EscherNet++: A Scalable Multi-View Framework for Amodal Completion, Novel View Synthesis and Feed-Forward 3D Reconstruction

Supplementary Material

This supplementary material provides extended details on feature-level masking in App. A with experimental results. Analysis of hierarchical masking is presented in App. B. In App. C, implementation details for models are discussed, including the detailed setup for scalable integration with pre-trained feed-forward 3D reconstruction models. App. D discusses possible extensions on the generated benchmark OccNVS. Limitation & future work are provided in App. E.

A. Ablation Study on Feature-Level Masking

In experiment, we empirically find the proper ratio for feature-level masking. Consider a batch of feature maps from the image encoder, its tenser shape is $[b * t, l, c]$, in which b is the batch size of samples, t is number of input views in each sample, l is the feature map area (number of feature vectors associated with each input view) and c is the feature dimension.

We start by masking all (100% of $b * t$ dimension) feature maps by half feature map area (50% of l) randomly and the performance is sub-optimal. Then we gradually decrease the ratio on the second dimension by 25% (in $b * t$ dimension), and finally found that 25 % is a proper ratio for feature-level masking. That is, we report performance of the model with 25% masked in $b * t$ dimension and 50% masked in l dimension in training, as the representative results of feature-level masking.

We also attach the full tables for evaluating models with **OccNVS** in the ablation study on feature-level masking. It is found that feature-level masking with proper ratio can improve overall performance including better understanding of semantics and geometry from input views. However, it will lead to sub-optimal performance is too large ratio if picked, as shown is Fig. 7, Tab. 5, Tab. 6, Tab. 7.

B. Empirical Analysis of Hierarchical Masking

1) Input-level masking is similar to compressed sensing [6]. The model is tasked with reconstructing underdetermined input using its generative capacity rather than solving an explicit optimization problem. 2) Feature-level masking resembles the principle of the information bottleneck. The CNN-based encoder extracts feature maps from input viewpoints as condition for the diffusion module. Feature-level masking pushes the encoder to capture richer contextual information beyond local adjacency to compensate for the masked feature vectors during training. 3) Masking ratio plays a critical role. A relatively high masking ratio proves detrimental (App. A). We hypothesize that overly aggressive masking forces the encoder to disproportionately capture global context at the expense of fine-grained local details.

C. Implementation Details of Models in Comparison

We compare our model with several recent SoTA models: Zero-1-2-3, Zero-1-2-3 XL [30] and EscherNet [25] for comparison in NVS tasks; DreamGaussian [47], Large Multi-View Gaussian Model(LGM) [48], SyncDreamer [31], InstantMesh [61] and EscherNet [25] for mesh quality comparison in 3D reconstruction tasks. OccNVS is used for comparison. For 3D reconstruction tasks, raw meshes from the models are normalized first and then compared with ground truth as in [25, 31].

Zero-1-2-3 & Zero-1-2-3 XL: It is the first work in diffusion-based NVS for objects. In its model design, one input view can be referenced at a time and one target view can be synthesized afterwards. As a result, Zero-1-2-3 and its XL version are only adopted for one-input settings.

EscherNet Our model shares the same model structure with EscherNet. The number of input and target views is not restricted, and can vary depending on the specific use cases. It makes use of an existing latent diffusion model [42], pre-trained on web-scale data, with U-Net [43] composed of residual blocks [11] and transformer blocks [51] as the backbone. It conditions its generation process on input visual information from a lightweight CNN-based vision encoder [57], and pose information from its camera positional encoding (CaPE). Low-resolution latents are decoded to images [24] as in [42].

As the result, EscherNet can be used for direct comparison in all tasks and settings in this paper, including NVS and 3D reconstruction. For NVS, EscherNet is able to synthesize multiple novels view from any query viewpoints. For 3D reconstruction, 36 fixed view are synthesized, with the azimuth from 0°to 360°with a rendering every 30°at a set of elevations (-30°, 0°, 30°) for reconstruction with NeuS, the same setting as reconstruction with our model.

We fine-tune our model based on public weights shared by authors of Eschernet, and we have confirmed with them about the performance of EscherNet in the experiments.

DreamGaussian: It is a two-stage model, which uses the first stage for reconstruction conditioned on a single input view and second image for texture refinement. Hence, there

Figure 7. Qualitative results with different ratios for feature-level masking.

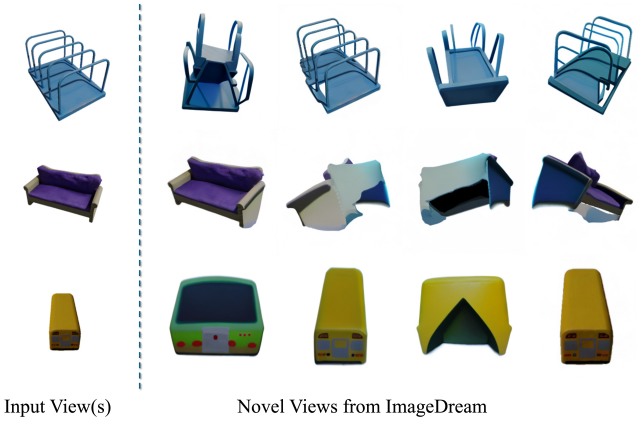

Figure 8. Examples of novel views generated by ImageDream. It struggles with significant elevations and azimuths. Therefore, the challenge is propagated to the reconstruction pipeline of LGM.

are no novel views required before reconstruction. Rotation is conducted for evaluation as in EscherNet. It is worth noting that DreamGaussian is among the fastest methods for reconstruction in our experiment.

LGM: As a two-stage method, LGM [48] depends on four views from fixed viewpoints synthesized by ImageDream [54] conditioned on one input view to reconstruct 3D. It is also a fast pipeline, however, it is found to struggle with significant elevation and azimuth angles in input views. Therefore, it does not perform well in our tests. The fundamental reason is that ImageDream may not be able to

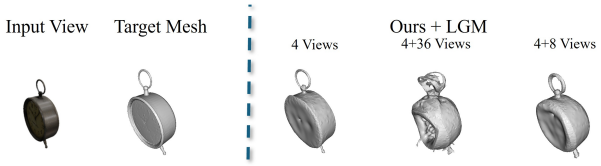

Figure 9. An examples of the training-free scalable reconstruction study with LGM. Different numbers of synthesized views are given.

provide consistent and reasonable novel views when conditioned on inputs with significant angles, as shown in Fig. 8. The same rotation mechanism is conducted as with DreamGaussain.

We further explore training-free scalability in the reconstruction phase with LGM. As shown in Sec. 3.3, it is found that **Goal 1** can be achieved naturally with our NVS model. However, **Goal 2** is difficult for LGM, because of its direct fusing mechanism as we illustrate in Eq. 4. We additionally synthesize views at zero elevation, following the setup in ImageDream, with each view spaced 30°apart. While this strategy mitigates the issues associated with non-zero-elevation views, it still fails to achieve **Goal 2** for scalability, with an example in Fig. 9. The fundamental reason is the lack of a unified underlying representation. The quantized performance can be found in Tab. 4.

SyncDreamer: 16 fixes views are synthesized conditioned on one input view and then given to NeuS [55] by SyncDreamer [31]. Compared with reconstruction time

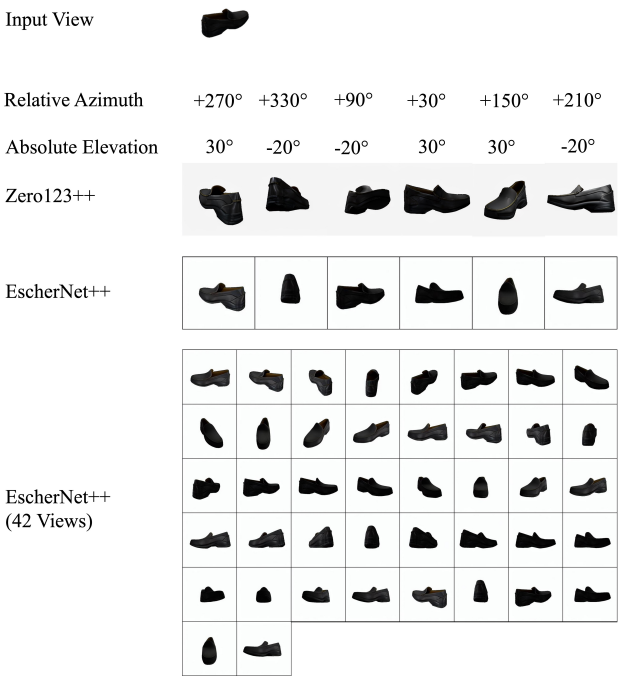

Figure 10. Examples of novel views generated by Zero123++ and EscherNet++ for reconstruction by InstantMesh. The last row contains all 42 views by our model. The scale and pose of the object in novel views by Zero123++ are not consistent sometimes, which can lead to confusion for InstantMesh.

which usually takes near 30 minutes, the time spent on synthesis is almost insignificant. That is, the time used to reconstruct an object from one input view to a complete mesh is largely dependent on the reconstruction method, which shares a similar case with reconstitution based on our model with overfitting methods like NeuS.

InstantMesh: In the original pipeline, Zero123++ Shi et al. [44] is used for NVS at the first stage and InstantMesh [61] construct the mesh based on novel views. Zero123++ is designed to generate 6 fixed views of an object with relative azimuth rotations and absolute elevations. The 6 input images have poses with alternating absolute elevations of 20°and -10°, and their azimuths are defined relative to the query image, beginning at 30°and increased by 60°for subsequent poses. However, it sometimes generate meshes with floaters around the object, which leads to erroneous scale in normalization, as shown in Fig. 6. It is found that we can make use of our model to generate more consistent novel views at the preferred viewpoints for InstantMesh so that the performance can be improved significantly without floaters in the final meshes. The performance can be further enhanced by providing more novel views covering more viewpoints to InstantMesh. We provide one example comparing novel views from Zero123++ and our method in Fig. 10. No extra training or extra reference time is induced in this whole process.

Method	# Ref. Views	# Nol. Views	GSO3D		Occluded GSO3D		Time
			Chamfer Dist. ↓	Volume IoU ↑	Chamfer Dist. ↓	Volume IoU ↑	Minutes ↓
Zero123++[44]+InstantMesh[61]	1	6	0.0608	0.4557	0.0655	0.2478	1.6
ImageDream[54]+LGM[48]	1	4	0.0877	0.2521	0.1787	0.095	1.5
Ours + NeuS	1	36	0.0305	0.6018	0.0376	0.5602	27
	2	36	0.0214	0.6921	0.0249	0.664	
	3	36	0.0185	0.7277	0.0197	0.7139	
	5	36	0.0182	0.7294	0.0189	0.7221	
	10	36	0.0168	0.7437	0.0176	0.7352	
Ours + InstantMesh	1	6	0.0304	0.5912	0.0392	0.5405	1.3
	2	6	0.0259	0.633	0.0301	0.5954	
	3	6	0.0251	0.6491	0.0257	0.6413	
	5	6	0.0238	0.6667	0.0291	0.6376	
	10	6	0.0275	0.6472	0.0282	0.6414	
Ours + InstantMesh	1	42	0.0278	0.6244	0.04	0.5501	1.3
	2	42	0.0224	0.6803	0.0311	0.6118	
	3	42	0.0265	0.6744	0.0277	0.6605	
	5	42	0.0253	0.6857	0.024	0.6886	
	10	42	0.0179	0.7295	0.0233	0.6987	
Ours + LGM	1	4	0.0337	0.57	0.0414	0.5099	1.5
	2	4	0.0293	0.613	0.031	0.5847	
	3	4	0.0269	0.6247	0.0283	0.6223	
	5	4	0.0256	0.6242	0.0266	0.6127	
	10	4	0.0248	0.6471	0.0264	0.6345	
Ours + LGM	1	40	0.0515	0.3128	0.0558	0.2724	1.5
	2	40	0.051	0.3186	0.0517	0.3395	
	3	40	0.0498	0.3012	0.0495	0.3101	
	5	40	0.0511	0.3437	0.0518	0.3596	
	10	40	0.0497	0.3107	0.0501	0.3201	
Ours + LGM	1	12	0.0382	0.513	0.0413	0.4452	1.5
	2	12	0.0302	0.5588	0.0325	0.5471	
	3	12	0.0294	0.5707	0.0302	0.5907	
	5	12	0.0288	0.594	0.0297	0.595	
	10	12	0.0293	0.5889	0.0284	0.6129	

Table 4. 3D reconstruction comparison from InstantMesh and LGM on GSO3D and Occluded GSO3D datasets, with integration with different NVS models, and different numbers of synthesized views from EscherNet++.

Although it is able to provide views from any viewpoints, we find that the six viewpoints used in the original pipeline and their absolute values are necessary to the network. Therefore, we define that the input views are at 0°azimuth angle and we rotate the meshes back before evaluation.

As noticed by authors of InstantMesh, InstantMesh is able to take in various numbers of input views because of its transformer-based structure. However, in contrast to their finding that decrease the number of input views can boost the performance in some hard cases, we find with our model, simply increasing the number of input views can further improve the overall reconstruction performance without extra overheads, as shown in Tab. 4, thanks to the ability to synthesize high-quality views from any query viewpoints from our model and the unified underlying representation adopted in this reconstruction model.

D. Extension on OccNVS

OccNVS is a flexible and extensible benchmark for generating paired occluded and complete views, enabling controlled evaluation of novel view synthesis under occlusion. It efficiently simulates diverse occlusion scenarios by varying viewpoint and occluder distance relative to the target object.

We outline two key directions for extending **OccNVS**: 1) Scale. Expand the dataset by including more target objects and occlusion masks to increase category diversity and geometric complexity. 2) Realism for near-object oc-

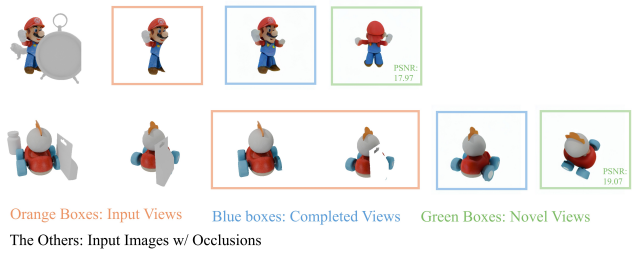

Figure 11. Examples of Possible extension on OccNVS. Occluders can be randomized and the occluded views can be rendered to test the model.

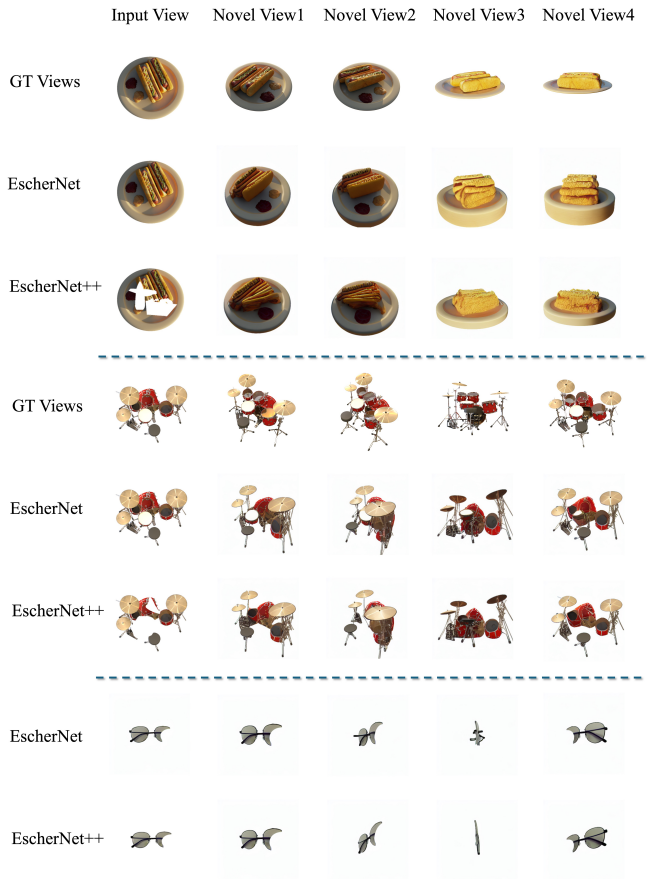

Figure 12. Examples of typical error cases of EscherNet++.

clusion. Introduce randomized near-object occluder placement, lighting variations, and naturalistic occlusion patterns to better reflect real-world conditions as shown in Fig. 11.

E. Limitations & Future Work

During our experiments, we identified two notable limitations in our model's performance.

1) EscherNet++ exhibits degraded performance on inputs that contain intricate details and complex spatial layouts, similar to the performance of the base model EscherNet. This is evident in the first two rows of Fig. 12, where despite the model's ability to infer novel views and fill in occluded regions, the synthesized outputs lack both visual and semantic consistency. Fine-grained textures and structural elements are either oversimplified or inconsistently rendered across views, suggesting that the models struggle to maintain coherence when completing regions with high-frequency details or complex shapes, partially because of the limited resolution (256*256) of both input and output views. 2) It possibly presents failure when presented with out-of-distribution (OOD) occluded inputs. The third row in Fig. 12 provides an example: the input includes a partially visible pair of eyeglasses, yet the model fails to recognize the object's class or underlying geometry. Instead of completing the missing regions in a plausible way, it generates a flat, unstructured form that resembles a piece of paper. The underlying reason is the lack of understanding in OOD inputs and possible occlusions.

Future work can explore robust architecture designs with more diverse datasets, more explicit guidance with multi-modal inputs, including more expressive visual and semantic features. A larger/high-quality dataset are recommended to fully exploit the potential of the model. Feed-forward 3D reconstruction methods also have the potential to be improved in terms of how to increase robustness to inconsistency in inputs views and utilize increasing number of views more efficiently. Last, a comprehensive framework is necessary to make our work more accessible in applications that includes object segmentation, pose estimation, etc, combined as integrated modules or a single unified model to achieve scene-level reconstruction. Concurrent works, such as Amodal3R [59] and SAM-3D [3], also highlight the promise of this direction.

Table 5. Performance comparison on GSO-30 and Occluded GSO-30 datasets with different ratios for feature-level masking.

Base Method	Input-Level Masking Ratio	Feature-Level Masking Ratio	# Ref. Views	GSO-30			Occluded GSO-30		
				PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
EscherNet (ckpt)	0.5	1.0	1	19.62	0.879	0.1	19.11	0.874	0.11
EscherNet (ckpt)	0.5	1.0	2	22.21	0.903	0.067	21.36	0.897	0.076
EscherNet (ckpt)	0.5	1.0	3	23.54	0.915	0.054	22.58	0.908	0.061
EscherNet (ckpt)	0.5	1.0	5	24.51	0.922	0.046	23.81	0.917	0.051
EscherNet (ckpt)	0.5	1.0	10	25.41	0.93	0.039	24.68	0.926	0.043
EscherNet (ckpt)	0.5	0.75	1	19.68	0.879	0.099	19.21	0.875	0.108
EscherNet (ckpt)	0.5	0.75	2	22.4	0.905	0.066	21.47	0.898	0.074
EscherNet (ckpt)	0.5	0.75	3	23.78	0.916	0.053	22.65	0.908	0.061
EscherNet (ckpt)	0.5	0.75	5	24.82	0.924	0.044	23.89	0.918	0.05
EscherNet (ckpt)	0.5	0.75	10	25.71	0.933	0.038	24.84	0.927	0.042
EscherNet (ckpt)	0.5	0.5	1	19.93	0.883	0.095	19.27	0.877	0.107
EscherNet (ckpt)	0.5	0.5	2	22.72	0.907	0.063	21.76	0.9	0.072
EscherNet (ckpt)	0.5	0.5	3	23.87	0.917	0.051	22.97	0.91	0.059
EscherNet (ckpt)	0.5	0.5	5	24.93	0.925	0.043	24.04	0.919	0.049
EscherNet (ckpt)	0.5	0.5	10	25.88	0.933	0.037	24.95	0.927	0.041
EscherNet (ckpt)	0.5	0.25	1	20.11	0.883	0.094	19.72	0.879	0.103
EscherNet (ckpt)	0.5	0.25	2	22.83	0.908	0.062	21.86	0.902	0.07
EscherNet (ckpt)	0.5	0.25	3	24.02	0.918	0.051	23.22	0.913	0.056
EscherNet (ckpt)	0.5	0.25	5	25.15	0.926	0.043	24.22	0.921	0.047
EscherNet (ckpt)	0.5	0.25	10	25.98	0.934	0.036	25.06	0.929	0.04
EscherNet (ckpt)	0.5	0	1	19.95	0.88	0.1	19.31	0.875	0.109
EscherNet (ckpt)	0.5	0	2	22.72	0.907	0.064	21.65	0.9	0.073
EscherNet (ckpt)	0.5	0	3	23.93	0.917	0.052	22.97	0.91	0.059
EscherNet (ckpt)	0.5	0	5	25.05	0.926	0.043	23.98	0.919	0.049
EscherNet (ckpt)	0.5	0	10	25.85	0.934	0.037	24.77	0.927	0.042

Table 6. Performance comparison on RTMV and Occluded RTMV datasets with different ratios for feature-level masking.

Base Method	Input-Level Masking Ratio	Feature-Level Masking Ratio	# Ref. Views	RTMV			Occluded RTMV		
				PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
EscherNet (ckpt)	0.5	1.0	1	10.62	0.532	0.401	10.37	0.525	0.414
EscherNet (ckpt)	0.5	1.0	2	12.38	0.58	0.31	12.14	0.574	0.322
EscherNet (ckpt)	0.5	1.0	3	13.23	0.606	0.267	13.02	0.6	0.279
EscherNet (ckpt)	0.5	1.0	5	14.23	0.628	0.232	13.94	0.62	0.243
EscherNet (ckpt)	0.5	1.0	10	15.2	0.654	0.192	14.96	0.648	0.201
EscherNet (ckpt)	0.5	0.75	1	10.29	0.522	0.418	10.12	0.518	0.428
EscherNet (ckpt)	0.5	0.75	2	12.3	0.577	0.316	12.17	0.576	0.32
EscherNet (ckpt)	0.5	0.75	3	13.3	0.606	0.267	13.1	0.6	0.278
EscherNet (ckpt)	0.5	0.75	5	14.3	0.63	0.227	14.01	0.623	0.239
EscherNet (ckpt)	0.5	0.75	10	15.17	0.652	0.193	14.9	0.647	0.203
EscherNet (ckpt)	0.5	0.5	1	10.37	0.521	0.415	10.23	0.518	0.42
EscherNet (ckpt)	0.5	0.5	2	12.3	0.575	0.318	12.08	0.571	0.327
EscherNet (ckpt)	0.5	0.5	3	13.23	0.604	0.272	13.1	0.599	0.28
EscherNet (ckpt)	0.5	0.5	5	14.26	0.628	0.229	14.02	0.622	0.24
EscherNet (ckpt)	0.5	0.5	10	15.21	0.652	0.192	14.93	0.645	0.202
EscherNet (ckpt)	0.5	0.25	1	10.5	0.523	0.408	10.34	0.52	0.416
EscherNet (ckpt)	0.5	0.25	2	12.57	0.583	0.303	12.32	0.577	0.316
EscherNet (ckpt)	0.5	0.25	3	13.45	0.608	0.262	13.29	0.603	0.269
EscherNet (ckpt)	0.5	0.25	5	14.38	0.631	0.223	14.16	0.627	0.232
EscherNet (ckpt)	0.5	0.25	10	15.42	0.658	0.186	15.13	0.652	0.196
EscherNet (ckpt)	0.5	0	1	10.78	0.53	0.391	10.57	0.526	0.405
EscherNet (ckpt)	0.5	0	2	12.57	0.582	0.301	12.26	0.575	0.315
EscherNet (ckpt)	0.5	0	3	13.5	0.609	0.259	13.31	0.609	0.259
EscherNet (ckpt)	0.5	0	5	14.37	0.63	0.223	14.12	0.624	0.233
EscherNet (ckpt)	0.5	0	10	15.38	0.658	0.185	15.08	0.65	0.195

Table 7. Performance comparison on NeRF and Occluded NeRF datasets with different ratios for feature-level masking.

Base Method	Input-Level Masking Ratio	Feature-Level Masking Ratio	# Ref. Views	NeRF			Occluded NeRF		
				PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
EscherNet (ckpt)	0.5	1.0	1	13.43	0.657	0.292	13.5	0.661	0.295
EscherNet (ckpt)	0.5	1.0	2	14.99	0.696	0.214	14.72	0.688	0.229
EscherNet (ckpt)	0.5	1.0	3	16.19	0.728	0.166	15.87	0.722	0.178
EscherNet (ckpt)	0.5	1.0	5	17.01	0.744	0.133	16.71	0.738	0.143
EscherNet (ckpt)	0.5	1.0	10	17.46	0.754	0.121	17.19	0.749	0.128
EscherNet (ckpt)	0.5	0.75	1	13.37	0.659	0.3	13.9	0.671	0.282
EscherNet (ckpt)	0.5	0.75	2	14.93	0.695	0.214	14.66	0.688	0.229
EscherNet (ckpt)	0.5	0.75	3	16.19	0.727	0.166	15.87	0.721	0.177
EscherNet (ckpt)	0.5	0.75	5	17.12	0.747	0.13	16.74	0.739	0.141
EscherNet (ckpt)	0.5	0.75	10	17.53	0.756	0.119	17.26	0.751	0.126
EscherNet (ckpt)	0.5	0.5	1	13.43	0.659	0.295	13.47	0.659	0.3
EscherNet (ckpt)	0.5	0.5	2	14.85	0.695	0.212	14.66	0.689	0.224
EscherNet (ckpt)	0.5	0.5	3	16.14	0.727	0.164	15.84	0.721	0.176
EscherNet (ckpt)	0.5	0.5	5	16.97	0.745	0.132	16.69	0.738	0.142
EscherNet (ckpt)	0.5	0.5	10	17.4	0.754	0.121	17.16	0.749	0.128
EscherNet (ckpt)	0.5	0.25	1	13.35	0.661	0.29	13.51	0.666	0.29
EscherNet (ckpt)	0.5	0.25	2	14.96	0.698	0.21	14.74	0.692	0.221
EscherNet (ckpt)	0.5	0.25	3	16.14	0.727	0.164	15.85	0.721	0.174
EscherNet (ckpt)	0.5	0.25	5	16.97	0.745	0.132	16.79	0.74	0.138
EscherNet (ckpt)	0.5	0.25	10	17.72	0.759	0.115	17.49	0.755	0.121
EscherNet (ckpt)	0.5	0	1	13.47	0.658	0.289	13.57	0.66	0.295
EscherNet (ckpt)	0.5	0	2	14.98	0.697	0.211	14.69	0.691	0.226
EscherNet (ckpt)	0.5	0	3	16.25	0.729	0.163	15.91	0.721	0.175
EscherNet (ckpt)	0.5	0	5	17.22	0.749	0.128	16.86	0.742	0.138
EscherNet (ckpt)	0.5	0	10	17.7	0.76	0.116	17.43	0.754	0.123