SNED: Superposition Network Architecture Search for Efficient Video Diffusion Model

Zhengang Li[1,2], Yan Kang[2], Yuchen Liu[2], Difan Liu[2], Tobias Hinz[3], Feng Liu[2], Yanzhi Wang[1]
[1]Northeastern University, [2]Adobe Research, [3]Adobe
[1]{li.zhen, yanz.wang}@northeastern.edu, [2,3]{yankang, yuliu, diliu, thinz, fengl}@adobe.com

Abstract

While AI-generated content has garnered significant attention, achieving photo-realistic video synthesis remains a formidable challenge. Despite the promising advances in diffusion models for video generation quality, the complex model architecture and substantial computational demands for both training and inference create a significant gap between these models and real-world applications. This paper presents SNED, a ***s****uperposition* ***n****etwork architecture search method for* ***e****fficient video* ***d****iffusion model. Our method employs a supernet training paradigm that targets various model cost and resolution options using a weight-sharing method. Moreover, we propose the supernet training sampling warm-up for fast training optimization. To showcase the flexibility of our method, we conduct experiments involving both pixel-space and latent-space video diffusion models. The results demonstrate that our framework consistently produces comparable results across different model options with high efficiency. According to the experiment for the pixel-space video diffusion model, we can achieve consistent video generation results simultaneously across 64×64 to 256×256 resolutions with a large range of model sizes from 640M to 1.6B number of parameters for pixel-space video diffusion models.*

1. Introduction

Generative modeling for video synthesis has made tremendous progress based on approaches, including GANs [21, 22, 24, 28, 30, 34, 40], autoregressive models [6, 37], VAEs [9, 35], and normalizing flows [15]. Among them, GANs have demonstrated remarkable success by extending the image-based generation to video generation with dedicated temporal designs. However, GANs encounter challenges such as mode collapse and training instability, making it difficult to scale them up for handling complex and diverse video distributions.

To overcome this challenge, diffusion models [3, 11, 26] have been studied, which establish a weighted variational bound for optimization by connecting Langevin dynamics [25] and denoising score matching [27]. Following this, approaches such as VDM [10], MCVD [33], Imagen Video [12], and LVDM [10] extended diffusion models to video generation, surpassing GANs in both sample quality and distribution coverage due to their stable training and scalability [3]. However, this success comes hand in hand with significant challenges posed by enormous model sizes and computational demands associated with diffusion models. These challenges manifest themselves in both the inference and training aspects.

Sampling from diffusion models is expensive as their immense number of parameters, heavy reliance on attention mechanisms, and need for several model evaluations result in substantial memory consumption. Even with advanced GPUs, tackling high-resolution video generation becomes a formidable burden due to this memory constraint. Some research efforts, such as Imagen Video [12], introduce a model chain to enhance video generation quality gradually. However, this approach further escalates the total model parameters and memory consumption. Beyond this, the extensive computational load leads to a significantly longer inference latency, amplifying the deployment cost and user waiting time. These barriers have presented substantial impediments to the commercialization of diffusion models, particularly in the context of video diffusion models.

Furthermore, when it comes to the training of diffusion models, challenges emerge from three key facets. Firstly, due to the substantial model parameter count and computational overhead, training costs soar, often requiring an entire month or even longer to train large-scale diffusion models on large datasets from scratch. This protracted training duration poses a challenge to the improvement of diffusion models. Moreover, given that diffusion models are still relatively nascent, our prior knowledge regarding their structural design remains limited. Consequently, model design heavily relies on trial and error, incurring additional expenses in terms of both time and resources. Lastly, because the objectives of these models vary widely, which has different model size constraints and different target video generation resolutions, model architectures often need to be tailored differently to suit each specific goal. Training these diverse models with distinct structures for varying objectives introduces additional overhead that can be burdensome and difficult to manage.

In the face of these challenges, it becomes imperative to explore strategies that mitigate the computational burden and streamline the network design process achieving different targets including cost constraints and resolution requirements at the same time. This is crucial not only for enhancing the efficiency of diffusion models but also for facilitating their broader applicability across various real-world scenarios.

In this paper, we introduce SNED, a **s**uperposition **n**etwork architecture search method for **e**fficient video **d**iffusion models, designed to achieve efficient model implementation without compromising high-quality generative performance. We explore the combination of network search with video diffusion model and enable a flexible range of options towards resolution and model cost, saving computation consumption for inference and training. Specifically, we implement a one-shot neural architecture search solution, enabling dynamic computation cost sampling. This means that once the supernet is trained, it achieves the differentiation of computational costs across various subnets within the supernet. Besides that, we introduce the concept of "super-position training" into our supernet training process. This breakthrough allows a singular supernet model to effectively manage different resolutions, offering a versatile solution for handling diverse resolution requirements. Consequently, this approach permits the re-utilization of super-resolution models in multiple instances, facilitating the training of models with a diverse range of cost and resolution options concurrently.

The contributions of this paper include:

- A video diffusion model supernet training paradigm that trains subnets with different model sizes and resolution options through a weight-sharing method.
- Increasing the search space in different search dimensions including dynamic channels and fine-grained dynamic blocks.
- The supernet training sampling warmup strategy to improve the training performance.
- Being compatible with different base architectures such as pixel-space and latent-space video diffusion models.
- According to the experiment for pixel-space video diffusion model, we can achieve consistent video generation results simultaneously across 64×64 to 256×256 resolutions with a large range of model sizes from 640M to 1.6B number of parameters for pixel-space video diffusion models.

2. Related Work

2.1. Classic Video Synthesis

Classic video synthesis endeavors to capture the underlying distribution of real-world videos, allowing the generation of realistic and novel video samples. Previous research primarily leverages deep generative models, including GANs [21, 22, 24, 28, 30, 34, 40], autoregressive models [6, 37], VAEs [9, 35], and normalizing flows [15]. Among these, GAN-based approaches stand out as the most dominant, owing to the remarkable success of GANs in image modeling.

MoCoGAN [30] and MoCoGAN-HD [28] excel in decomposing latent codes into content and motion subspaces. Notably, MoCoGAN-HD [28] utilizes the potent pretrained StyleGAN2 as the content generator, resulting in higher-resolution video generation. StyleGAN-V [24] and DiGAN [40] introduce implicit neural representation to GANs, facilitating the modeling of temporal dynamics continuity. These models build upon StyleGAN3 and employ a hierarchical generator architecture for long-range modeling, enabling the generation of videos with evolving content over time.

Despite the success of GANs, they often face challenges such as mode collapse and training instability. Autoregressive methods have also been explored for video generation. VideoGPT [37], utilizing VQVAE [32] and a transformer, autoregressively generates tokens in a discrete latent space. TATS [6] enhances the VQVAE [32] with a more powerful VQGAN [4] and integrates a frame interpolation transformer for rendering long videos in a hierarchical manner.

2.2. Diffusion Model

Besides the classic video synthesis models, diffusion models, a category of likelihood-based generative models, have exhibited notable advancements in image and video synthesis tasks, surpassing GANs in both sample quality and distribution coverage due to their stable training and scalability [3]. Noteworthy among these models is DDPM [11], which establishes a weighted variational bound for optimization by connecting Langevin dynamics [25] and denoising score matching [27]. Despite its slow sampling process requiring step-by-step Markov chain progression, DDIM [26] accelerates sampling iteratively in a non-Markovian manner, maintaining the original training process [16]. ADM [3] outperforms GAN-based methods with an intricately designed architecture and classifier guidance.

While diffusion models have excelled in image synthesis, their application to video generation has been limited. VDM [10] extends diffusion models to the video domain, introducing modifications such as a spatial-temporal factorized 3D network and image-video joint training. MCVD [33] unifies unconditional video generation and conditional frame prediction through random dropping of conditions during training, akin to the classifier-free guidance approach. Make-A-Video [23] and Imagen Video [12] leverage diffusion models for large-scale video synthesis conditioned on text prompts, which conduct diffusion and denoising processes in pixel space. Besides that, LVDM [10] extends the video generation work to the latent space and explores how hierarchical architectures and natural extensions of conditional noise augmentation enable the sampling of long videos. However, the efficiency model optimization of the video diffusion model is still waiting for exploration. In this paper, we further explore the combination of network search with the video diffusion model and enable a flexible range of options toward resolution and model cost.

2.3. Neural Architecture Search

2.3.1 NAS Strategies

There is a growing trend in designing efficient Deep Neural Networks (DNNs) through Neural Architecture Search (NAS). NAS strategies can be broadly categorized into the following approaches based on their searching strategies. Firstly, Reinforcement Learning (RL) methods, such as [41–43], utilize recurrent neural networks as predictors to validate the accuracy of child networks over a proxy dataset. Secondly, Evolution methods, exemplified by works [18, 19], employ a pipeline involving parent initialization, population updating, and the generation and elimination of offspring to discover desired networks. Thirdly, One-Shot NAS, as demonstrated in studies such as [1, 8, 38], involves training a large one-shot model containing all operations and shares the weight parameters among all candidate models.

Weight-sharing NAS, inspired by the above methodologies, has gained popularity due to its training efficiency [20, 36, 39]. In this approach, an over-parameterized supernet is trained with weights shared across all sub-networks in the search space, significantly reducing computational costs during the search.

Although most of the mentioned works primarily focus on traditional Convolutional Neural Network (CNN) architectures, recent studies have extended the scope to include the search for efficient Vision Transformer (ViT) architecture. Examples include Autoformer [2], which entangles the model weights of different ViT blocks in the same layer during supernet training with an efficient weight-sharing strategy. This approach reduces both the training model storage consumption and the overall training time.

2.3.2 Generation Model NAS

While generation models have achieved significant success in designing neural architectures, their implementation often demands substantial time, effort, and expert knowledge. For instance, [13] devised intricate generators and discriminator backbones to efficiently generate high-resolution images. Recognizing the need to alleviate the burden of network engineering, researchers have explored efficient automatic architecture search techniques for GANs.

In 2019, AutoGAN [7] introduced an architecture search scheme for GANs utilizing NAS algorithms. It defined a search space to capture deformations in GAN architecture and employed an RNN controller to guide search operations. Later, AutoGAN-Distiller (AGD) [5] is developed by applying AutoML to GAN compression. AGD performs end-to-end search for efficient generators based on the original GAN model via knowledge distillation. In 2021, alphaGAN [29] is introduced, which is a fully differentiable search framework solving bi-level minimax optimization problems. Later, StyleGAN2 [14] expanded the search space by integrating backbone characteristics.

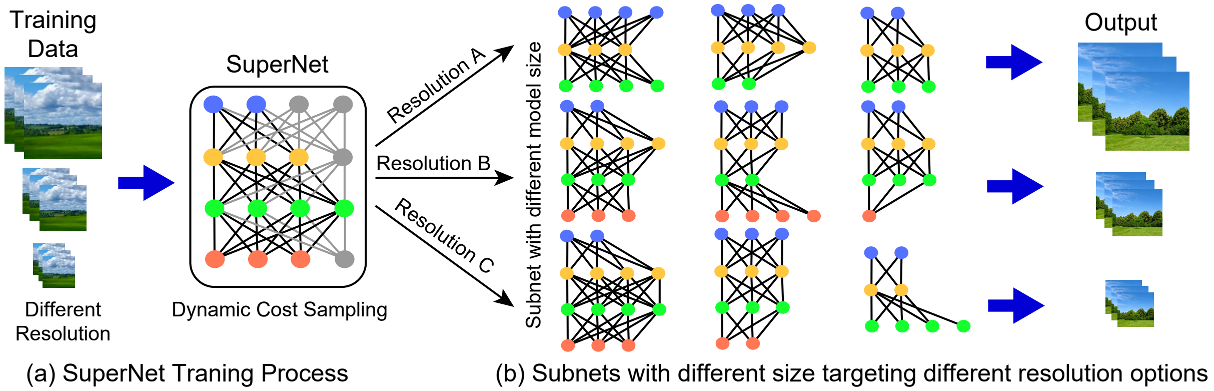
While the majority of studies have concentrated on GAN-based generation models, the research realm of video diffusion model NAS remains largely unexplored. Given the substantial computation demands of video diffusion models, there is a critical need to delve into more efficient video diffusion model architecture designs.

3. Methodology

3.1. Overview of SNED

In this paper, we present a framework termed the "SNED: **S**uperposition **N**etwork Architecture Search for **E**fficient Video **D**iffusion Models", designed to effectively search for and optimize video diffusion models across multiple dimensions. Our framework introduces two key advancements that address critical challenges in video generation.

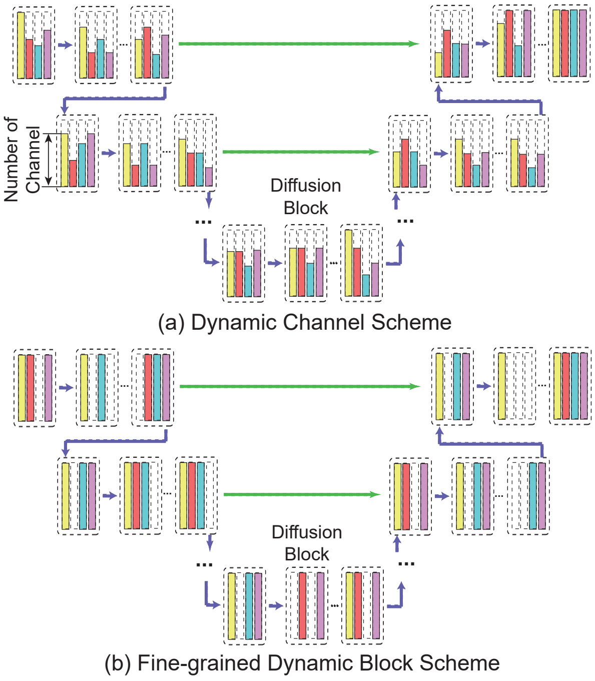
The overview framework of SNED is shown in Fig. 1. Firstly, we implement a one-shot Neural Architecture Search (NAS) solution, enabling dynamic computation cost sampling. This means that once the supernet is trained, it achieves the differentiation of computational costs across various subnets within the supernet. This feature empowers users to select the appropriate subnetwork based on specific model sizes and computational cost constraints, enhancing flexibility and adaptability. Secondly, we introduce the concept of "super-position NAS training" into our supernet training. This breakthrough allows a singular supernet model to effectively manage different resolutions, offering a versatile solution for handling diverse resolution requirements. Consequently, this approach permits the reutilization of super-resolution models in multiple instances, considerably mitigating the memory overhead within the video diffusion model framework. By leveraging these advancements, our framework not only streamlines the intricate process of video diffusion model optimization but also



(a) SuperNet Traning Process

(b) Subnets with different size targeting different resolution options

Figure 1. Overview of SNED framework. (a) We train a supernet with network dynamic cost sampling and multiple input resolution options. In each iteration, a subnet of the supernet is sampled for the training, and other parts (grey) is frozen. (b) After the training, we obtain subnets with different model costs for each resolution option.



(a) Dynamic Channel Scheme

(b) Fine-grained Dynamic Block Scheme

Figure 2. Dynamic cost scheme for SNED framework.

substantially reduces memory consumption, paving the way for more efficient and resource-conscious video generation processes.

3.2. Dynamic Cost Training in SNED

In each iteration, we randomly select a sampled subnet architecture from the search space and obtain its weights from the supernet. We then compute the losses of the subnet and update the corresponding weights with the remaining supernet weights frozen. The architecture search space P is encoded in a supernet denoted as $\mathcal{S}(P, W_P)$, where W_P is the weight of the supernet that is shared across all the candidate architectures. Algorithm 1 illustrates the training procedure of our supernet. Here, our dynamic cost search space includes the dynamic channel space and the fine-grained dynamic block space. The schemes of these two search spaces are shown in Fig. 2. Here, different color bars denote the different components inside a diffusion model, which include ResBlock, temporal self-attention, temporal cross-attention, and spatial attention. The length of the color bars denotes the number of channels of the subnets sampled during training.

Algorithm 1 Superposition Supernet Training.

```
Input: Training iteration $N$, search space $\mathcal{P}$, supernet $\mathcal{S}$, loss function $L$, train dataset $D_{train}$, initial supernet weights $\mathcal{W}$, candidate weights $\mathcal{W}_p$, Output resolution $R$.

for $i$ in $N$ iterations do
    for data, labels in $D_{train}$ do
        Randomly sample one subnet architecture and resolution $R$ from search space $\mathcal{P}$.
        Obtain the corresponding weights $\mathcal{W}_p$ from supernet $\mathcal{W}$
        Compute the gradients based on $L$
        Update the corresponding part of $\mathcal{W}_p$ in $\mathcal{W}$ while freezing the rest of the supernet $\mathcal{S}$
    end for
end for
Output $\mathcal{S}$
```

Dynamic Channel Search Space: As the different numbers of channels have different acceleration performances for the hardware implementation, the SNED search space includes replacing the original number of channels with different percentage ratios, including 100% (full number of channels), 90%, 80%, 70%, 60%, 50%, and 40%. Each layer inside the diffusion blocks can be assigned an independent ratio in each iteration of supernet training.

Fine-grained Dynamic Block Search space: To expand our search space during the supernet training and investigate the potential of the video diffusion model, we add the fine-grained dynamic block search process inside each diffusion block. The basic supernet diffusion block contains four components: ResBlock (convolutional residual block), temporal self-attention block, cross-attention block, and spatial attention block. Our Algorithm enables the drop of a part of the blocks inside the whole diffusion block in each iteration of supernet training. Specifically, if all the attentions inside the diffusion block are dropped, the corresponding feed-forward layer will also be dropped.

3.3. Super-position Training in SNED

We introduce the super-position training mechanism to address different video resolution targets during the supernet training. Here, super-position refers to the utilization of weight-sharing techniques, allowing different subnets to adjust to various resolution processing needs while keeping most of their weights shared. This approach serves the dual purpose of parameter efficiency and the ability to achieve video diffusion models with different resolutions in a single training session.

During each training iteration, besides the sampling of the subnet, we also randomly sample a video generation resolution and preprocess the training data based on that. To balance the training memory workload of different resolution branches, we constraint the maximum model size for different resolutions to ensure an acceptable memory consumption. By leveraging this super-position training method, we are not only optimizing the model's resource allocation but also streamlining the training process itself. This minimizes the computational burden and accelerates the development of video diffusion models tailored to different resolution needs.

3.4. Supernet Training Sampling Warmup

To achieve a better and faster NAS training performance, we propose the supernet training sampling warmup strategy. This strategy is deployed at the beginning of the supernet training process, improving the supernet's stability and robustness during training.

We gradually increase our search space for both fine-grained dynamic block and dynamic channel during the training, rather than directly applying a full random subnet sampling among the whole search space at the beginning. Specifically, we will apply 30000 training iterations for sampling warmup. The minimum percentage of channels and fine-grained blocks will be decreased from 100% to 40% in a step schedule manner.

4. Experimental Result

4.1. Experimental Setup

In this section, we present the configuration of our SNED framework. Our experiments consist of two primary components: the pixel-space video diffusion model and the latent video diffusion model. To enable the different resolution options under the super-position mechanism, we process the training data into a form suitable for training our cascading pipeline, we spatially resize videos using antialiased bilinear resizing to different resolutions including 64×64, 128×128, and 256×256. To enable the text-to-video conditional training, a frozen text-encoder [17] is added at the beginning of the model pipeline.

We train the pixel-space video diffusion model pipeline using an internal dataset comprising 19 million video-text pairs. For the base model and spatial super-resolution (SSR) model inside the pipeline, we use a total batch size of 256 and 64 during training, respectively. Both models undergo 140,000 training iterations, with a fixed learning rate of 0.0001. The training process utilizes 64 A100 GPUs.

For the latent-space video diffusion model, We start with LVDM [10] as a baseline and subsequently train it using our algorithm. For a fair comparison, we employ the same publicly available datasets Sky Timelapse. The hyperparameter settings for our experiments align with those of LVDM [10] to ensure a fair evaluation.

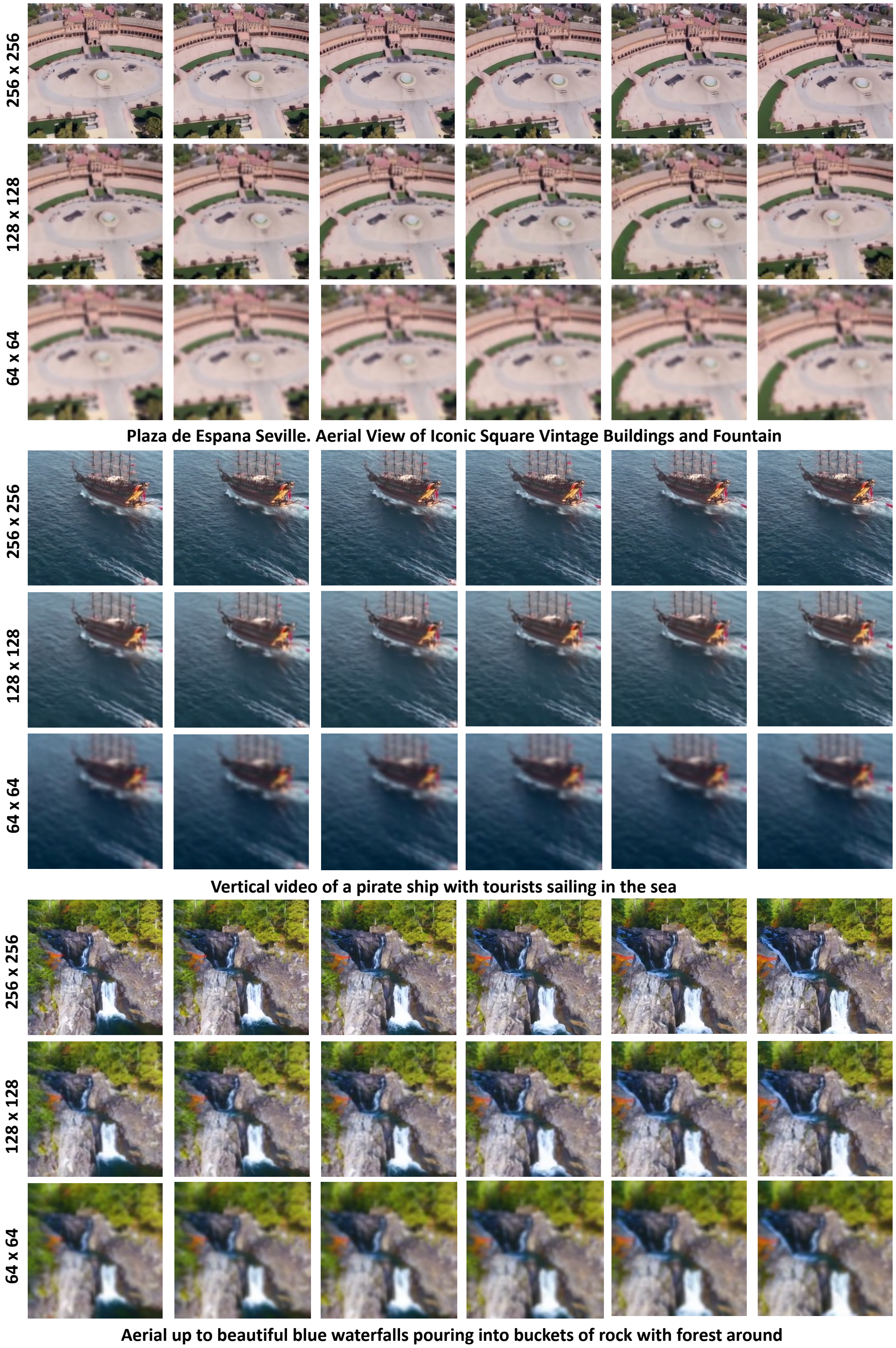
4.2. Pixel-Space Video Diffusion Model NAS

For the pixel-space video diffusion model, our approach is inspired by the model chain proposed by Imagen-video [12] to realize high-quality video generation. The model chain comprises the base model and the spatial super-resolution model (SSR). The base model and SSR are determined by our framework (SNED) to meet various computational resource constraints and resolution targets. Our SNED framework allows for different resolution options in SSR model with weight-sharing subnets. For the supernet architecture of both the base model and SSR model, we apply an imagen-like modified 2D UNet. Each block inside the UNet consists of ResBlock, temporal self-attention and cross-attention, and spatial-attention.

To attain different resolutions, we recursively deploy our SSR model multiple times instead of integrating multiple SSR models, as demonstrated in Imagen-video [12]. This approach significantly reduces the total model size.

4.3. Latent-Space Video Diffusion Model NAS

Given that Imagen-video does not release its model and dataset, conducting a direct comparison is challenging. To



Plaza de Espana Seville. Aerial View of Iconic Square Vintage Buildings and Fountain

Vertical video of a pirate ship with tourists sailing in the sea

Aerial up to beautiful blue waterfalls pouring into buckets of rock with forest around

Figure 3. Results of pixel-space video diffusion model for different resolution options.

showcase the flexibility and efficiency of our framework, we add additional autoencoder and autodecoder to a latent-space video diffusion model for evaluation. The whole model pipeline follows the basic version of LVDM [10] [1]. We first compress video samples to a lower dimensional latent space by the video autoencoder. Then we perform the video generation in the latent space. The encoder and decoder both consist of several layers of 3D convolutions. To ensure that the autoencoder is temporally shift-equivariant, we follow [10] to use repeat padding in all three-dimensional convolutions. The prediction model applies a 3D U-Net architecture to estimate the noise distribution, which consists of space-only $1 \times 3 \times 3$ shape 3D convolution, and spatial attention module.

We start with LVDM [10] as a baseline and subsequently train each part of it inside our NAS framework. Similar to the pixel-space diffusion model, we apply the super-position NAS training to the diffusion prediction model. Since the encoder model is only applied during the training stage, we only apply the super-position training on it without the dynamic cost NAS.

4.4. Evaluation Results

4.4.1 Pixel-Space Video Diffusion Model Visualization

In Fig. 3 and Fig. 4, we present the results of our pixel-space diffusion model. Fig. 3 shows the generation results from the pixel-space SSR model. We show 6 frames for each of them. Due to the space limitation, we only show the full model size (428M) result of SSR for different resolution options. The corresponding text prompts are listed under each group of video frames. Fig. 4 illustrates the visualization of the pixel-space base model, transforming the input text (depicted on the left side of the figure) into the corresponding output video. For clarity, we showcase three frames from each video using two different noise seeds. Additionally, for each input text, we display results generated by models of varying sizes—40% (640M), 60% (960M), 80% (1.28B), and 100% (1.6B) of the parameters compared to the supernet with 1.6B number of parameters. This comprehensive visualization highlights the stability and adaptability of our video generation process achieved through the SNED training strategy.

4.4.2 Latent-Space Video Diffusion Model Visualization

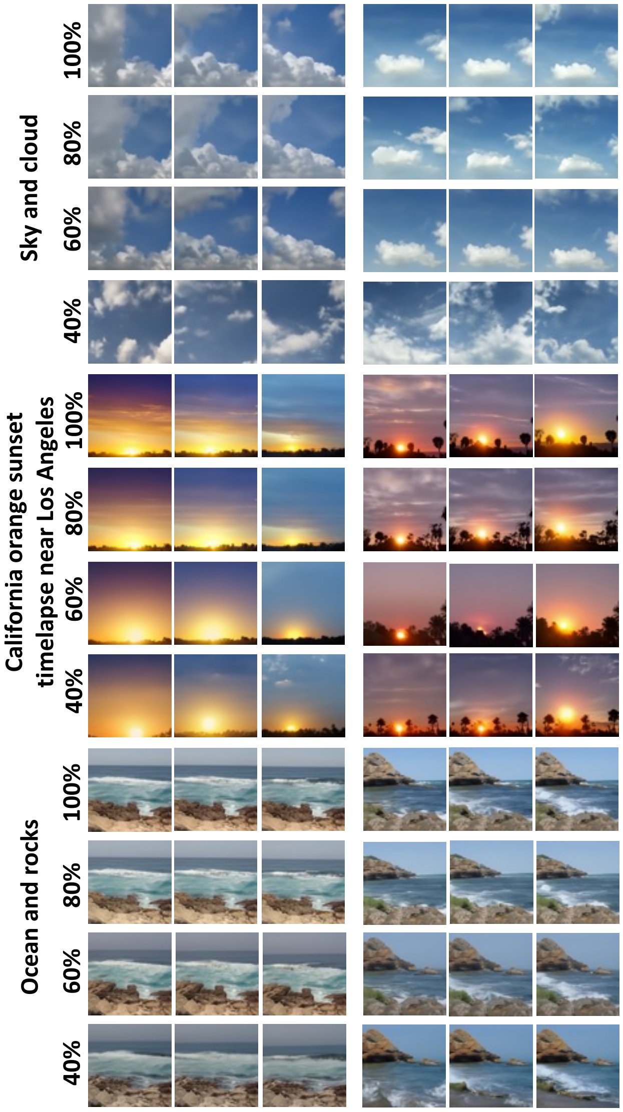


Figure 4. Result of different pixel-space base model subnets with different model sizes. The values of percentage indicate the relative model size compared with the supernet. We show the results of each subnet with two different noise seeds.

The results from the latent-space video diffusion model are depicted in Fig. 5. In this comparison, we present the outcomes from three subnets with distinct model sizes (548M, 411M, and 274M) and compare them with the released model from LVDM[10]. The original output of LVDM[10] is featured in the first row for reference. All showcased videos in the figure use a consistent resolution of 256×256 and comprise the same number of frames (16) for unconditional short video generation, aligning with the specifications employed in LVDM [10]. We show the first frame of each generated video.

From Fig. 5, we can see that, compared with the original LVDM model, all three subnets provide comparable output results for the sky timelapse. Both LVDM and SNED provide generation outputs with high fidelity and diversity, in-

[1] Since the authors have not released the long-term version training code, we follow and compare it with their released basic latent diffusion model on GitHub. The matrix score is different from that they showed in their paper.

cluding different skies, clouds, and ground at different times of the day.

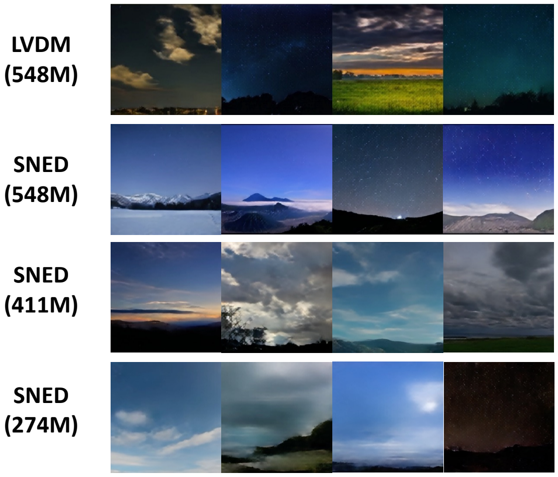


Figure 5. Comparison with LVDM under the resolution of 256×256 on Sky Time-lapse dataset. We present the first frame of each video. Three subnets with different numbers of parameters are included in the comparison.

4.5. Model Matrix Evaluation

For quantitative evaluation, we report the commonly-used FVD [31] and KVD [31] in our experiment. For the pixel-space video diffusion base model, we calculate FVD and KVD scores between 512 real and fake videos with 12 frames, which are presented as FVD_{12} and KVD_{12}. All results for the score evaluation are calculated among ten runs to get the average value. The computation is based on the internal dataset comprising 19 million video-text pairs. The latency evaluation is based on one Nvidia A100 GPU.

Table 1. Quantitative comparisons of different subnets for pixel-space video diffusion base model.

Model	#Params (B)	$FVD_{12} \downarrow$	$KVD_{12} \downarrow$	Time (s)
SNED-B	1.60	544.4	25.8	24.4
SNED-L	1.28	490.5	13.0	21.2
SNED-M	0.96	452.2	14.4	18.1
SNED-S	0.64	472.3	16.8	16.0

As shown in Table 1, we report the quantitative evaluation for our SNED models of varying sizes —- small size 40% (640M), medium size 60% (960M), large size 80% (1.28B), and base size 100% (1.6B) of the parameters compared to the supernet (1.6B), which are indicated as SNED-S, SNED-M, SNED-L, and SNED-B, respectively. From the results, we can see that all of the subnets show a stable score according to both FVD and KVD, which proves our framework's robustness. Small subnets even obtain better FVD and KVD scores compared with the supernet (SNED-B). Among them, SNED-M achieves the best FVD score (452.2), and SNED-L achieves the best KVD score (13.0). Our smallest subnet SNED-S obtains a 472.3 FVD score and a 16.8 KVD score with only 16.0s latency, which achieves 1.53× of speedup with a better matrix score compared with the supernet model (latency 24.4s).

Table 2. Quantitative comparisons of different subnets under resolution of 256×256.

Model	#Params (M)	$FVD_{16} \downarrow$	$KVD_{16} \downarrow$	Time (s)
LVDM	548	295.1	20.9	86.8
SNED	548	298.3	20.8	86.8
SNED	411	348.2	23.5	74.2
SNED	274	472.3	28.7	66.7

For the latent-space diffusion model, we compare our matrix score with the baseline LVDM [10] and report them in Table 2. The score computation process follows that used in [10], utilizing 16 frames of generated fake videos for evaluation on the Sky Timelapse dataset. Here we use the released model (548M number of parameters) from LVDM as our supernet architecture, then train it with our dynamic cost schemes. Model size options of 548M, 411M, and 274M are shown in the Table.

5. Conclusion

This paper introduces SNED, the superposition network architecture search for an efficient video diffusion model. In our training paradigm, we target various model cost and resolution options using a weight-sharing method and incorporate both dynamic channel and fine-grained dynamic block to expand our search space. Additionally, we propose the supernet training sampling warmup to improve the training performance. Our proposed method is compatible with different base architectures such as pixel-space and latent-space video diffusion models. According to the experimental results for the pixel-space video diffusion model, we can achieve consistent video generation results simultaneously across 64×64 to 256×256 resolutions with a large model size range from 640M to 1.6B number of parameters. To the best of our knowledge, this is the first NAS framework targeting the video diffusion model.

6. Acknowledgement

This work was completed during Zhengang's internship at Adobe Research and was partially supported by the National Science Foundation under grant number CNS-1909172.

References

[1] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, pages 550–559, 2018. 3

[2] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. *arXiv preprint arXiv:2107.00651*, 2021. 3

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[5] Yonggan Fu, Wuyang Chen, Haotao Wang, Haoran Li, Yingyan Lin, and Zhangyang Wang. Autogan-distiller: Searching to compress generative adversarial networks. *arXiv preprint arXiv:2006.08198*, 2020. 3

[6] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *European Conference on Computer Vision*, pages 102–118. Springer, 2022. 1, 2

[7] Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3224–3234, 2019. 3

[8] Zichao Guo, Xiangyu Zhang, Haoyuan Mu, Wen Heng, Zechun Liu, Yichen Wei, and Jian Sun. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, pages 544–560. Springer, 2020. 3

[9] Jiawei He, Andreas Lehrmann, Joseph Marino, Greg Mori, and Leonid Sigal. Probabilistic video generation using holistic attribute control. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018. 1, 2

[10] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 1, 2, 3, 5, 7, 8

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2

[12] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 1, 3, 5

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 3

[15] Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv preprint arXiv:1903.01434*, 2019. 1, 2

[16] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[17] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020. 5

[18] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Aging evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, 2019. 3

[19] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, pages 4780–4789, 2019. 3

[20] Manas Sahni, Shreya Varshini, Alind Khare, and Alexey Tumanov. Compofa: Compound once-for-all networks for faster multi-platform deployment. *arXiv preprint arXiv:2104.12642*, 2021. 3

[21] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE international conference on computer vision*, pages 2830–2839, 2017. 1, 2

[22] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10-11): 2586–2606, 2020. 1, 2

[23] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

[24] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3626–3636, 2022. 1, 2

[25] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 1, 2

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2

[27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2

[28] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. *arXiv preprint arXiv:2104.15069*, 2021. 1, 2

[29] Yuesong Tian, Li Shen, Guinan Su, Zhifeng Li, and Wei Liu. Alphagan: Fully differentiable architecture search for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6752–6766, 2021. 3

[30] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018. 1, 2

[31] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 8

[32] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[33] Vikram Voleti, Alexia Jolicoeur-Martineau, and Chris Pal. Mcvd-masked conditional video diffusion for prediction, generation, and interpolation. *Advances in Neural Information Processing Systems*, 35:23371–23385, 2022. 1, 2

[34] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in neural information processing systems*, 29, 2016. 1, 2

[35] Jacob Walker, Ali Razavi, and Aäron van den Oord. Predicting video with vqvae. *arXiv preprint arXiv:2103.01950*, 2021. 1, 2

[36] Dilin Wang, Meng Li, Chengyue Gong, and Vikas Chandra. Attentivenas: Improving neural architecture search via attentive sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6427, 2021. 3

[37] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 1, 2

[38] Shan You, Tao Huang, Mingmin Yang, Fei Wang, Chen Qian, and Changshui Zhang. Greedynas: Towards fast one-shot nas with greedy supernet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1999–2008, 2020. 3

[39] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *European Conference on Computer Vision*, pages 702–717. Springer, 2020. 3

[40] Sihyun Yu, Jihoon Tack, Sangwoo Mo, Hyunsu Kim, Junho Kim, Jung-Woo Ha, and Jinwoo Shin. Generating videos with dynamics-aware implicit generative adversarial networks. *arXiv preprint arXiv:2202.10571*, 2022. 1, 2

[41] Zhao Zhong, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Practical block-wise neural network architecture generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2423–2432, 2018. 3

[42] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[43] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 8697–8710, 2018. 3