Accelerating Diffusion Sampling with Classifier-based Feature Distillation

Wujie Sun
College of Computer Science
Zhejiang University
Hangzhou, China
sunwujie@zju.edu.cn

Defang Chen
College of Computer Science
Zhejiang University
Hangzhou, China
defchern@zju.edu.cn

Can Wang†
College of Computer Science
Zhejiang University
Hangzhou, China
wcan@zju.edu.cn

Deshi Ye*
College of Computer Science
Zhejiang University
Hangzhou, China
yedeshi@zju.edu.cn

Yan Feng
College of Computer Science
Zhejiang University
Hangzhou, China
fengyan@zju.edu.cn

Chun Chen
College of Computer Science
Zhejiang University
Hangzhou, China
chenc@zju.edu.cn

Abstract—Although diffusion model has shown great potential for generating higher quality images than GANs, slow sampling speed hinders its wide application in practice. Progressive distillation is thus proposed for fast sampling by progressively aligning output images of N-step teacher sampler with N/2-step student sampler. In this paper, we argue that this distillation-based accelerating method can be further improved, especially for few-step samplers, with our proposed Classifier-based Feature Distillation (CFD). Instead of aligning output images, we distill teacher’s sharpened feature distribution into the student with a dataset-independent classifier, making the student focus on those important features to improve performance. We also introduce a dataset-oriented loss to further optimize the model. Experiments on CIFAR-10 show the superiority of our method in achieving high quality and fast sampling. Code is available at https://github.com/zju-SWJ/RCFD.

Index Terms—diffusion model, knowledge distillation, image generation, fast sampling

I. INTRODUCTION

Image generation is an important research field in computer vision and various models have been invented, such as generative adversarial networks (GANs) [1] and diffusion models [2]. The adversarial nature of GANs requires careful architecture and hyper-parameter selection to stabilize the model training, while the recent diffusion models can overcome these weaknesses and achieve better performance [3]. However, diffusion models require a greatly slower iterative sampling to get the final denoised images. Accelerating the sampling process has become critical.

Two main acceleration directions for diffusion models are training-free sampling and training schedule [2]. Training-free sampling [4]–[6] aims to propose efficient sampling methods to boost sampling speed for the pre-trained diffusion models, while training schedule [7], [8] changes the traditional ways of training to achieve high efficiency in subsequent sampling, and gives model the potential for more powerful performance.

Recently, knowledge distillation-based training schedule methods [7], [9] have exhibited strong capabilities in fast sampling and high performance, surpass other methods [4]–[6], [8] with large margins. Inspired by the idea of distilling the knowledge in a powerful teacher model into a compact student model [10], [11]. Progressive Distillation (PD) [7] asks the student sampler to mimic the teacher sampler’s two-step output with a single step. In this way, the sampler maintains a decent performance when progressively halving its sampling steps. However, little work has been done upon this.

In this paper, by using an additional classifier, we further demonstrate the power of knowledge distillation in speeding up diffusion sampling. We argue that strictly aligning the individual pixels in output images of the student and teacher samplers is difficult, especially for student samplers with few sampling steps. With the help of a classifier, we can get the high-level feature distributions based on the images output by teacher and student. By calculating the KL-divergence of these two distributions, student is able to focus on those important features (which are closely related to image composition), thus reducing the learning burden and ensuring the consistency of the image. We name it Classifier-based Feature Distillation (CFD). Notice that at this point, our classifier is NOT necessarily trained on the target dataset, since it is only used for feature extraction and does not involve category information. Therefore, our method can be applied to non-classification datasets, provided that a classifier with accurate feature extraction ability is available. Such classifier, which does not require adversarial training and pre-training on the target dataset, makes our work very different from previous works with classifiers for image generation and refinement [12], [13]. For
classifiers trained on the target dataset, we further propose **Regularized CFD** (RCFD) which combines CFD with entropy and time θ
and time t. We propose a novel classifier-based distillation method θ = t
Experiments on CIFAR-10 show that our method outper-
−
could
ELA TED WORK
Our method does not involve adversarial training, and , the sampling process is then repeated σ(1)
1 are pre-defined time-related functions,  and
≤ is
is guided by dataset-oriented loss to further improve performance, which can be used when a classifier pre-trained on the target dataset is available.

## III. RELATED WORK

### A. Diffusion Model

Diffusion model aims to sample high-quality images from random noises, which contains two processes: training and sampling. A standard training process is proposed by DDPM [14]. The well-trained network with parameter θ could take noisy image z_t and time 0 ≤ t ≤ 1 as inputs, and outputs the predicted denoised image x_t = θ(z_t, t) = θ(z_t). Starting from t = 1, the sampling process is then repeated N times to get the final generated image. Since such sampling process is very time-consuming, DDIM [4] proposes an implicit sampling to speed up, which can be represented as

\[ z_s = \alpha_s \underbrace{\theta(z_t)}_{\text{predicted denoised image } x_t} + \sigma_s \frac{z_t - \alpha_t \theta(z_t)}{\sigma_t}, \]

where α and σ are pre-defined time-related functions, z_0 is the final denoised image, and 0 ≤ s < t ≤ 1. We provide a more detailed explanation in supplementary material. Based on DDIM, Progressive Distillation (PD) [7] uses knowledge distillation to improve sampling speed. Other methods such as PNDMs [5] and DPM-Solver [6] also manage to speed up sampling, but fail to outperform PD with huge margins.

### B. Knowledge Distillation

Knowledge distillation [15] is an efficient method for model compression. Diverse knowledge such as logits [10] and intermediate features [16], [17], can be transferred from a superior teacher model to a compact student model. In addition, online knowledge distillation [18] introduces multiple training models, while self-distillation [19] contains only a single model architecture. Although knowledge distillation has a wide applications such as image classification [15], object detection [20], and semantic segmentation [21], [22], distillation for fast diffusion sampling [7] has rarely been explored yet. We believe this field holds great promise.

### C. Classifier for Image Generation

Classifier is important for image classification. Recent works show that it can also be applied to image generation [12], [13]. However, these methods need a robust classifier with adversarial training, which increases training difficulty. Classifier is also used in diffusion models to provide class-related guidance and improve performance [3]. Different from the above works, in this paper, we use the classifier to extract the feature/prediction distribution of images and transfer it to the student model as knowledge. Such classifier does not require adversarial training and can be pre-trained on a different dataset.

## III. METHODOLOGY

### A. Progressive Distillation

Progressive Distillation (PD) [7] introduces knowledge distillation to speed up sampling. Once teacher sampler with N steps is given, student sampler with N/2 steps is trained to speed up sampling. Assuming that the sampling time is now t, we can get the target denoised image x_T by sampling the teacher model for two steps. The detailed derivations for x_T are as follows:

Once we have the N-step teacher and the current time t, we can get t’ = t - 1/N and t'' = t - 2/N. z_t' and z_t'' are then calculated as

\[ z_t' = \alpha_t' \eta(z_t) + \sigma_t' \frac{z_t - \alpha_t \eta(z_t)}{\sigma_t}, \]

\[ z_t'' = \alpha_t'' \eta(z_t) + \sigma_t'' \frac{z_t - \alpha_t \eta(z_t)}{\sigma_t}, \]
\[ z_{t^*} = \alpha_{t^*} \eta(z_{t^*}) + \sigma_{t^*} \frac{z_{t^*} - \alpha_{t^*} \eta(z_{t^*})}{\sigma_{t^*}}, \]  

where \( \eta \) is the teacher model.

Assume student has denoised image \( \theta(z_t) \) and gets noisy image \( \tilde{z}_{t^*} \) in one step. If well aligned, we should have

\[ \tilde{z}_{t^*} = \tilde{z}_{t^*} = \alpha_{t^*} \theta(z_t) + \sigma_{t^*} \frac{z_t - \alpha_t \theta(z_t)}{\sigma_t}. \]

To achieve that, \( \theta(z_t) \) should be aligned with the following objective (i.e., distillation target \( x^T \)):

\[ \theta(z_t) = \frac{z_t - (\sigma_t/\sigma_t)z_t}{\sigma_t} = x^T. \]

The training loss for PD can thus be represented as

\[ L_{PD} = w_t \| x^T - \theta(z_t) \|_2^2, \]

where \( w_t = \max(\alpha_t^2/\sigma_t^2, 1) \) is used for better distillation.

Directly aligning images is very effective when the sampler has many steps, but it degrades rapidly when there are few steps. We believe that when the sampling steps are small, it becomes difficult for the student to strictly align the pixels on the image, which hinders the model learning. Therefore, we argue that in this situation, the student model should pay more attention to learning the key features associated with images, so as to improve the learning efficiency and quality.

**B. Classifier-based Feature Distillation**

A classifier \( cls \) is usually composed of two parts, feature extractor \( extr \) and fully connected layers.

Instead of aligning \( x^T \) and \( \theta(z_t) \) as PD [7], we use a classifier to extract features and use them as transferred knowledge. To be more specific, student’s output image \( x^S = \theta(z_t) \) and teacher’s derived image \( x^T \) are input to the same extractor \( extr \), and output the last features before the fully connected layers, which can be represented as

\[ F^S = extr(x^S), \quad F^T = extr(x^T). \]

After that, we convert feature into distribution using softmax function \( \sigma(\cdot) \), and calculate the KL-divergence between teacher and student feature distributions

\[ L_{CFD} = KL \left( \sigma(F^S), \sigma(F^T) \right), \]

where temperature \( 0 < \tau < 1 \) is used to sharpen the distribution. Note that \( \tau \) is only applied to teacher feature distribution to refine the distilled knowledge, which we find to be more effective than applying to both distributions. In this case, the upper limit of student performance is no longer the teacher, so in some cases (see section IV), the student can even surpass the teacher model!

KL-divergence can give large feature values greater weight in gradient descent, thus helping the model focus more on aligning these features. After the image is input into the feature extractor, the features change from the shallow fine-grained features to the deep coarse-grained features as the layer increases. Deep features contain more semantic information related to categories, which is crucial for image composition.

By aligning important teacher features, and reducing the interference of irrelevant features on model training, students with poor ability can learn more useful knowledge to generate high-quality images and improve performance.

Note that the loss in (8) is NOT oriented to a specified dataset, since we only use the feature extractor and do not include the subsequent fully connected layers for classification. Due to this advantage, our proposed distillation method can be extended to more datasets, such as CelebA and LSUN bedrooms. Next, we further introduce dataset-oriented loss to help the model better improve performance.

**Dataset-oriented loss.** For a \( N \)-step sampler, as the sampling step increases, the image obtained by \( \theta(z_t) \) tends to be clearer. A clearer denoised image in the early steps will benefit the subsequent sampling steps.

By feeding the images obtained from each sampling step into a classifier, we can calculate the entropy as follows:

\[ L_{entropy} = -\sum_{c=1}^{C} p_c \log p_c, \quad p = \sigma(cls(x^S)), \]

where \( C \) is the class number, and \( p \) denotes prediction results. Fig. 2 shows that sampling with fewer steps yields a larger entropy and generates more blurred images. This means if we minimize the entropy of prediction results, we could get relatively clearer images, especially for early sampling steps.

In addition, with the progressive distillation, it inevitably makes the current sampler’s output image distribution deviate more and more from the original optimal one. Since the dataset we used is balanced, we expect the predicted probabilities to remain equal for each class within each batch:

\[ L_{diversity} = \sum_{c=1}^{C} \tilde{p}_c \log \tilde{p}_c, \quad \tilde{p} = \frac{\sum_{b=1}^{B} p^b}{B}, \]

where \( B \) is the batch size. Combining these two losses with \( L_{CFD} \) can lead to better results, while using only the dataset-oriented loss is less effective due to the lack of teacher guidance, as shown later in the experiment.
Overall loss. The overall loss function can be represented as
\[
L_{RCFD} = L_{CFD} + \beta |\gamma L_{\text{entropy}} + (1 - \gamma)L_{\text{diversity}}|,
\]
where \(\beta\) and \(\gamma\) are hyper-parameters, and RCFD stands for Regularized Classifier-based Feature Distillation.

IV. EXPERIMENT

A. Setting

We demonstrate the superiority of our method using unconditional CIFAR-10, the most commonly used dataset on diffusion-based image generation task. We use the cosine schedule introduced in [8] to calculate \(\alpha_t\) and \(\sigma_t\). We use the U-Net [23] as the diffusion model. ResNet18 [24] and DenseNet201 [25] are used as the classifiers. The base diffusion model is trained with 1024 steps.

We compare our method with DDIM (ICLR 2021) [4], PD (ICLR 2022) [7], PNMDs (ICLR 2022) [5], and DPM-Solver (NIPS 2020) [6]. Inception Score (IS) [26] and Fréchet Inception Distance (FID) [27] of each method are reported. The distillation-based acceleration method requires iterative training to halve sampling steps. Based on results in [7] and our own experiments, we find that performance changes rapidly in distillation from 8-step to 1-step. So we focus on distillation process starting from 8-step, and distill the models using PD [7] from 1024 to 8 steps without the classifier. We re-implemented DDIM and PD for better comparison. More experiment details are provided in supplementary material.

B. Result

The comparison result is shown in Table I. As we can see, distillation-based methods (RCFD and PD) surpass other methods with large margin (4-step distillation-based samplers can achieve the performance of other samplers with 10+ steps). Also, the difference between the 8-step sampler obtained by PD and the 1024-step DDIM sampler (base diffusion model) is small, indicating the effectiveness of distillation.

In addition, RCFD with DenseNet201 achieved 6.14 (\(\pm 0.71\%\)), 2.35 (\(\pm 1.6\%\)), and 1.03 (\(\pm 21.3\%\)) FID improvement compared to PD in the 1, 2, and 4-step samplers, respectively, demonstrating its superiority. Also, with the help of the classifier, we offer the possibility for the student sampler (4-step of RCFD-DenseNet201) to significantly outperform its teacher (8-step sampler obtained from PD).

As we can see from Fig. 3, since PD only aligns images, the images can easily lack significant category information and become blurred and meaningless in the late stages of distillation. As RCFD highlights salient features and weakens useless features, it makes the generated images more discriminative and realistic, and may differ significantly from images generated by the base sampler.

C. Ablation Study

In this section, we perform ablation studies to verify the importance of each component in our method. If not specified, we use ResNet18 as the classifier and use the 8-step sampler trained by PD as the teacher to train a 4-step student.

### Table I

| Sampling Steps | Method                  | IS ↑ | FID ↓  |
|---------------|------------------------|------|--------|
| 1             | RCFD-DenseNet201        | 8.87 | 8.92   |
|               | RCFD-ResNet18           | 8.56 | 12.03  |
|               | PD [7]                  | 7.88 | 15.06  |
| 2             | RCFD-DenseNet201        | 9.19 | 5.97   |
|               | RCFD-ResNet18           | 9.09 | 6.12   |
|               | PD [7]                  | 8.70 | 7.42   |
| 4             | RCFD-DenseNet201        | 9.34 | 3.80   |
|               | RCFD-ResNet18           | 9.24 | 4.24   |
|               | PD [7]                  | 9.04 | 4.83   |
| 8             | PD [7]                  | 9.14 | 4.14   |
|               | DDIM [4]                | 8.14 | 20.97  |
| 10            | PNMDs [5]               | -    | 7.05   |
| 12            | DPM-Solver [6]          | -    | 4.85   |
| 1024          | DDIM [4]                | 9.21 | 3.78   |

### Table II

**Impact of different pre-trained classifiers on performance.**

**We only use \(L_{CFD}\) for fair comparison.** For PD, the IS is 9.04, and the FID is 4.83. The teacher has IS 9.14 and FID 4.14.

| Pre-trained Dataset | Classifier | IS ↑ | FID ↓ |
|---------------------|------------|------|-------|
| CIFAR-10            | ResNet18   | 9.14 | 4.42  |
|                     | ResNet50   | 9.16 | 4.24  |
|                     | DenseNet201| 9.34 | 3.80  |

### Table III

**Performance comparison with SOTA methods on CIFAR-10. Higher IS and lower FID are better. For longer distillation iterations, RCFD achieves FID 7.87 with only 1 step, while PD only achieves 12.07 (shown in supplementary material).**

#### Ablation study on classifier.
In this section, we try different classifiers and see how the performance changes. As shown in Table II, no matter what classifier we use, we can achieve better results than PD. However, if possible, it is better to train the classifier on the target image generation dataset.

In addition, as the classifiers become more and more powerful, they also help the student samplers produce higher quality images, which even achieve significantly better FID than the teacher. We believe that for a more powerful classifier, it will extract more accurate and meaningful features, therefore, it provides students with more effective knowledge for distillation, thus helping students produce better images.

#### Ablation study on each loss.
Three losses are included in our method, \(L_{CFD}\), \(L_{\text{entropy}}\), and \(L_{\text{diversity}}\). \(L_{CFD}\) is a dataset-independent loss, which introduces classifier-based distillation to align student’s feature distribution with teacher’s sharpened feature distribution. The latter two are dataset-oriented losses, where \(L_{\text{entropy}}\) is used to generate clearer images and \(L_{\text{diversity}}\) maintains the class balance.

As we can see from Table III, with only \(L_{CFD}\), we can already achieve better performance than PD. Although good results cannot be achieved using \(L_{\text{entropy}}\) and \(L_{\text{diversity}}\) when \(L_{CFD}\) is not available, optimal performance can be achieved by combining all three terms. The reason is that the teacher constraint \((L_{CFD})\) will prevent the generated images from being too abstract and meaningless during training (as shown in Fig. 4), which improves the model performance.

#### Ablation study on distilled distribution.
Distilling the feature
distribution is dataset-independent and yields better results than distilling the prediction distribution, as shown in Fig. 5. It can be seen that aligning student’s feature distribution with teacher’s slightly sharpened feature distribution (temperature $0.9 \leq \tau < 1$) obtains better results than PD and always outperforms distilling the prediction distribution.

For the feature distribution, over large temperature will make the teacher’s feature distribution tend to be uniformly distributed, hindering the learning of important features and making the image meaningless, while over small temperature makes few features to be highlighted, making the image too abstract and causing performance degradation.

For the prediction distribution, since it has smaller constraints compared to the feature distribution (i.e., different feature distributions may yield the same prediction results), the learning of image’s details can be weakened, which leads to bad performance.

**Ablation study on dual softmax temperature.** In our method, softmax temperature $\tau$ is only used for the teacher, as
Fig. 6. Impact of dual temperature and distilled distributions on performance. The FIDs of PD and CFD are shown by the blue dotted line and red dashed line, respectively. For better comparison, we use $L_{\text{CFD,dual}}$ only.

shown in (8). We now apply the same temperature $\tau$ to both the student and the teacher, and change the loss as

$$L_{\text{CFD,dual}} = \sigma^T \frac{\text{KL}(\sigma^T(F^S), \sigma^T(F^T))}{\sigma^T(F^T)}.$$  \hspace{1cm} (12)

Fig. 6 shows that, for a wide range of temperatures, aligning feature distributions and prediction distributions achieves better performance than PD, but fails to outperform the original $L_{\text{CFD}}$ which only uses temperature for the teacher.

Although large dual temperature helps to improve performance, we believe that such aligning (no matter it is feature or prediction distribution) determines that the upper limit of the student is the teacher (unlike traditional knowledge distillation for image classification, there is no additional guidance such as labels during distillation), which limits performance improvement.

V. CONCLUSION

In this paper, we propose a classifier-based distillation method to speed up the sampling of diffusion models. We let student align its feature distribution with teacher’s sharpened feature distribution, rather than aligning the generated images. In this way, student can focus on learning important features that make up an image, resulting in even better performance than the teacher. This distillation method is also applicable when the classifier is pre-trained on other datasets. When the classifier pre-trained on the target dataset is available, we propose a dataset-oriented loss to further improve performance. Experiments on CIFAR-10 show the superiority of our method.

REFERENCES

[1] D. Saxena and J. Cao, “Generative adversarial networks (gans) challenges, solutions, and future directions,” ACM Computing Surveys (CSUR), vol. 54, no. 3, pp. 1–42, 2021.
[2] H. Cao, C. Tan, Z. Gao, G. Chen, P.-A. Heng, and S. Z. Li, “A survey on generative diffusion model,” arXiv preprint arXiv:2209.02546, 2022.
[3] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” Advances in Neural Information Processing Systems, vol. 34, pp. 8780–8794, 2021.
[4] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in International Conference on Learning Representations, 2020.
[5] L. Liu, Y. Ren, Z. Lin, and Z. Zhao, “Pseudo numerical methods for diffusion models on manifolds,” in International Conference on Learning Representations, 2021.
[6] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “Dpm-solver: a fast ode solver for diffusion probabilistic model sampling in around 10 steps,” arXiv preprint arXiv:2206.00927, 2022.
[7] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” in International Conference on Learning Representations, 2021.
[8] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in International Conference on Machine Learning. PMLR, 2021, pp. 8162–8171.
[9] E. Luhman and T. Luhman, “Knowledge distillation in iterative generative models for improved sampling speed,” arXiv preprint arXiv:2101.02388, 2021.
[10] G. Hinton, O. Vinyals, J. Dean et al., “Distilling the knowledge in a neural network,” arXiv preprint arXiv:1503.02531, vol. 2, no. 7, 2015.
[11] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, “Knowledge distillation with the reused teacher classifier,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11,933–11,942.
[12] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, “Image synthesis with a single (robust) classifier,” Advances in Neural Information Processing Systems, vol. 32, 2019.
[13] R. Ganz and M. Elad, “Bigroc: Boosting image generation via a robust classifier,” arXiv preprint arXiv:2108.03792, 2021.
[14] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Advances in Neural Information Processing Systems, vol. 33, pp. 6840–6851, 2020.
[15] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” International Journal of Computer Vision, vol. 129, pp. 1789–1819, 2021.
[16] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “Fitnets: Hints for thin deep nets,” arXiv preprint arXiv:1412.6550, 2014.
[17] D. Chen, J.-P. Mei, Z. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, “Cross-layer distillation with semantic calibration,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, 2021, pp. 7028–7036.
[18] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, “Online knowledge distillation with diverse peers,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, 2020, pp. 3430–3437.
[19] T. Furlanello, Z. Lipton, M. Tschantzen, L. Itti, and A. Anandkumar, “Born again neural networks,” in International Conference on Machine Learning. PMLR, 2018, pp. 1607–1616.
[20] Z. Li, P. Xu, X. Chang, L. Yang, Y. Zhang, L. Yao, and X. Chen, “When object detection meets knowledge distillation: A survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.
[21] C. Sha, Y. Liu, J. Gao, Z. Yan, and C. Shen, “Channel-wise knowledge distillation for dense prediction,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 5311–5320.
[22] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, “Cross-image relational knowledge distillation for semantic segmentation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12,319–12,328.
[23] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[25] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” Advances in neural information processing systems, vol. 29, 2016.
[27] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” Advances in neural information processing systems, vol. 30, 2017.