DICE - Denoising Cascade: An Attention-Free Diffusion Architecture

Anonymous ICCV submission

Paper ID 12825

Figure 1. 4 pairs comparing between DiT (left of pair) and DICE (right of pair)

Abstract

*Vision transformers have played a vital role in the development of high fidelity image generation, which has accumulated into the development of DiT. While this paper was surmountable in accelerating the diffusion methodology, its largest drawback has been it's immense computational demands to train and inference. We propose a novel attention-free alternative architecture, DICE (**D**enoising **C**ascad**e**), to address these shortcomings through the use of multi-resolution stages inspired by Diffusion model with the Multi-Resolution network (DiMR) in combination with intermittent denoising layers by Differential Transformers. By prioritizing parameter efficiency and lower memory utilization, we hope to show the feasibility of our solution in providing an accessible alternative to image diffusion within a compute constrained environment.*

1. Introduction

Image generation has exponentially evolved over the last two decades [16] [8] [6] [28] [25], resulting in improvements to high fidelity images that rival against real samples [1]. This is no doubt as a direct result of "Attention is all you need" [36], which later evolved to bring global contexts in the vision domain [7].

However, to combat Attention's limitations in vision tasks, the development of SWIN Transformers was introduced, which promoted a hierarchical structure [24] to alleviate fundamental scaling issues with Attention [15]. Ironically, this new structure resembled a CNN [43], which propelled the success of ConvNeXt which was able to achieve higher performance by focusing on key structural improvements such as reverse bottlenecks and more modern activation functions [25] that were becoming prevalent in the attention domain.

Consequently, the field of 'attention-alternatives' that embrace this new approach (such as RWKV [29], Mamba [10], and even updated forms of RNNS [27]) can exceed the performance of Attention without the use of a global context in text and even vision tasks [44] [25]. While these methods have limitations in their own right, they all reinforce the same prioritization of parameter and compute efficiency that can be achieved when diverting away from Attention.

As such, our novel architecture does not utilize Attention in an attempt to aim for more efficient results for high

resolution image synthesis. Instead, we base our approach from the principals learned from ConvNeXt and focus on taking mechanisms found in Attention-based solutions and instead focus on translating them in a purely CNN based architecture.

Our overall structure is originally based from Cascade Diffusion [13] where we replace the Attention layers with ConvNeXt, similar to DiMR's approach [22]. However, we significantly changed the residual connections of the model to leverage noise predictions that are made within the intermediate layers, which we show helps improve overall FID and Inception metrics. In addition to the architectural changes, we modify the ConvNeXt blocks themselves to include 'signal prioritization' technique introduced by Differential Transformers [41]. While this technique was intended for the text domain, we show that it is able to maintain its viability in the vision space by significantly improving convergence and reducing training times. In aggregate, these two techniques allow us to avoid the use of Attention. As a result, we also avoid the handling of the 'patchification process' which creates the dilemma of having to balance compute complexity against image fidelity due to the quadratic scaling of Attention mechanisms [36].

We provide preliminary results to supports our developed solution, and advocate for future research into it's viability in creating similarly efficient solutions in the vision space more broadly, namely for classification, segmentation, and video generation.

2. Related Work

Attention Mechanism Attention [36] is a widely used mechanism that was initially utilized for natural language processing (NPL) tasks by enabling a global context while also efficiently parallelizing on the GPU. Despite its text based origins, it has driven multiple advancements for vision tasks [7] [24] [8] and a wide variety of other domains [40] [39] [3]

Diffusion Models By iteratively predict noise patterns within a given input, Diffusion models [35] [12] have shown to be incredibly useful in a diverse range of tasks. However, Latent Diffusion Models (LDM) [30] are more commonly employed [20] [17] [42] to reduce computational complexity and training stability by operating within a pre-trained latent space which generally helps improve overall visual fidelity.

DiT DiT is a latent diffusion model that took an alternative approach to the common U-Net [31] structure by instead employing a conventional vision transformer [7]. They also showed that employing smaller patches within the 'patchification' pre-processing stage, model performance improves drastically at the cost of higher compute requirements, which aligns with the conclusions of other research papers [34] [24].

Cascade diffusion Cascade Diffusion [13] is a technique where the task of noise prediction is distributed across multiple stages, which each outputs its own unique sample that is then passed onto the higher resolution stages. This semi-super-resolution technique allows for SOTA Fréchet Inception Distance (FID) scores of "4.88 at 256x256 resolutions" for ImageNet by more efficiently utilizing model parameters.

They primarily support their findings with empirical reasoning that the majority of sample quality is derived from the lower stages. This concept shares striking resemblance to FastGan [21], which similarly prioritizes compute in the lower resolution spaces to achieve high parameter efficiency.

For our proposal, we strongly believe that such research is a great approach to creating an efficient solution when used in combination with several other advancements that have been made post publication to Cascade Diffusion.

DiMR [22] extends upon the work of DiT by utilizing ConvNeXt blocks within a Cascade-style architecture for the high resolution synthesis stages to reduce computational complexity. As a result, the paper was able to achieve SOTA FID of 1.70 for 256x256 ImageNet. They also advocated for a simplified conditioning mechanism, which only utilizes two parameters per layer, which aided in reducing the overall parameter count.

Overall we believe this work can benefit us in achieving our aims by providing evidence into the feasibility of cascade diffusion in combination with attention alternatives (ConvNeXt in this case).

Differential Transformer Differential Transformer [41] argues that a "transformer often over allocates attention to irrelevant contexts" which appears as "noise" in the attention maps. This makes the model over-prioritise irrelevant relationships, amplifying hallucinations and preventing effective long-context modeling.

Their solution was to use the difference between two attention maps to amplify the common signal that both maps agreed upon. This resulted in a 65% reduction in parameter count for the same overall performance. To generate these two signals that can be used to identify relevant relationships, they modify the attention mechanism by duplicating the Query (Q) and Key (K) components. The duplicated signal can then simply be subtracted to find any similarities between them. This is only possible as the query/key values

undergo softmax transformations, which convert them into relationship probabilities.

While Differential Transformers were created within the text domain, we believe that by converting these concepts into the vision field we can yield similar improvements.

CovnNeXt This paper [25] 'tests the limits of what a pure ConvNet can achieve' by applying iterative improvements to conventional convolutional neural networks [9] that have been identified in the vision transformer space. Their main findings conclude that a large 7x7 kernel, in combination with an inverted bottleneck, GELU activation functions, skip connections to a multilayer perceptron (MLP), and a normalization layer can in fact outperform the previous SOTA SWIN Transformer [24]. While this method does not intrinsically have a method of providing the same global attention like a transformer based solution does, other solutions [37] [14] show how this mechanism can be substituted by manipulating channel information. We rely on the principals that ConvNeXt presents to validate that translating Differential Transformers into the vision domain is a viable solution.

3. Proposed Method

3.1. Denoising Stages

Typical cascading solutions take the output of each stage and then adds them to the inputs of the upper stages as a form of a residual connection. However, in the case of cascade this may bring several drawbacks as the residual must be upsampled as a fundamental result of distributing the workload among multiple resolutions.

To illustrate this concern, we simulate the upsampling of the latent space within Stable Diffusion [30] using linear interpolation. As shown from Figures 2 to 4, we argue that this upsampling of the residual introduces artifacting due to critical depth information being stretched across the width/height dimensions, resulting in color and structural noise.

Figure 2. 1x residual

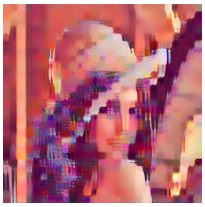

Figure 3. 2x residual

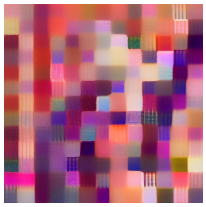

Figure 4. 4x residual

Instead, our proposed architecture avoids this issue entirely by treating the output of each stage as a direct noise prediction, which can be more succinctly upsampled to remove noise in the inputs of the upper stages.

This also has a dual benefit over the conventional method as it not only allows for noise variation predictions which has been shown to improve training stability [18], but also avoids additional loss functions that would otherwise interfere with the residual process.

3.2. ConvNeXt 'hands'

This novel approach follows in the footsteps of Differential Transformers which attempts to mitigate noise by "canceling" opposing attention signals [41]. This results in allowing the model to prioritize the most important relationships within the text, rather than attempting to maintain an equally valued global state.

In a similar vein, the DICE architecture applies this technique within ConvNeXt to mimic this concept of token prioritization. The primary roadblock to this approach is that it is not directly translatable for the vision domain as convolutional layers output feature maps rather than probabilities. For this reason, we use the following pragmatic function to in an attempt to deliver a similar effect:

$$f(x_{\text{left}}, x_{\text{right}}) = \frac{x_{\text{left}} + x_{\text{right}}}{1 + e^{(x_{\text{left}} - x_{\text{right}})^2 \cdot \lambda}}$$

Figure 5. Signal extraction function

This provides the same "denoising" effect as proposed by differential transformer while simultaneously accounting for the negative components of a feature map. This provided us the opportunity to also include a learnable parameter, λ, which controls the signal difference penalty to provide the model flexibility in signal selectivity, allowing it to 'amplify' or reduce any inherent noise as it sees fit.

Figure 6. Left Features

Figure 7. Right Features

Figure 8. Denoised Features

To illustrate the denoising effect, Figures 6 to 7 simulate an example left/right feature map (normally sampled), and the resulting refined features ($\lambda = 1$). As shown, the function promotes matching features, and subdues dissimilar ones successfully.

Figure 9 visualizes this left/right signal selectivity, where we can observe matching signals forming a hill for positive signals, and a trough for negative signals. Alternatively, the function flattens in regions where the signal difference is

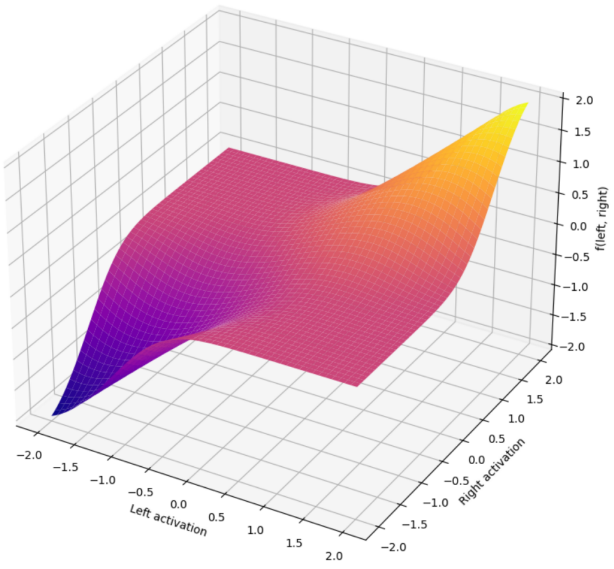

Figure 9. **Signal Extraction Function** Comparing the output of a single feature given the left &right outputs from a convolutional layer

larger. We can also imagine how as the λ parameter increases, the gradient along the function becomes steeper, and as such only retains the closest matching signals.

This figure ultimately illustrates how the continuous function meets the intended effect that Differential Transformers strives for, without the reliance on Attention or on it's softmax activation functions.

3.3. DICE block design

Cascade structure. We distribute the model weights within multiple hierarchies (1x, 0.5x, 0.25x) which enables the majority of the model processing to occur within the lower resolutions at a much higher channel dimension thus more efficiently, as similarly demonstrated by Cascade Diffusion and DiMR.

Modified ConvNeXt Blocks. As outlined in section 3.2, we modify the ConvNeXt Block by duplicating the 7x7 convolutional layer to extract two candidate features from the input, as shown within figure 10. This can then be used in combination with our proposed function to compute the extracted signal.

AdaLN-Zero Conditioning While DiMR promotes a simplified version of conditioning (Time-Dependent Layer Normalization), we instead utilize DiT's conventional AdaLN-Zero method as it has more research to validate its results [28] [4].

Hyperparameters For the specific structure of our model, we use a 'base' channel size of 1152, which multiplies by 1x, 2x, and 3x for the first, second, and third stage

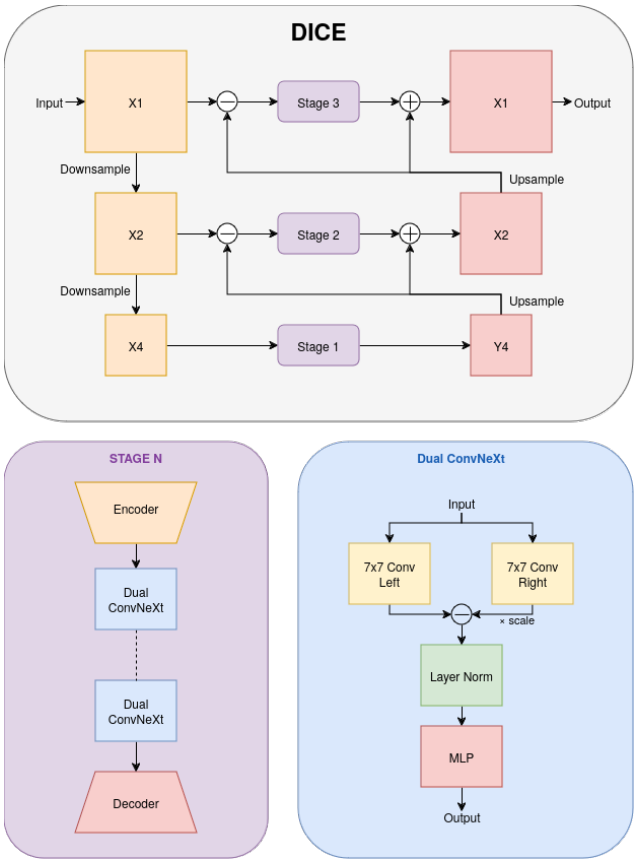

Figure 10. **Overall DICE Architecture**: We modify the ConvNeXt block to include the signal prioritization technique, then chain these blocks within a single stage. Finally, the stages are arranged to denoise each hierarchical representation of the input.

respectively. Similarly, the number of layers follows 6x, 2x and 1x. This was loosely inspired by DiMR-Xl/2R while also attempting to fit within the compute availability minimizing the number of parameters.

Weight Initialization We initialize the convolutional and fully connected layers by sampling from a normal distribution with a standard deviation of 0.02, following the original ConvNeXt implementation.

4. Experiments

We would like to pre-face our methodology by explaining that our compute capabilities are drastically limited in comparison to the likes of DiT or DiMR. To fully alleviate the potential for any compute bias that the official implementations may have, we retain them in the same compute environment as DICE. The primary drawback of our methodology in this instance is that we will not be able to directly compare against fully converged SOTA models, where we instead draw our conclusions from the sole differences in

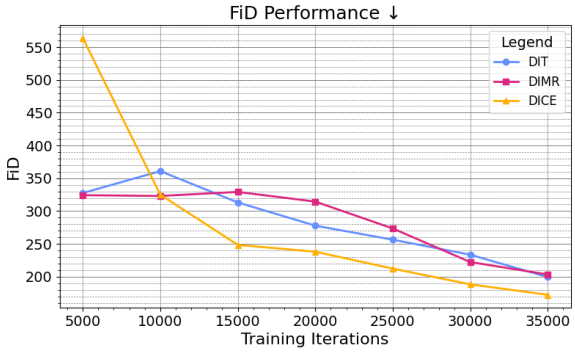

Figure 11. **FID Performance** of DiT, DiMR, & DICE. We show improvements over the baseline, despite initial underperformance at the 5000 iteration mark

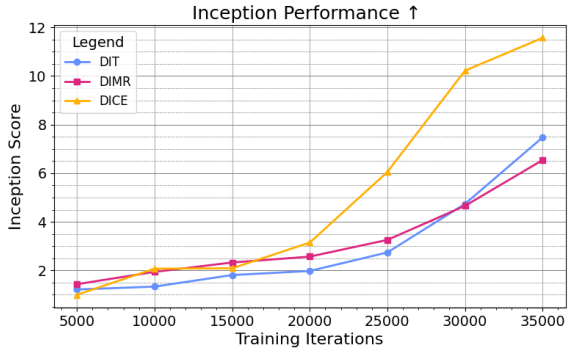

Figure 12. **Inception Performance** of DiT, DiMR, & DICE. Shows faster convergence with DICE after 20000 iterations of training

architecture.

Baseline models hyperparameters For our experiment, we limited ourselves on training DiT & DiMR as our baseline models, alongside our developed DICE architecture. These models were selected as they inspired the DICE architecture, while also providing the ability for us to compare against existing SOTA solutions. More specifically, we use the 'flagship' model that DiT & DiMR presented, notably DiT-XL/2 & DiMR-Xl/2R respectively.

4.1. Experimental Setup

Compute Environment We conducted all experiments on a single NVIDIA RTX 3090 24 GB GPU. This included training DiT, DiMR, and DICE from randomly initialized weights for the same number of training samples & batch size. Because the compute ceiling can be considered our control variable, we believe this ensures the validity of our results.

Dataset For our image dataset, we used ImageNet-1k [5] for class-conditional latent diffusion image generation as it was used in DiT and DiMR due to its research prevalence. This dataset contains 1,281,167 labeled 224x224 images that were preprocessed with a bicubic 256x256 resize and a mean/std normalization following the work of ViT [7].

Training Setup Our experiment was developed using the PyTorch library and trained each model for 35 thousand steps (4 epochs). We utilized a batch size of 128, an image size of 256x256, and a fixed learning rate of 1×10^{-4}. To reduce compute requirements, we trained at half precision mode and employed checkpointing. The only augmentation that was used was horizontal flips.

Following best practices, we employed an exponential moving average model (EMA) of each model's weights with a decay rate of 0.9999, which where used for calculating performance metrics on the validation set.

Each model is trained with the same dataset and with $seed = 42$. This results in each model being trained with the same latent samples, labels, and timestamps throughout the entire training run.

Computing Results After training DiT, DiMR, and DICE, we use ADM's TensorFlow evaluation suit [6] to compute FID [11], SFID [26], Inception Score [32], and Precision/Recall [19] every 5000 training steps. Following in the footsteps of DiT, we generate samples using 250 DDPM steps without performing classifier-free guidance.

One major limitation we encountered was the compute required to fully evaluate each model. For a comparable standard, FID requires 50 thousand samples, however, due to our compute limitations, we were limited to only 500. To remedy this issue, we used the same seed across each model to generate comparable sets. As a result, our scores are not compatible with other research papers, but can still be used to validate relative performance.

We also computed a 'static analysis' of each model, which calculates the number of trainable parameters, floating point operations (flops), multiply-accumulate operation (multi-adds), and number of images per second to compute a sample with 250 DDPM steps. This will later be used in combination with the other metrics to determine which model is the most effective.

4.2. Experimental Results

Model	Inception ↑	FID ↓	SFID ↓	Precision ↑	Recall ↑
DiT	7.47	199.35	381.14	0.124	0.15
DiMR	6.54	203.31	323.94	0.188	0.05
DICE	**11.56**	**169.09**	**312.55**	**0.218**	**0.27**

Table 1. **Performance metrics** of baseline models vs DICE

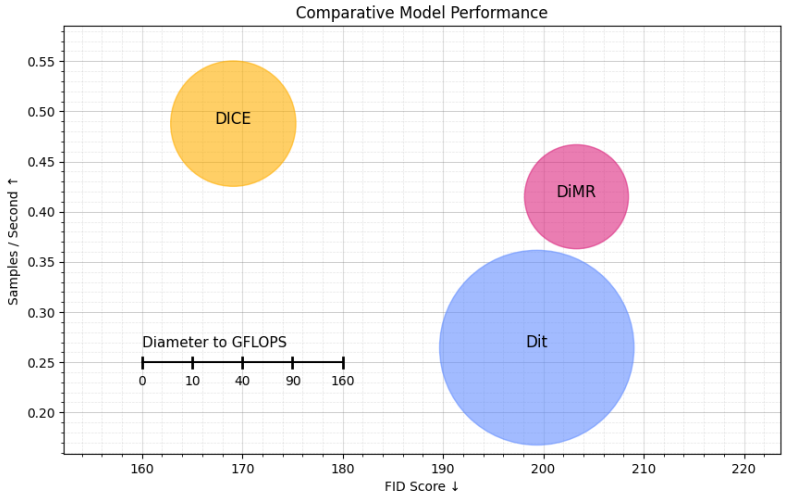

Figure 13. **Bubble graph** of DiT/DiMR/DICE performance, bubble area indicates the number of GFlops in the model. While the best performing model, DICE, inferences the fastest, it has a higher flop count indicating higher compute useage.

As shown in table 1, compared to the baseline models, our architecture is able to outperform them on all available metrics. We achieve a greater than 50% improvement to the inception score, and final FID of 169. Obviously, these statistics pale in comparison to the SOTA, but as demonstrated shows a successful relative improvement.

4.3. Static Analysis

Model	Parameters ↓	Mult-Adds ↓	GFlops ↓	Samples/Sec ↑
DiT	674,834,720	**0.68**	114.46	0.2645
DiMR	**107,452,688**	1.31	**32.68**	0.4149
DICE	178,425,240	1.05	47.49	**0.4878**

Table 2. **Static metrics** of baseline models vs DICE

When performing static analysis, the baseline alternative models generally achieve a lower number of operations needed to perform an inference. However, despite this, the lack of the attention mechanism allows DICE to still perform inferences at a higher samples/second rate. This is due to the reduced complexity of CNNs allowing for higher throughput on the GPU during inferencing [25].

5. Discussion

While it is fortunate that the models tested mostly utilize the memory provided, they do so with vastly different numbers of parameters and floating / multiply-accumulate operations. This can be attributed to the low training resolution (providing a theoretical advantage to the low compute requirements of attention [36] [33] [38]) in combination with the model architectures that are diverse and handle intermediate latent maps in vastly unique ways.

As a consequence, for our overall analysis in Figure 13, we prioritize throughput, Gflops, and FID as they are the most practical metrics in the performance of these models.

Figure 13 in combination with Figures 11 and 12 shows the ability for DICE to converge faster and in general outperform the baseline models without a significant advantage in compute / memory requirements.

6. Discussion

While it is fortunate that the models tested mostly utilize the memory provided, they do so with vastly different numbers of parameters and floating / multiply-accumulate operations. This can be attributed to the low training resolution (providing a theoretical advantage to the low compute requirements of attention [36] [33] [38]) in combination with the model architectures that are diverse and handle intermediate latent maps in vastly unique ways.

As a consequence, for our overall analysis in Figure 13, we prioritize throughput, Gflops, and FID as they are the most practical metrics in the performance of these models.

Figure 13 in combination with Figures 11 and 12 shows the ability for DICE to converge faster and in general outperform the baseline models without a significant advantage in compute / memory requirements.

7. Limitations & Future work

Throughout our research, we had to make multiple compromises on the brevity of our investigation due to compute limitations. While we felt that we mitigated the most of-

fending issues, unfortunately we were only able to provide relative results within our environment. For future work, we more than promote training DICE with the same level of compute as DiT / DiMR to produce metrics that can better validate the successfulness of our method.

8. Conclusion

We introduced Denoising Cascade (DICE) as a promising attention-free alternative that achieves better results when compared against our baseline models despite being placed in the same limited compute environment. We hope this inspires future research into the viability of such similar methods, and possibly in other vision domains such as video generation.

8.1. Ethical & Social Impact

As part of studies, we conduced a brief self-reflection on the ethical and social impacts of our work.

Emissions can be estimated from the approximately 200 GPU hours utilized in the testing, training, and validation phases of our work. This equates to an equivalent 39.2 kg of CO2 emissions, which were offset by using solar energy.

Abusive use of image generation (such as for achieving misinformation) is a growing concern [23] [2]. However, we believe that our solution has broader implications for the vision domain rather than it's current practical accomplishments, but is still a valid concern the context of future work.

References

[1] Sifat Muhammad Abdullah, Aravind Cheruvu, Shravya Kanchi, Taejoong Chung, Peng Gao, Murtuza Jadliwala, and Bimal Viswanath. An analysis of recent advances in deepfake image detection in an evolving threat landscape, 2024. 1

[2] Yuki M. Asano, Christian Rupprecht, Andrew Zisserman, and Andrea Vedaldi. Pass: An imagenet replacement for self-supervised pretraining without humans, 2021. 7

[3] N. Bouatta, P. Sorger, and M. AlQuraishi. Protein structure prediction by alphafold2: are attention and symmetries all you need? *Acta Crystallogr D Struct Biol*, 77(Pt 8):982–991, 2021. 2

[4] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation, 2024. 4

[5] 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, 2009. 5

[6] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. 1, 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 2, 5

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. 1, 2

[9] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):826–834, 1983. 3

[10] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024. 1

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 5

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851. Curran Associates, Inc., 2020. 2

[13] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation, 2021. 2

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2019. 3

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. 1

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 1

[17] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Pagoda: Progressive growing of a one-step generator from a low-resolution diffusion teacher, 2024. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 3

[19] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019. 5

[20] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. 2

[21] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis, 2021. 2

[22] Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models, 2024. 2

[23] Xuannan Liu, Zekun Li, Peipei Li, Shuhan Xia, Xing Cui, Linzhi Huang, Huaibo Huang, Weihong Deng, and Zhaofeng He. Mmfakebench: A mixed-source multimodal misinformation detection benchmark for lvlms, 2024. 7

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 1, 2, 3

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s, 2022. 1, 3, 6

[26] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W. Battaglia. Generating images with sparse representations, 2021. 5

[27] Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences, 2023. 1

[28] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023. 1, 4

[29] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand

Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023. 1

[30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 2, 3

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[32] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 5

[33] Xuyang Shen, Dong Li, Ruitao Leng, Zhen Qin, Weigao Sun, and Yiran Zhong. Scaling laws for linear complexity language models, 2024. 6

[34] Yuyang Shu and Michael E. Bain. Retina vision transformer (retinavit): Introducing scaled patches into vision transformers, 2024. 2

[35] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2256–2265, Lille, France, 2015. PMLR. 2

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1, 2, 6

[37] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module, 2018. 3

[38] Yu-Huan Wu, Shi-Chen Zhang, Yun Liu, Le Zhang, Xin Zhan, Daquan Zhou, Jiashi Feng, Ming-Ming Cheng, and Liangli Zhen. Low-resolution self-attention for semantic segmentation, 2025. 6

[39] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models, 2023. 2

[40] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: a geometric diffusion model for molecular conformation generation, 2022. 2

[41] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer, 2024. 2, 3

[42] Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation, 2024. 2

[43] Chong Zhou, Chen Change Loy, and Bo Dai. Interpret vision transformers as convnets with dynamic convolutions, 2023. 1

[44] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024. 1