

Optimizing NeRFs for Dynamic Scenes with Image Inpainting and Motion Segmentation

Lilian Chen

Stanford University

`lilianch@stanford.edu`

Abstract

In recent years, Neural Radiance Fields (NeRFs) [1] have gained considerable attention for its capabilities in implicit neural scene representation and novel view synthesis. Particularly, NeRFs have found applications in a range of domains, including robotics, augmented reality, motion planning, and autonomous driving contexts. However, despite its ability to achieve state-of-the-art visual quality on generated novel-view images, such quality often can only be achieved in static context scenes. Therefore, given more dynamic scenes, such as different lighting conditions and moving objects in the background, often leads to diminished reconstruction quality with NeRF. This can be especially problematic in autonomous driving scenarios, for example, which observes diverse lighting and weather conditions, in addition to dynamic objects such as other cars and pedestrians. This work thus explores methods to efficiently preprocess an input set of images and camera poses and feed it into the NeRF pipeline. By ensuring that the input information is more consistent, the NeRF model can more effectively learn and render the scene, leading to better overall performance and visual quality. We show that by our processing methods, we are able to achieve a heightened reconstruction quality with fewer artifacts.

1. Introduction

Neural Radiance Fields, which was first pioneered by Mildenhall et al. in 2020 [5], achieved quality novel view synthesis by training a deep network which maps 5D input coordinates consisting of spatial location (x, y, z) and viewing direction (θ, ϕ) to an output of volume density and view-dependent emitted radiance at that spatial location. Thus, feeding this 3D encoded scene into a multi-layer perceptron, we can synthesize novel viewpoints by querying 5D coordinates along the camera rays and utilize classic volume rendering techniques to project the output colors and densities into an image.

However, NeRF’s ability to construct novel photo-realistic views operates under the assumption that the scene is static—that is, without moving objects in its input. Specifically, because NeRFs require consistent training data in order to overfit to a scene, the viewpoints used to train the model should correspond to the same scene, and the presence of a moving object within the scene would therefore introduce inconsistencies across different images taken from various viewpoints, confusing the model and leading to artifacts or blurry reconstructions. Furthermore, accurately modeling dynamic scenes would require not only understanding the geometry and appearance but also the temporal dynamics of object, adding a layer of complexity that NeRFs, with its original formulation, would not be equipped to handle. And as expected, such assumption on static scenes has been problematic in NeRFs applications to the real world because a single given scene can vary immensely from time to time, even seconds apart. For example, the lighting would change as the sun moves across the sky, the weather may be different from day to day, and people in the background will be constantly moving. These changes especially apply in autonomous vehicle applications, where moving cars, pedestrians, and bikers, among many other dynamic entities, are common features encountered in driving scenarios.

To handle dynamic scenes, research methods have included time-aware extensions which incorporates temporal information [8] and/or geometry priors [15] in order to reduce artifacts left from moving objects. To address this challenge, we propose a NeRF framework leveraging image segmentation, where we reconstruct non-static input scenes by explicitly removing non-scene dynamic objects and then perform occlusion inpainting, as shown in Fig 1. Specifically, we use Mask-RCNN for semantic segmentation, enhanced with moving shadow detection, to identify and exclude objects that are not part of the static scene, such as people and bicycles. After segmentation, we apply DeepFill V2, an advanced inpainting algorithm, to recreate the background occluded by these dynamic objects. Finally, we input the processed images and utilize

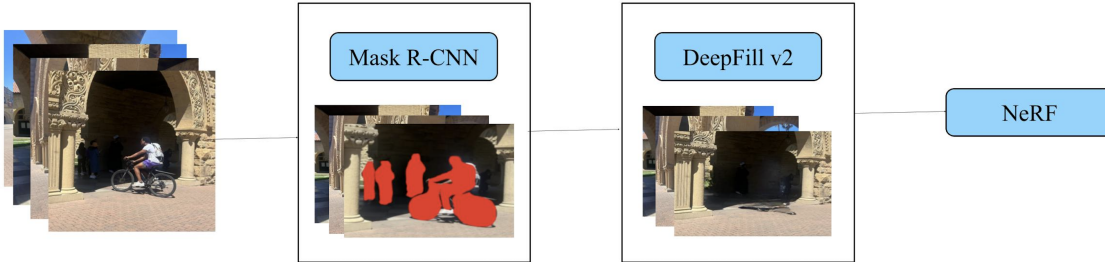


Figure 1: Our Method

Structure-from-Motion for pose estimation and perform NeRF reconstruction to produce high-quality images from new perspectives.

2. Related Works

2.1. Dynamic Scene Handling in NeRFs

Dynamic Scene Handling in Neural Radiance Fields is a research area focused on extending NeRFs to handle scenes with moving objects and other such temporal changes. While NeRFs have shown remarkable success in synthesizing photorealistic views of static scenes, its performance degrades significantly in dynamic environments, yielding artifacts in the reconstructed scenes. In this section, we review key advancements and methodologies thus far which have addressed the challenges posed by dynamic scenes in NeRFs.

2.1.1 Geometry Prior Utilization

Incorporating geometry priors into NeRFs have been a well-explored method to enhance dynamic scene handling, as it allows insight into the 3D structure of a scene, which is crucial for accurately modeling occlusions. When dynamic objects move, understanding the underlying geometry helps in determining what parts of the scene should be visible or hidden from different viewpoints, therefore assisting with maintaining a more consistent representation of the scene’s structure. Geometry priors also generally allow better generalization across different viewpoints and object poses, especially as objects can appear in various orientations and positions in dynamic scenes. A robust geometric prior helps the model understand these variations, enabling it to predict unseen views more accurately and handle new object configurations effectively. GeoNeRF employs a geometry reasoner to obtain fine and high-resolution geometry priors, showing its usefulness in enabling sophisticated occlusion reasoning and detailed image rendering via classical volume rendering techniques [3]. H-NeRF is an instance of using geometric priors to target rendering and tempo-

ral reconstruction of humans in motion, integrating neural scene representation and implicit statistical geometric human models using signed distance functions [15]. These advancements highlight the crucial role of geometry priors in improving NeRF performance for dynamic scenes, addressing key challenges such as occlusions, motion handling, and geometric consistency. However, the integration of geometric priors implies having access to 3D ground-truth geometry, which is often expensive or impossible to obtain for scenes encountered in the wild.

2.1.2 Temporal Coherence Integration

Ensuring the consistency and smoothness of visual properties over time, given a sequence of dynamic scenes, is crucial for ensuring that elements within a reconstructed scene appear stable and avoiding visual artifacts such as flickering or jittering. D-NeRF is a notable approach that extends NeRF to dynamic scenes, including time as an additional input to the system and decomposing learning into a canonical scene and scene flow, ultimately being able to render high-quality images for scenes with non-rigid objects [9]. Neural Scene Flow Fields are based on a variation of NeRFs that models the dynamic scene as a time-variant continuous function of appearance, geometry, and 3D scene motion and captures 3D scene dynamics effectively, allowing an effective space-time view synthesis [4]. A temporal interpolation approach, which extracts features from space-time inputs and interpolates them across time frames allows capturing of short-term and long-term temporal features, achieving state-of-the-art results in both rendering quality and training speed [8]. However, these methods either fail in cases with large deformations between temporally consecutive input images [9], have prohibitively high training and rendering times [4], or are non-generalizable to dynamic regions that are not observed in the training sequence [8]. NeRF-W is able to deal with large deformations, such as photometric and environmental variations, across an input photo collection as it optimizes an appearance embedding for each input image, allowing it to maintain consistency across different images and contributing to a greater temporal coherence.



Figure 2: Sample Images in Dataset

2.1.3 Static-Dynamic Scene Decomposition

Static-dynamic scene decomposition is an increasingly popular approach to extend NeRFs to handle dynamic scenes. By decomposing a scene into its static and dynamic components, NeRFs can better model each part’s unique characteristics, allowing high-fidelity reconstruction without being influenced by the movement and variability of dynamic objects. EmerNeRF introduces a self-supervised method for stratifying scenes into static and dynamic fields, which is then used to parametrize an induced flow field—the coupling of these three fields enables the quality representation of highly-dynamic scenes [16]. D2NeRF achieves state-of-the-art in decoupling dynamic and static 3D objects and image segmentation for moving objects by representing the moving objects and the static background by two separate neural radiance fields, with only one allowing for temporal changes, in addition to a shadow field network to detect and decouple dynamically moving shadows [14]. In general, as a result of ground truth annotations for segmentation being expensive, many static-dynamic scene decomposition methods in NeRFs rely on self-supervised methods [6, 7]. However, a common limitation of these approaches is the difficulty in successfully reconstructing the background when a moving object dominates the image and obscures different perspectives, especially in areas with insufficient observations or frequent occlusions. Our approach attempts to overcome these challenges by utilizing a NeRF framework which explicitly removes dynamic objects in the scene, combined with occlusion inpainting to use static-dynamic scene decomposition with more accurate reconstruction results.

3. Dataset

3.1. Data Collection

4. Evaluation

Because our focus was on a scene containing dynamic objects, we chose a central pillar in the main quad, a relatively busy area, for our data collection. Here, we collected 85 images by periodically circling the pillar over a span of 5



Figure 3: COLMAP Pose Estimation Visualized

minutes. Over the 85 images, there include a range of moving objects, such as pedestrians standing in the background or bicyclists biking by. As we did not take a frame-by-frame sequence of but instead and continually moved around the pillar, it is often the case that certain background objects are only present in only one or a few images of the set, yielding to a not wholly temporally-consistent image sequence, as shown in Fig 2. This figure demonstrates a variety of scenarios: some images have no moving objects visible, some have very visible pedestrians dominating the scene, and others show partially occluded pedestrians further in the background.

4.1. COLMAP Pose Estimation

To feed our custom dataset into the NeRF pipeline, we must include the camera poses for each image. We achieve this preprocessing step by applying COLMAP [11]. COLMAP facilitates image-based 3D reconstruction by first recovering a sparse representation of the scene and the camera poses of the input images using Structure-from-Motion (SfM). SfM is the process of reconstructing 3D structure from its projections into a series of images taken from different viewpoints. The input is a set of overlapping images of the same object, while the output is a 3D reconstruction of the object, including the intrinsic and extrinsic camera parameters of all images. Typically, SfM systems divide this process into three stages: feature detection and extraction, feature matching and geometric verification, and structure and motion reconstruction. The output from SfM then serves as the input to Multi-View Stereo (MVS) to recover a dense representation of the scene. This process ensures that our dataset includes accurate camera poses necessary for the NeRF pipeline.

5. Method

5.1. Baseline

For our baseline evaluation, we remove 3 images from the dataset to serve as validation viewpoints. Initially, we train the NeRF model on the entire dataset without any preprocessing. Using COLMAP for pose estimation, this approach attempts to match all images, including those with

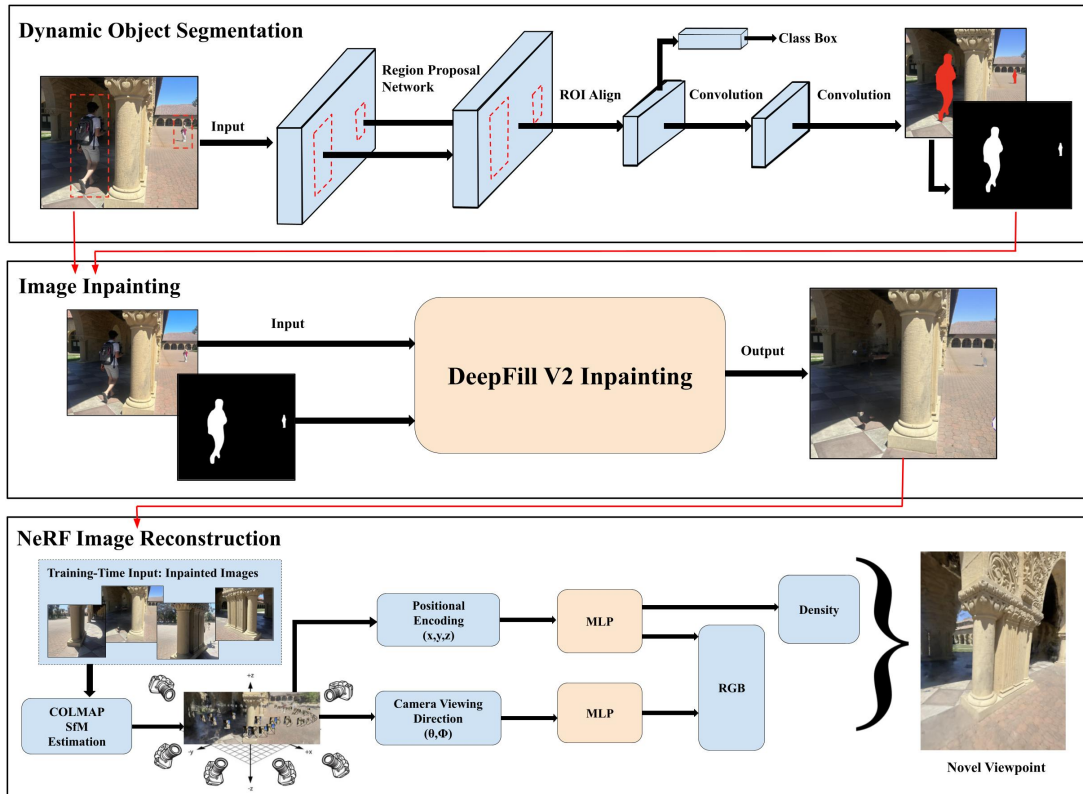


Figure 4: Flow of Network

dynamic elements, which we expect to contain numerous artifacts due to the presence of moving objects that are present in many images of the dataset. As a second baseline, we train the NeRF model exclusively on static images, which we recover by retrieving the images containing no segmented masks during our segmentation step. While COLMAP performs pose estimation on this reduced set, where because of the absence of dynamic images, which reduces the dataset by approximately 42.4% to a total of 49 images, we expect its reconstruction to be less accurate in some regions due to the diminished dataset coverage.

5.2. Model Flow

We divide the process of handling dynamic scenes in NeRFs is divided into three models, as shown in Fig. 4. First, an instance segmentation model is used to mask dynamic objects. Next, the masks along with the original input images inform the inpainting of the occluded backgrounds of the masked areas. Finally, the inpainted results are fed into a Neural Radiance Field to generate the final video rendering of the scene

5.2.1 Segmentation

In our model, the segmentation stage is crucial for accurately identifying and creating masks from dynamic objects in the input data to be used in the inpainting stage. We experimented with several semantic segmentation architectures to qualitatively determine the most effective model for our task. Initially, we evaluated U-Net and DeepLabV3 architectures, but found that they struggled with accurately segmenting more complex scenes with our dynamic objects. We ultimately found Mask R-CNN to visually demonstrate superior performance on our data. For our implementation, we utilized Mask R-CNN with a ResNet-50 backbone, as it offered a good balance between accuracy and computational efficiency. We also attempt to couple the segmentation framework with shadow-removal features with implementations as outlined in Mask-ShadowGAN [2] and WRSD [13], however were unable to fully integrate it into our pipeline.

Our pipeline uses Mask R-CNN [1] for instance segmentation, which is similar to Fast R-CNN [10] as it processes an image through a backbone convolutional neural network (CNN) to extract feature maps, which are then used to propose Regions of Interest (RoI). These regions are then pro-

Quantitative Results									
	Scene 1			Scene 2			Scene 3		
	Static	Dynamic	Inpainted	Static	Dynamic	Inpainted	Static	Dynamic	Inpainted
PSNR	12.23900	17.78892	19.05412	11.60325	17.65667	19.06477	11.61922	17.26071	19.75633
SSIM	0.430463	0.551500	0.559876	0.386409	0.490666	0.506976	0.331359	0.489227	0.494153
MSE	8747.833	7993.848	7784.771	8124.500	7859.569	7964.972	8098.355	7709.193	7877.753
MS-SSIM	0.556070	0.748474	0.792299	0.454635	0.796186	0.742656	0.433555	0.699996	0.735969
LSPIPS	0.353489	0.322726	0.374649	0.338432	0.304704	0.388221	0.319227	0.368444	0.364292

Table 1: Quantitative results for different scenes and categories

cessed to be passed through individual CNNs to classify objects in its breadth. However, Mask R-CNN also builds upon the foundation established by Faster R-CNN [10] by incorporating an additional branch specifically for predicting segmentation masks for each Region of Interest (RoI). The mask branch, which is a small fully convolutional network (FCN), is applied to each RoI to predict segmentation masks in a pixel-to-pixel manner, significantly improving the granularity of the segmentation process.

For our purposes, we leverage pretrained weights from the COCO V1 dataset for the Mask R-CNN model and pretrained weights from ImageNet for the ResNet-50 backbone. The use of COCO V1 weights allows our model to benefit from extensive training on a diverse set of images containing various objects, which improves its ability to generalize to different scenes in our dataset.

5.2.2 Occlusion Inpainting

Following segmentation, we can input the resulting binary masks into an inpainting framework, which will handle recreating aspects of the scene affected by removing the area within the masks. We use the DeepFill v2 model, with weights trained from the Places2 dataset, which deals with scene understanding and is therefore relevant to our case of scene reconstruction. DeepFill v2 is similar to its predecessor DeepFill v1, and their network architecture largely remains the same, with notable features such as a Contextual Attention layer which allows the generator to utilize information from distant spatial locations for the reconstruction of more local areas and a two-stage coarse-to-fine network structure. In this two-stage approach, the first generator network creates a coarse reconstruction, while the second generator network further refines upon the coarse image. However, DeepFill v2 takes a departure from DeepFill v1 in that it proposes Gated Convolution as a replacement to standard convolution in v1, which would improve handling of irregular masks. Specifically, in gated convolutions, the output is modulated by a gate that controls the contribution of the input features. The gate itself is another convolutional layer followed by a sigmoid activation function, which produces

values between 0 and 1 to serve as multiplicative gates. It is multiplied element-wise with another convolutional layer that can be followed by any such activation function. The equation for the output of a gated convolution can thus be expressed as:

$$y = (Wx + b) \odot (Gx + c)$$

5.3. Evaluation Method

We validate the reconstruction quality of our approach by visually assessing the novel view synthesis and providing quantitative results by comparing the reconstructed image to ground truth with Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Multi-Scale Structural Similarity Index (MS-SSIM), Mean Squared Error (MSE), and Learned Perceptual Image Patch Similarity (LPIPS) metrics to provide a robust evaluation of the reconstructed image quality from various perspectives.

6. Experiments

6.1. Quantitative

We assess the success of our framework through image similarity measures as discussed in 5.3. See Table 1 for Quantitative Results, where "Static" denotes the second baseline and "Dynamic" denotes the first baseline discussed in 5.1, while "Inpainted" denotes the processed segmentation+inpainting steps before being fed into the NeRF pipeline.

We notice that for all 3 scenes, in most of the image similarity metrics, our inpainted model outperforms both the static model and dynamic model, though by a slight margin.

6.2. Qualitative

We can also visually assess the resulting images of each preprocessing technique by checking for artifacts and other such irregularities. See Table 2 for Qualitative Results.

We notice that the static model images are very blurry in many areas, which can be attributed to the fact that we took many images away during its training because they













Qualitative Results				
	Ground Truth	Static	Dynamic	Inpainted
Scene 1				
Scene 2				
Scene 3				

Table 2: Qualitative results for different scenes and categories



Figure 5: Artifacts in Dynamic (Left) vs Inpainted (Right) Novel View Reconstruction

contained dynamic objects. Because NeRFs rely on overfitting to a specific scene, by removing even images with dynamic objects, it removes information and thus greatly lowers the novel view synthesis quality. Meanwhile, we see that Dynamic does not suffer from this blurriness, but instead contains many artifacts that are a result of the NeRF interpreting the dynamic objects in the training images as part of the scene, as shown in Fig 5. Finally, the inpainted images as a result of our framework does not suffer from this blurriness faced by the static model or many artifacts that blur the background faced by the dynamic model.

7. Conclusion

We found that segmenting+inpainting a dataset before feeding it into a NeRF model yields novel view reconstruction results that both qualitatively and quantitatively surpass results from simply feeding a dataset containing dynamic objects into a NeRF model.

To acknowledge a few limitations, it is important to note that segmentation+inpainting techniques are not yet completely seamless and often struggle with certain challenges, such as when considering an object’s shadows as well. Additionally, the quality of inpainting reconstruction can diminish when dealing with complex textures or large occluded regions, as the algorithms may not accurately predict the missing content. In future works, there is substantial room for improvement in developing more sophisticated inpainting methods and integrating better context-awareness to enhance the dynamic scene reconstruction process.

Nevertheless, adapting Neural Radiance Fields (NeRFs) to handle dynamic scenes opens up exciting applications, particularly in the field of autonomous driving. By incorporating motion masks and advanced inpainting techniques, NeRFs can effectively reconstruct high-fidelity 3D scenes from multiple viewpoints, even in the presence of moving objects as is the case in most real world scenarios. This capability is crucial for NeRFs usage in broader contexts, such as in autonomous vehicles, which require precision and reliability in reconstruction to navigate safely. As we continue

to refine these techniques, we continue to improve the accuracy and quality of dynamic scene reconstruction, and grow closer to reaching the full potential for NeRFs in many applications from virtual reality to robotics and beyond.

8. Acknowledgements

GitHub codes from the baseline papers were used for processing. Furthermore, NeRFStudio was used to create the NeRF renderings. See below for the links to the code: Mask R-CNN [1]: https://github.com/matterport/Mask_RCNN Deepfill v2 [17]: https://github.com/JiahuiYu/generative_inpainting NeRF Studio [12]: <https://github.com/nerfstudio-project/nerfstudio>

References

- [1] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn, 2018.
- [2] X. Hu, Y. Jiang, C.-W. Fu, and P.-A. Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2019.
- [3] M. M. Johari, Y. Lepoittevin, and F. Fleuret. Geonerf: Generalizing nerf with geometry priors, 2022.
- [4] Z. Li, S. Niklaus, N. Snavely, and O. Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes, 2021.
- [5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.
- [6] T.-A.-Q. Nguyen, L. Roldão, N. Piasco, M. Bennehar, and D. Tsishkou. Rodus: Robust decomposition of static and dynamic elements in urban scenes, 2024.
- [7] T. Otonari, S. Ikehata, and K. Aizawa. Entity-nerf: Detecting and removing moving entities in urban scenes, 2024.
- [8] S. Park, M. Son, S. Jang, Y. C. Ahn, J.-Y. Kim, and N. Kang. Temporal interpolation is all you need for dynamic neural radiance fields, 2023.
- [9] A. Pumarola, E. Corona, G. Pons-Moll, and F. Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes, 2020.
- [10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.
- [11] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [12] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, J. Kerr, and A. Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings, SIGGRAPH '23*. ACM, July 2023.
- [13] F.-A. Vasluianu, T. Seizinger, and R. Timofte. Wsrp: A novel benchmark for high resolution image shadow removal. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1826–1835, 2023.
- [14] T. Wu, F. Zhong, A. Tagliasacchi, F. Cole, and C. Oztireli. D²nerf: Self-supervised decoupling of dynamic and static objects from a monocular video, 2022.
- [15] H. Xu, T. Alldieck, and C. Sminchisescu. H-nerf: Neural radiance fields for rendering and temporal reconstruction of humans in motion, 2021.
- [16] J. Yang, B. Ivanovic, O. Litany, X. Weng, S. W. Kim, B. Li, T. Che, D. Xu, S. Fidler, M. Pavone, and Y. Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision, 2023.
- [17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention, 2018.