
Keypoint localization using SIFT [13]. We used SIFT (Scale-Invariant Feature Transform) as a feature detector to localize the two views' key points separately. SIFT has been shown to perform well compared to other traditional methods. Figure 1a provides an example pair with key points.

Brute force matching. Having obtained both key point features and their descriptors from the previous step, we performed a brute-force matching process to match the key points in the first view (source points) with the key points in the second view (destination points). We present matches between two views in Figure 1b.

Finding homography transformation [8]. We leveraged the homography [8] matrix to translate the transformation among the views with provided source and destination points matches from the previous step. However, we know the found transformation is not thoroughly accurate and free of errors. Therefore, to overcome this issue, we used RANSAC [7] to conclude with better estimations of the transformation. As a result, only some of the matches was categorized as inliers. Inlier matches are shown in Figure 2a

Creating non-overlapping patches. After finding the homography matrix, we divided each view into non-overlapping patches (16×16 here) and matched patches from view 1 to view 2, see Figure 2b.

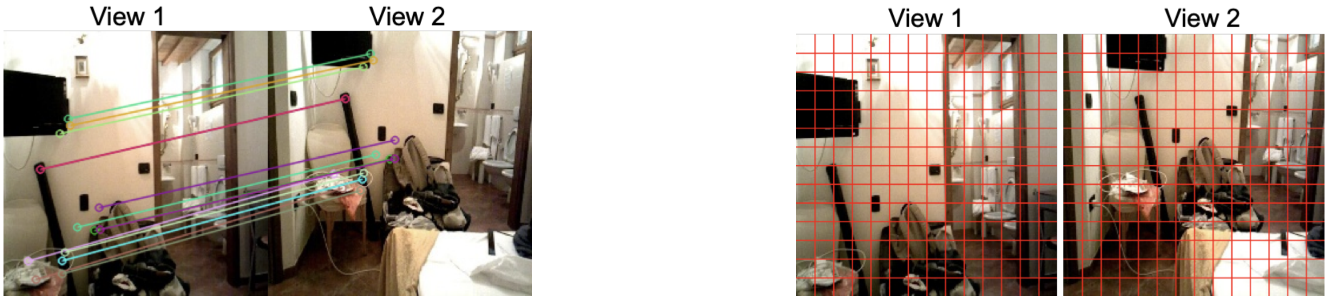
Obtaining the patch correpondences To find a corresponding patch in the second view for a particular patch in the first view, we performed the following steps: 1. Randomly sampled a suitable number of points within the specific patch in the first view (e.g., 100 points). In Figure 3a, random green points are sampled within the green patch of the first view. 2. Applied the homography matrix H to the sampled points to determine their corresponding positions in the second view. 3. Determined the patch number in which each corresponding point falls, such as $patch(x = 17, y = 0) = 1$. 4. Identified the patch that contains the



(a)

(b)

Figure 1. **(a)** A pair of images with SIFT key points. **(b)** Matching key points of images with a brute force matcher.



(a)

(b)

Figure 2. **(a)** Inlier matches after finding the homography matrix. **(b)** Dividing each image to non-overlapping patches.

maximum number of corresponding points as the match for the specific patch in the first image. In Figure 3b, the blue points represent the positions of the corresponding points in the second view that fall within nearby patches. It can be observed that the majority of the blue points cluster within a specific patch, which is marked as the matched patch for the green patch. This match is illustrated in Figure 4a.

Measuring the visual overlap We repeated the procedure from the previous step for all patches in the first view to determine their matches in the second view. We computed the count of patches in the first view that have a matching patch within the boundaries of the second view, provided that the matching patch has not been previously matched with another patch from the first view. Then, we divided this count by the total number of patches, serving as a metric to measure the overlap.

To ensure a comprehensive evaluation, we performed the mentioned algorithm both for finding $overlap(view1, view2)$ and its inverse, $overlap(view2, view1)$. We chose the minimum value between these two overlap metrics as the final overlap measure.

Subsequently, we retained pairs with an overlap ranging from 50% to 75% along with corresponding patches information. Figure 4b showcases all patches from the first view that have their matches falling within the second view. Additionally, Figure 5 provides an illustrative example of a retained pair of images from each dataset, along with their corresponding patches.

3. Downstream tasks

3.1. Finetuning details

For fine-tuning depth estimation, semantic segmentation, and surface normal estimation we adopt the task-specific decoders from Multi-

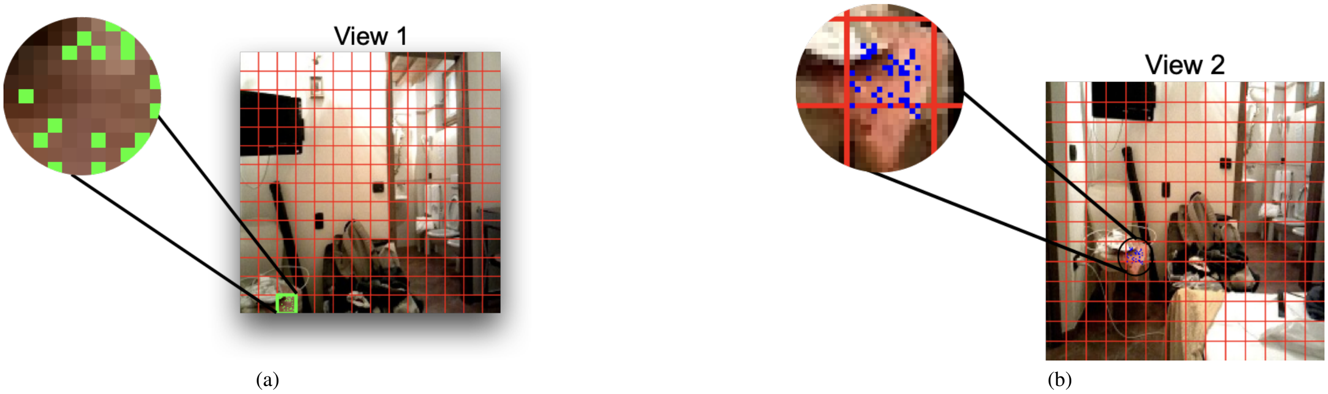


Figure 3. **(a)** Sampling random points from a patch in the first view. **(b)** Blue points are the corresponding points of the green points in the second view.

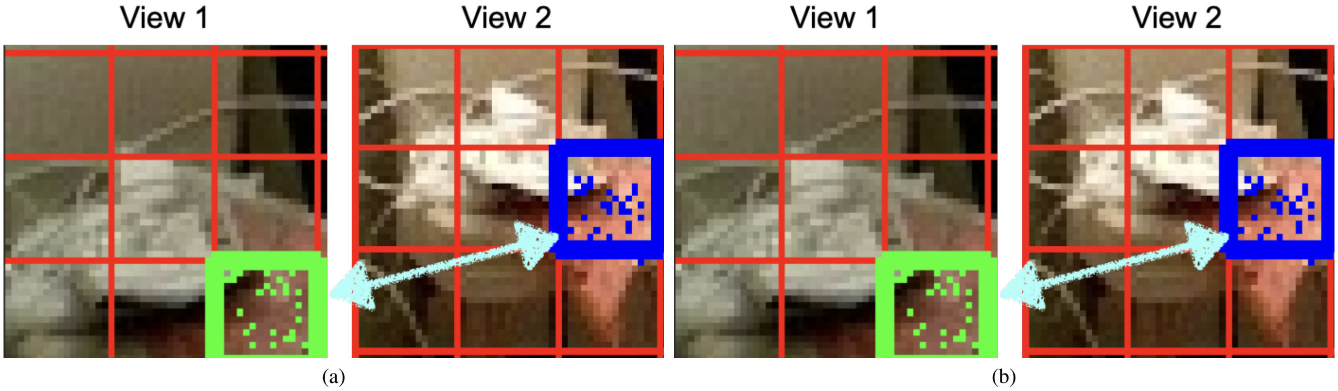


Figure 4. **(a)** The green patch from the view 1 is matched with the blue patch in view 2. **(b)** Two views with their matching patches (matching patches have the same color).

MAE [2]. For pose estimation, we use the ViT-Pose [22] decoders. In Table 2 , we provide the details of the hyperparameters used for finetuning CroCo [20] pretrained on MIMIC-3M on NYUv2 [14], ADE20K [25], Taskonomy [24], MSCOCO [12].

3.2. Error estimates

To estimate the variability associated with our fine-tuned models we compute the error estimates for each of our fine-tuned models. Specifically, we create 100 test sets from each of the down-stream (val/test) datasets by sampling with replacement and then report the minimum, maximum, mean, and standard deviation of the metric in Table 3. Overall we observe that the mean values are close to the numbers reported in the main paper and the standard deviation is small.

3.3. Visualizations of the fine-tuned models

In this section, we provide the visualizations of the depth maps, semantic segmentation masks, surface normal predictions, and pose regression outputs after finetuning CroCo pretrained using MIMIC-3M. For finetuning NYUv2 for depth, ADE20K for semantic segmentation, and Taskonomy for surface normals, we followed Multi-MAE [2] and used the settings from 3.1. For finetuning on MS COCO we used ViTPose [22].

Depth Estimation. Figure 6 shows the input RGB file, predicted depth maps, and ground truth depth maps from the validation set after finetuning on NYUv2.

Semantic Segmentation. Figure 7 shows the RGB images, predicted semantic segmentations, and the ground truth labels from the ADE20K val-

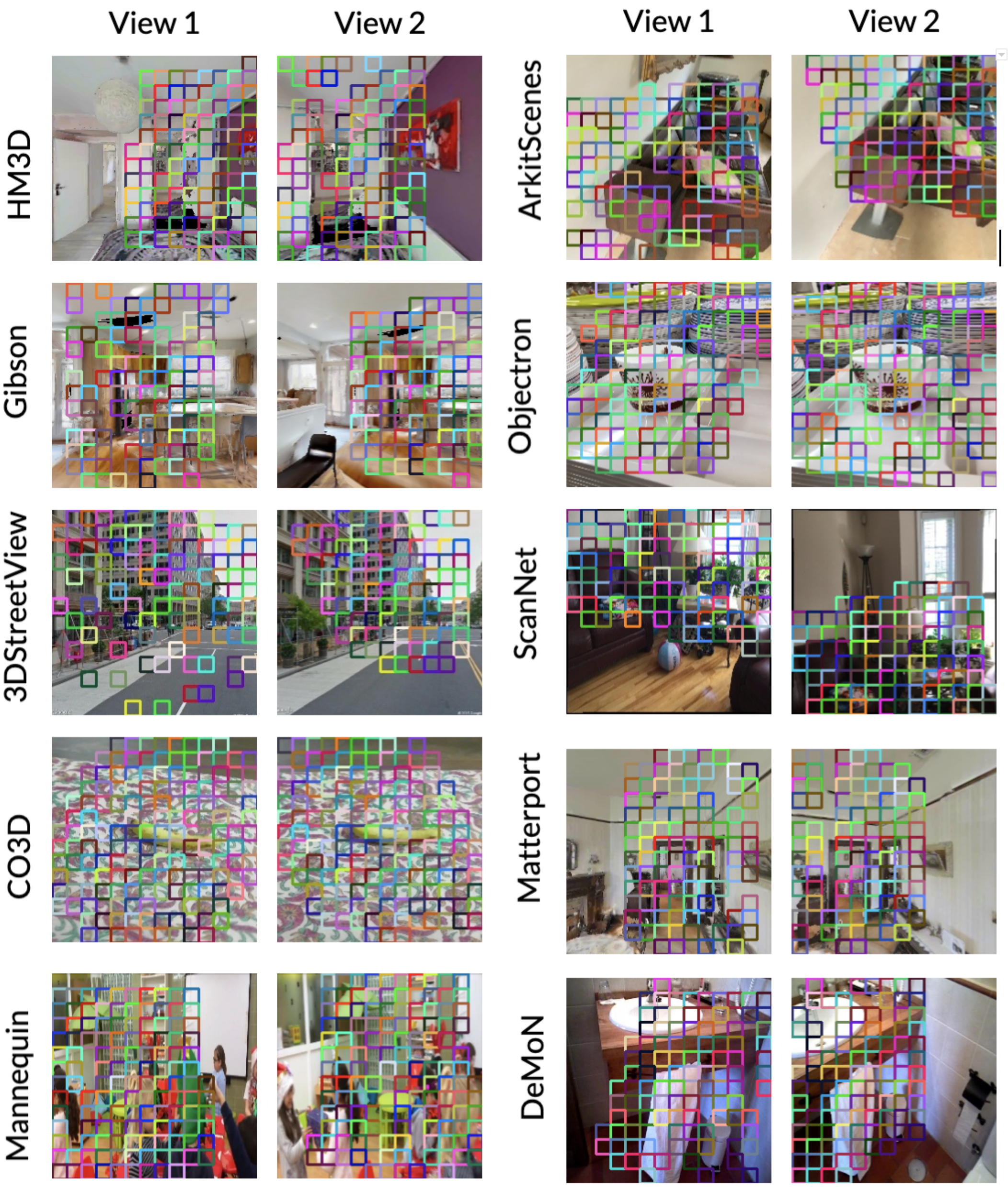


Figure 5. Visualizations of the patchwise correspondences (matching patches have the same color).

idation set after finetuning.

Surface Normals. Figure 8 shows predicted surface normals from the Taskonomy test set after finetuning.

Edges. Figure 9 shows predicted edges from the Taskonomy test set after finetuning.

Curvature. Figure 10 shows predicted curves from the Taskonomy test set after finetuning.

Pose estimation. Figure 11 shows the predicted keypoints from MS COCO validation set after finetuning.

4. Details on the reconstructions experiment

In this study, we collected 500 test image pairs from the Gibson dataset to ensure a fair evaluation process. We made a careful selection to exclude scenes present in the MIMIC-3M dataset, and confirmed that the MULTIVIEW-HABITAT dataset

Table 2. Hyperparameters used for fine-tuning NYUv2 (depth estimation), ADE20K (semantic segmentation), Taskonomy (surface normals), and MSCOCO(pose estimation)

Hyperparameter	NYUv2(depth)	ADE20K(sem.seg.)	Taskonomy (surf.norm.)	MSCOCO(pos.est.)
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	0.0001	0.0005	0.0003	0.0005
Layer-wise lr decay	0.75	0.75	0.75	0.75
Weight decay	0.0003	0.05	0.05	0.1
Adam β	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
Batch size	64	16	8	512
Learning rate schedule.	Cosine decay	Cosine decay	Cosine decay	Linear Decay
Training epochs	2000	64	100	210
Warmup learning rate	-	0.000001	0.000001	0.001
Warmup epochs	100	1	5	500
Input resolution	256 × 256	512 × 512	384 × 384	224 × 224
Augmentation	ColorJitter, RandomCrop	HorizontalFlip, ColorJitter	-	TopDownAffine
Drop path	0.0	0.1	0.1	0.30

Table 3. Error estimates for fine-tuning NYUv2 depth, ADE20K semantic segmentation, Taskonomy surface normal prediction

Task(metric)	Dataset (Val/Test)	Min	Max	Standard Deviation	Mean	Reported value
Depth Estimation (δ_1)	NYUv2 [18]	90.17	92.91	0.56	91.70	91.79
Semantic Segmentation (mIOU)	ADE20K [25]	39.75	43.36	0.75	41.71	42.18
Surface Normal Estimation (L1)	Taxonomy [24]	48.28	54.09	1.24	50.78	53.02

did not include Gibson scenes. Following this, we employed a random masking approach on the target image, utilizing the same masking matrix for inputs of both the model trained on MIMIC-3M and the one trained on MV-Habitat. The purpose of this consistent masking procedure was to enable a comparative assessment of the reconstruction performance on equivalent image patches. Then, each model separately reconstructed the masked target view using the reference view. For the overall reconstruction loss, we got the average over 500 test pairs, which reconstruction loss for each pair was an average of l2 loss over masked pixels. See reconstruction examples of both models in Figure 12.

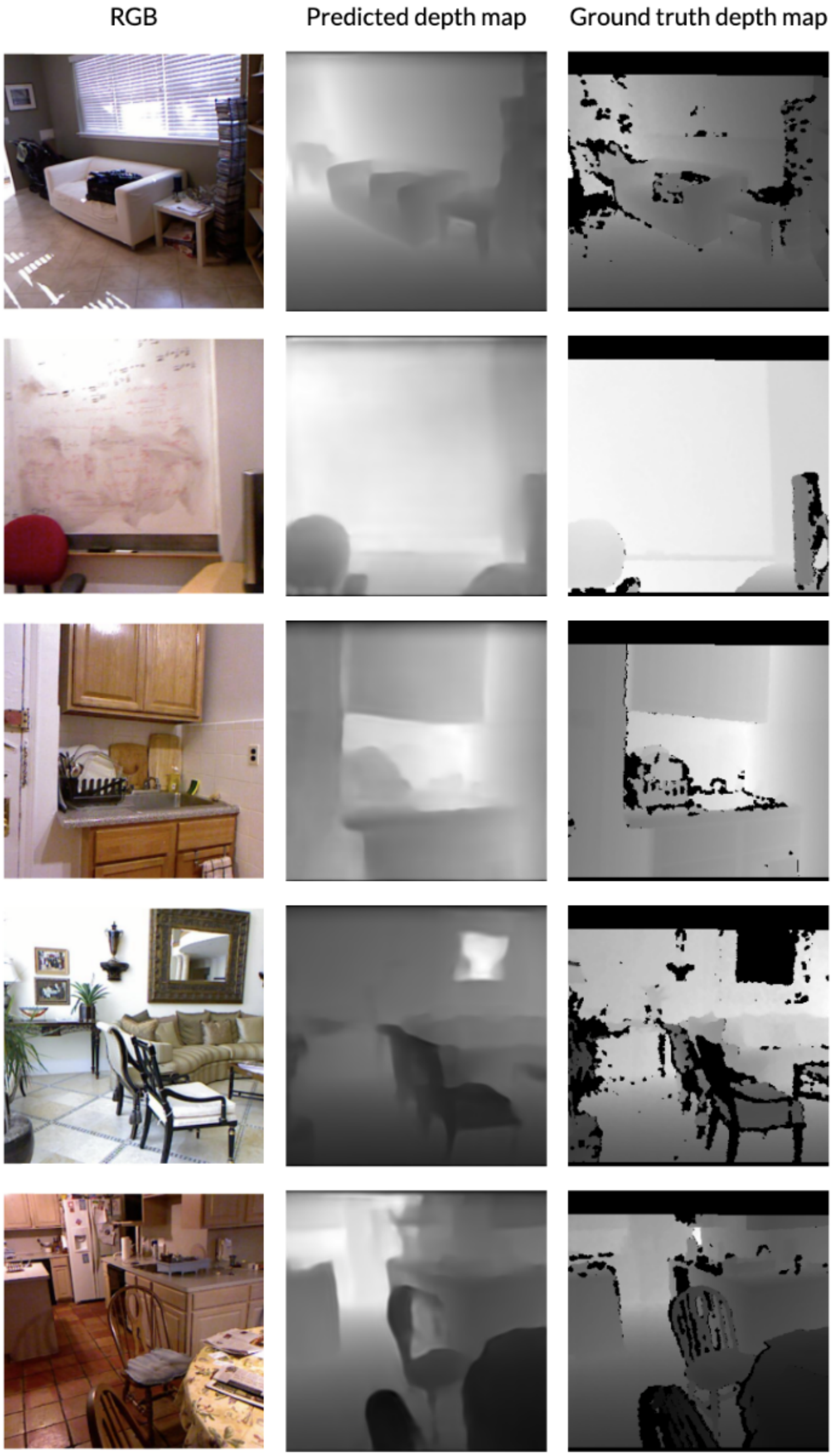


Figure 6. Visualizations of the depth maps

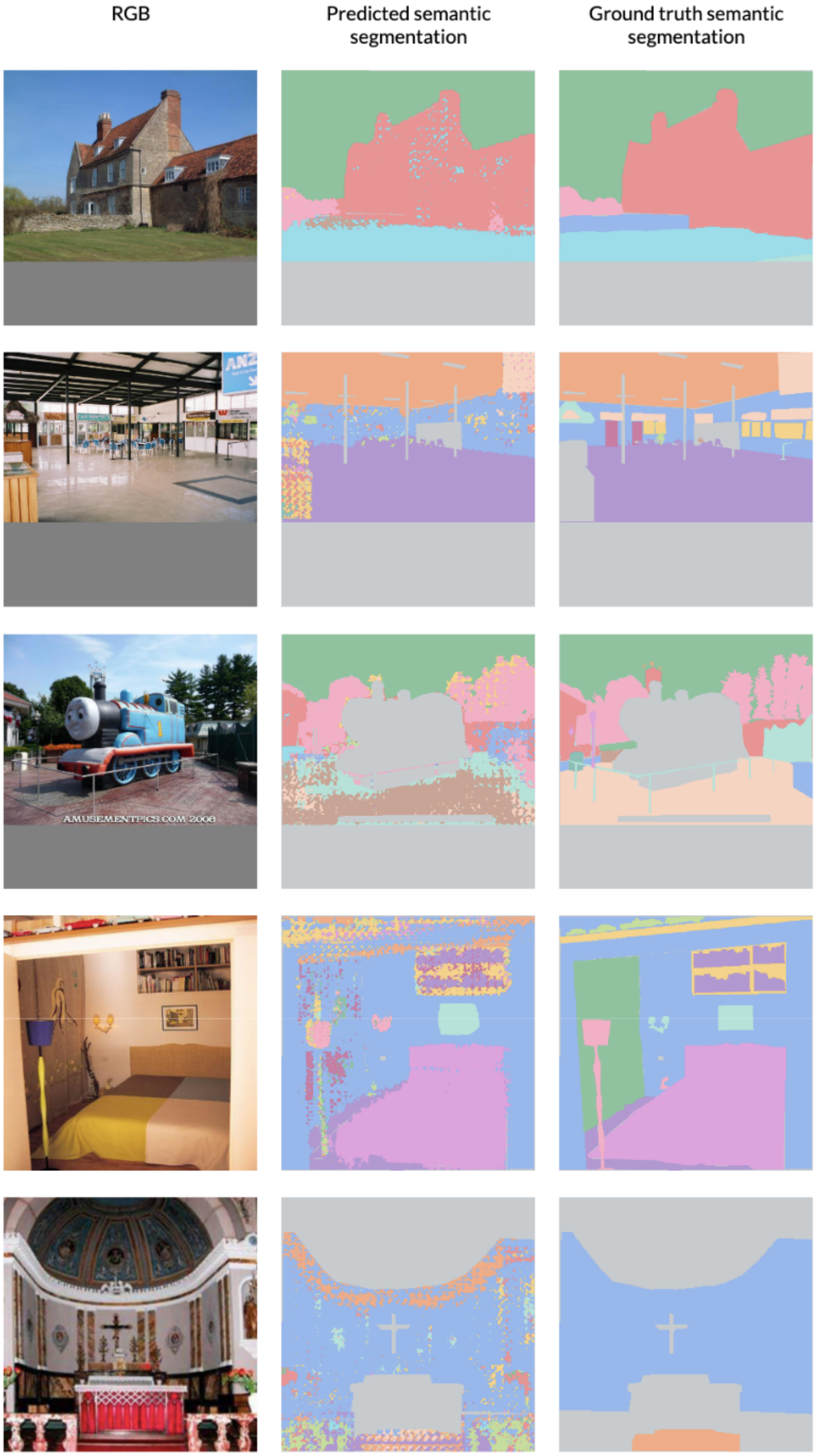

RGB
Predicted semantic segmentation
Ground truth semantic segmentation
AMUSEMENTPICS.COM 2008

Figure 7. Visualizations of the segmentation maps

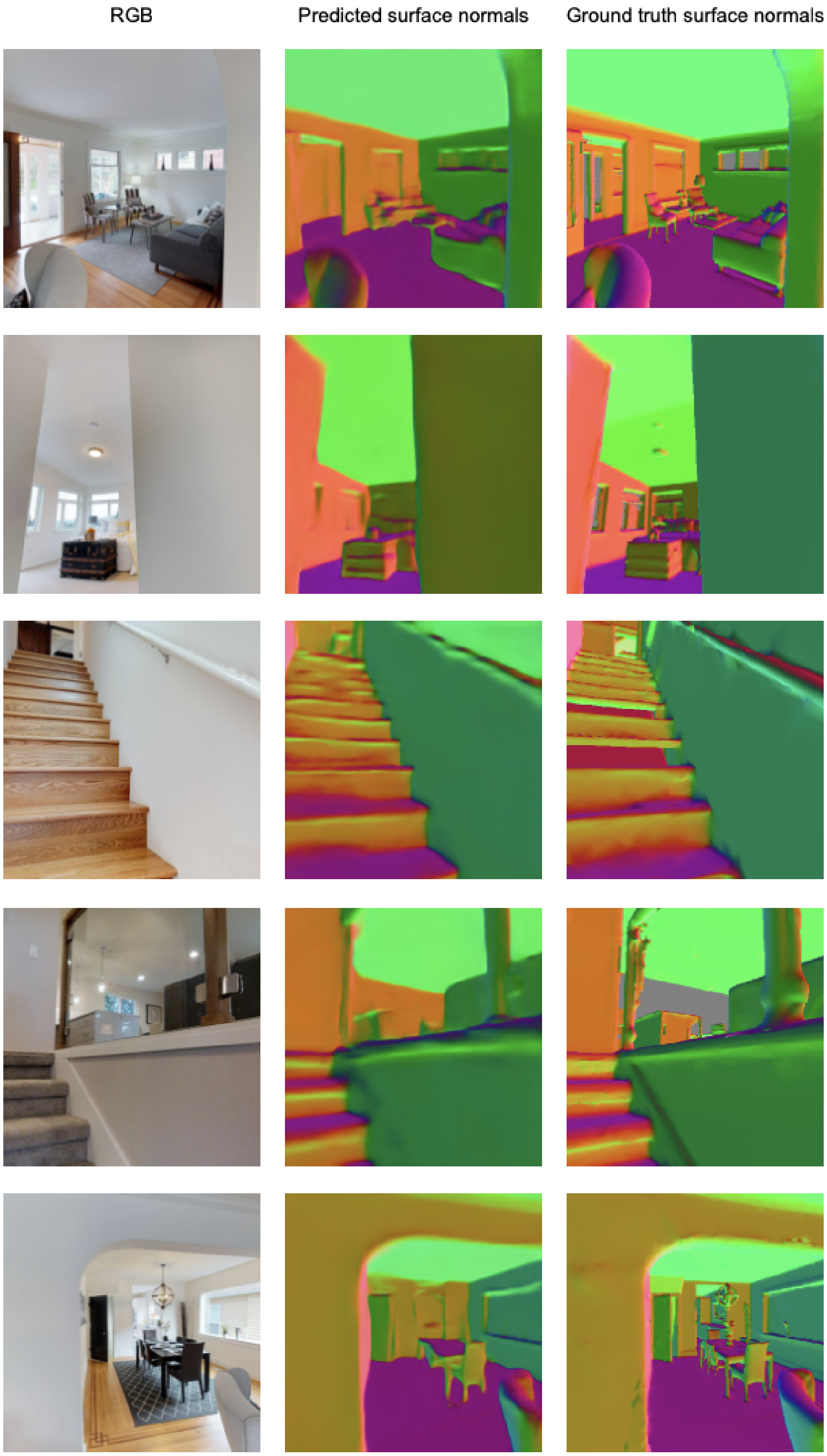


Figure 8. Visualizations of the surface normal predictions

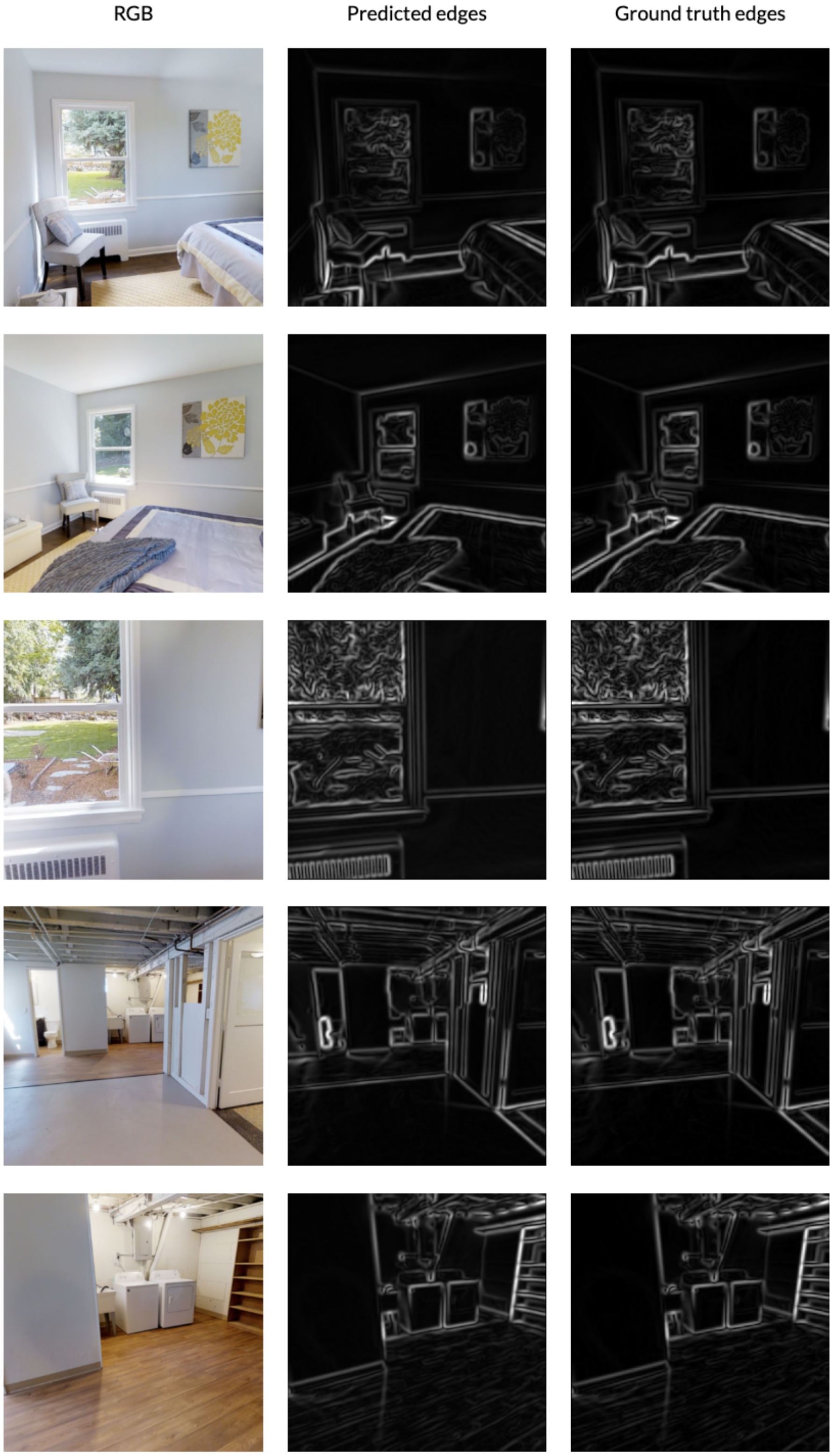


Figure 9. Visualizations of the predicted edges

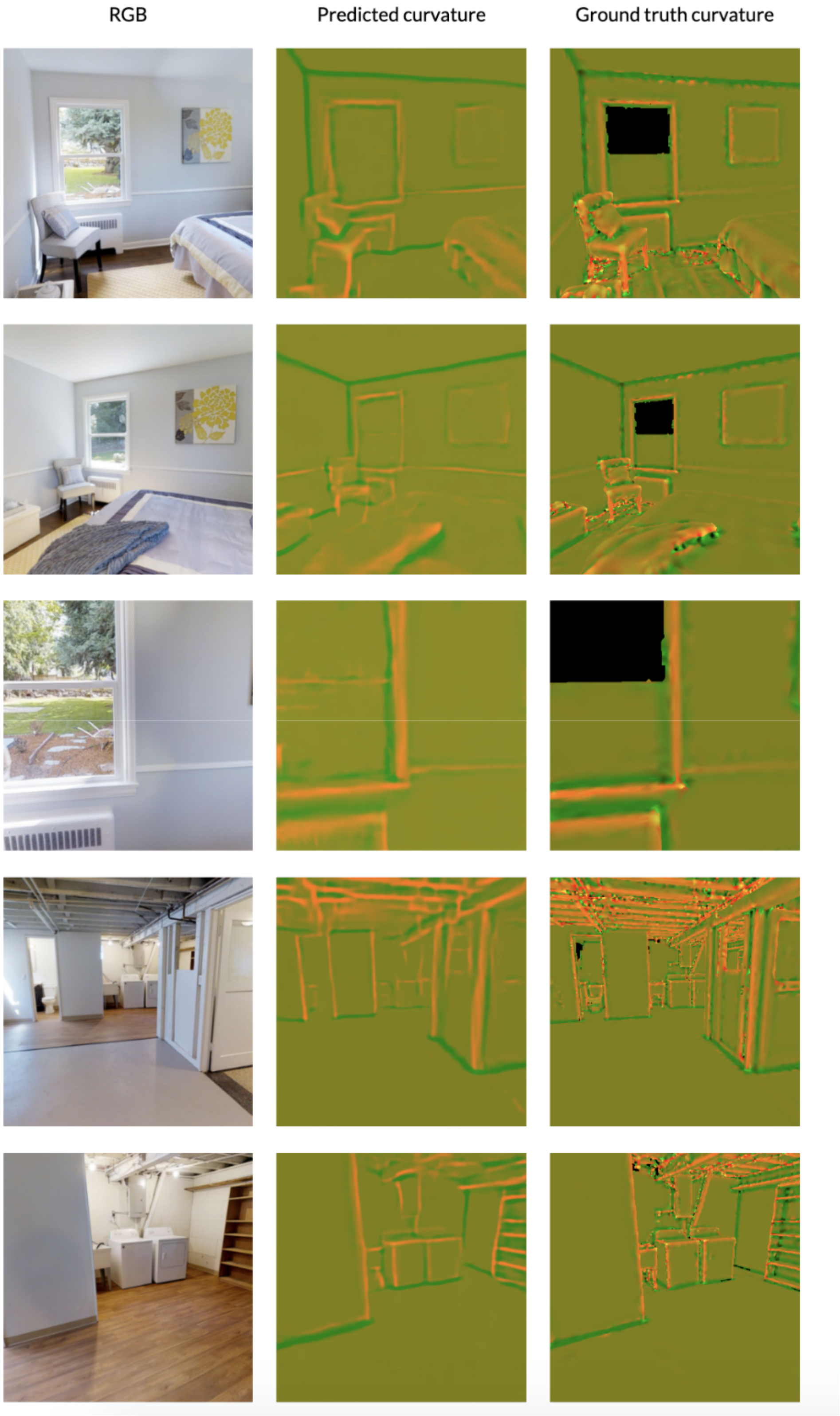


Figure 10. Visualizations of the predicted curvature maps

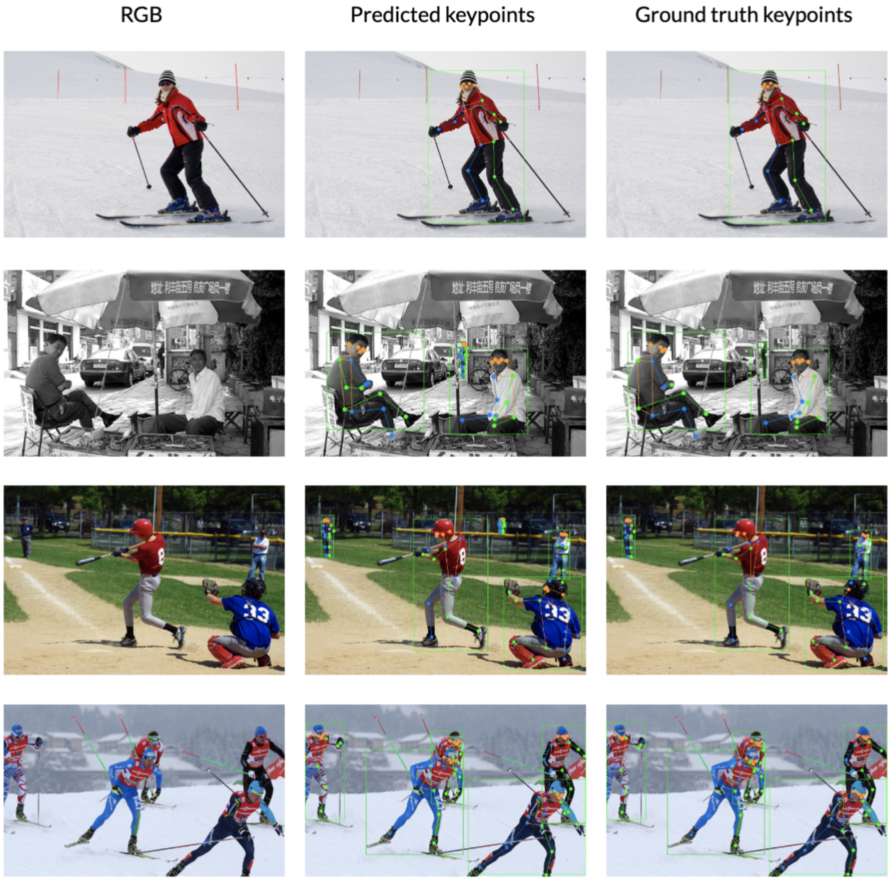


Figure 11. Visualizations of the pose estimation

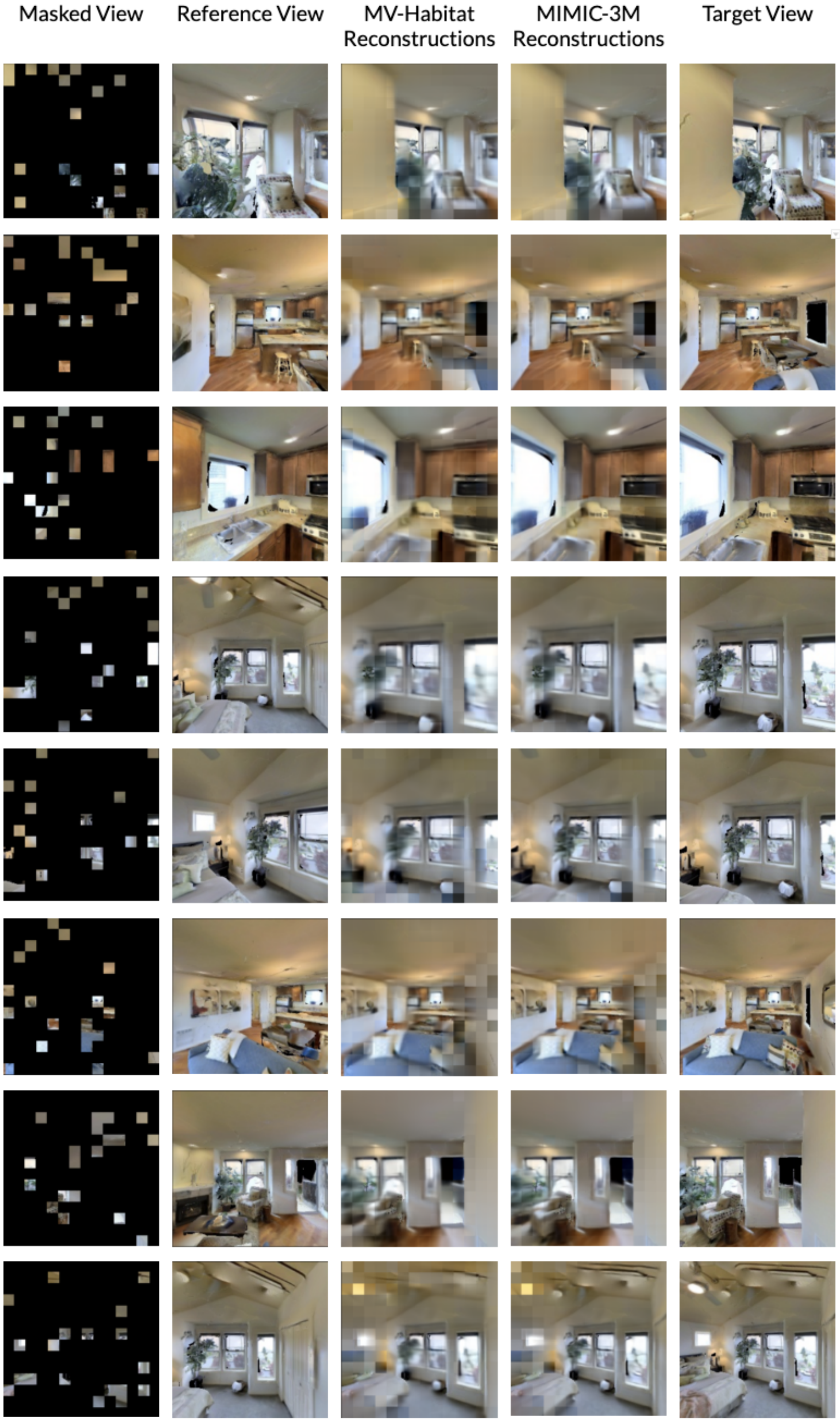


Figure 12. Visualizations of the reconstructions

References

[1] Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7822–7831, 2021. 1

[2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*, pages 348–367. Springer, 2022. 1, 4

[3] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. 1

[4] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1

[5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[7] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2

[8] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[10] 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*, pages 16000–16009, 2022. 1

[11] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4521–4530, 2019. 1

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 4

[13] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 2

[14] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 4

[15] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 1

[16] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 1

[17] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[18] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, 2012. 6

[19] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 1

[20] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022. 1, 4

[21] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 1

[22] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in Neural Information Processing Systems*, 35:38571–38584, 2022. 4

[23] Amir R Zamir, Tilman Wekel, Pulkit Agrawal, Colin Wei, Jitendra Malik, and Silvio Savarese. Generic 3d representation via pose estimation and matching. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The*

Netherlands, October 11-14, 2016, Proceedings, Part III 14, pages 535–553. Springer, 2016. 1

[24] Amir R. Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018. 1, 4, 6

[25] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 1, 4, 6