# Video Generation Models and Optimization Techniques

Zhengze Yu
University of Chinese Academy of Sciences (UCAS)
yuzhengze21@mails.ucas.ac.cn

## Abstract

*In recent years, the surge of large models in the field of text generation has sparked imagination about their potential in other domains. Among these, video—an essential medium of expression in social media—has become deeply integrated into everyday learning and life. Naturally, a series of remarkable applications have emerged in the area of video generation. For example, OpenAI's Sora [6] is capable of generating minute-long videos that flexibly simulate various aspects of the real world; similarly, Kuaishou's Keling AI can rapidly produce short videos or extend given image–text inputs into videos of up to three minutes. Although these applications are still in their infancy and have yet to truly understand the physical world, their groundbreaking achievements and impressive performance in terms of temporal continuity, consistency, and diversity have deeply inspired me and ignited my enthusiasm for exploration in this field.*

*Through this literature review, I gained a comprehensive understanding of the video generation landscape, including its fundamental paradigms, the underlying models within these paradigms, and the evolution and variants of the Diffusion Model [3]. While appreciating the remarkable capabilities of these models, I also realized that their success relies heavily on vast computational resources, which raises questions about the feasibility of further scaling. Consequently, I explored optimization techniques such as model acceleration, compression, knowledge distillation, and efficient fine-tuning. The structure of this review follows this logic: Section 1 introduces video generation methods; Section 2 discusses representative models; and Section 3 presents model optimization techniques. Finally, I summarize the insights gained from this study and reflect on how this process has enhanced my understanding of literature review methodology and its importance for future research work.*

## 1. Paradigms and Principles of Video Generation Models

### 1.1. Paradigms for Long Video Generation

In video generation, one of the most desirable goals is the ability to produce temporally continuous videos lasting several seconds or even minutes at a stable frame rate, rather than short transient motions. Due to computational constraints, existing models are unable to generate very long sequences directly. Therefore, paradigms for long video generation have been proposed to decompose this complex task into tractable sub-processes. These paradigms focus on generating individual frames or short clips that can be logically assembled into coherent long videos. Two major paradigms dominate long video generation: the **divide-and-conquer** paradigm and the **temporal autoregressive** paradigm. As illustrated in Figure 1, the divide-and-conquer approach first identifies key frames and filler frames, generating intermediate frames to weave a temporally consistent long video. In

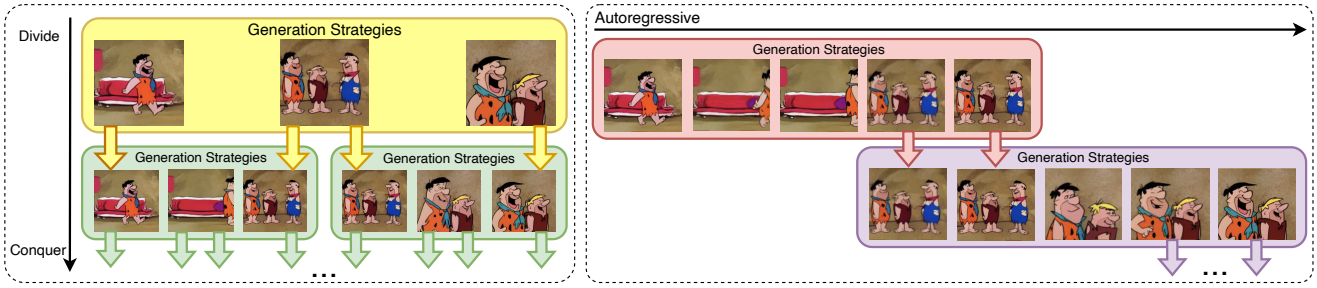


图 1. Illustration of the divide-and-conquer and temporal autoregressive paradigms. Image adapted from [6].

contrast, the temporal autoregressive paradigm generates short video segments sequentially based on prior conditions. It abandons hierarchical structures and directly produces fine-grained segments guided by information from previously generated frames.

### 1.2. Model Principles

**Diffusion Model [3].** The Diffusion Model is a powerful generative framework that produces data by reversing a gradual noising process.

In the forward diffusion phase, Gaussian noise is progressively added to the data over multiple steps, transforming it into a noise distribution. This process is modeled as a Markov chain over latent variables $\{x_t\}_{t=0}^{T}$, where $x_0$ denotes the original data and $x_T$ represents the fully noised sample. The forward diffusion process is defined by a series of transition probabilities $q(x_t|x_{t-1})$, which inject noise into the data:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}), \tag{1}$$

where $\{\beta_t\}_{t=1}^{T}$ are small predefined variances that gradually increase noise levels.

The Denoising Diffusion Probabilistic Model (DDPM) interprets generation as a reverse diffusion process, progressively denoising samples to transform random noise into data drawn from the target distribution. DDPM defines a denoising function $\epsilon_\theta(x_t, t)$ to estimate the noise added at each step. The reverse process is formulated as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{2}$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are neural network–parameterized functions that progressively refine the data. Training aims to minimize the difference between noisy data and its denoised reconstruction, typically by optimizing a variational lower bound. The key insight of DDPM is learning to reverse the diffusion process by estimating conditional probabilities at each step. Through iterative denoising, the model can generate high-quality samples from pure noise.

**Transformer [7].** The Transformer architecture operates on the principles of self-attention, multi-head attention, and position-wise feed-forward networks.

The self-attention mechanism allows the model to assign dynamic importance to different parts of an input sequence. It computes attention scores as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{3}$$

where $Q$, $K$, and $V$ represent query, key, and value matrices derived from input embeddings, and $d_k$ is the key dimensionality.

The multi-head attention mechanism enhances the model' s ability to capture diverse dependencies by applying $h$ distinct learned projections to $Q$, $K$, and $V$, computing attention in parallel, and concatenating the results:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)\mathcal{W}^O, \tag{4}$$

where

$$\text{head}_i = \text{Attention}(Q\mathcal{W}_i^Q, K\mathcal{W}_i^K, V\mathcal{W}_i^V), \tag{5}$$

and $\mathcal{W}_i^Q$, $\mathcal{W}_i^K$, $\mathcal{W}_i^V$, and $\mathcal{W}^O$ are learnable parameter matrices.

The position-wise feed-forward network (FFN) in each Transformer layer applies two linear transformations separated by a ReLU activation:

$$\mathrm{FFN}(x) = \max(0, x\mathcal{W}_1 + b_1)\mathcal{W}_2 + b_2, \tag{6}$$

where $\mathcal{W}_1$, $b_1$, $\mathcal{W}_2$, and $b_2$ denote weights and biases. Each sublayer is wrapped with residual connections and layer normalization:

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x)), \tag{7}$$

where $\mathrm{Sublayer}(x)$ represents the operation of the sublayer. To encode positional information absent in attention, positional encodings are added to the input embeddings:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\mathrm{model}}}), \tag{8}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\mathrm{model}}}), \tag{9}$$

where $pos$ denotes the position and $i$ the dimension index.

**GAN (Generative Adversarial Networks) [2].** Generative Adversarial Networks (GANs) are unsupervised learning frameworks that pit two neural networks against each other in a zero-sum game. The *generator* aims to produce data indistinguishable from real samples, while the *discriminator* seeks to differentiate between real and generated data. The overall optimization objective can be expressed as:

$$\max_D \min_G \mathbf{V}(G, D), \tag{10}$$

with the value function defined by:

$$\mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]. \tag{11}$$

Here, $x$ is drawn from the real data distribution $p_{\mathrm{data}}(x)$, and $z$ is a latent vector sampled from a prior distribution $p_z(z)$ (usually uniform or Gaussian). During training, $D$ and $G$ are alternately optimized: $D$ improves at distinguishing real from fake data, while $G$ learns to fool $D$. The process continues until the generator produces data indistinguishable from real samples.

**Autoregressive Models.** Autoregressive (AR) models form a family of statistical models used to describe and predict future values in a temporal sequence. An AR model assumes that the current observation is a linear combination of its previous observations plus a noise term. An AR model of order $o$, denoted AR($o$), is defined as:

$$v_t = \sum_{i=1}^{o} \theta_i v_{t-i} + \epsilon_t, \tag{12}$$

where $v_t$ is the current value, $\theta_1, \theta_2, \ldots, \theta_o$ are model parameters, $v_{t-1}, \ldots, v_{t-o}$ are previous observations, and $\epsilon_t$ is white noise with zero mean and finite variance. Autoregressive modeling has been extended to capture complex temporal dependencies and serves as the foundation of various sequence modeling architectures. By modeling dependencies among sequence elements, AR models can generate or predict subsequent elements, making them essential for tasks such as language modeling, time-series forecasting, and generative image or video modeling.

## 2. Diffusion Models and Video Generation [5]

### 2.1. Development

The introduction of diffusion models by Ho et al. marked a milestone in the field of text-to-image (T2I) generation, paving the way for influential applications such as DALL · E, Midjourney, and Stable Diffusion. Since videos can be regarded as sequences of images enriched with both spatial and temporal information, researchers soon began exploring diffusion-based architectures for generating high-fidelity videos from textual descriptions.

The Video Diffusion Model (VDM) represents a pioneering advancement in text-to-video (T2V) generation, extending conventional image diffusion frameworks to handle video data. VDM addresses the critical challenge of maintaining temporal consistency while generating realistic videos. Through an innovative 3D U-Net architecture, traditional 2D convolutions are en-

hanced into 3D operations and integrated with temporal attention mechanisms to ensure temporally coherent video sequences.

Similarly, MagicVideo and its successor, MagicVideo-V2, mark significant innovations in T2V generation. These models adapt latent diffusion frameworks to address challenges such as limited data availability, complex temporal dynamics, and high computational costs. MagicVideo introduces a 3D U-Net–based architecture with additional modules such as a video distribution adapter and directional temporal attention, enabling efficient, high-quality video synthesis with strong temporal consistency and realism. Operating in latent space, the model focuses on keyframe generation and efficient video composition. MagicVideo-V2 builds upon this with a multi-stage pipeline encompassing text-to-image, image-to-video, video-to-video, and frame interpolation modules.

The use of latent spaces has become a recurring theme among many frameworks. The Latent Video Diffusion Model (LVDM) introduces a hierarchical structure that compresses videos into low-dimensional latent representations, enabling efficient long-video generation with reduced computational requirements. Traditional frameworks are typically restricted to generating short clips constrained by the number of frames provided during training. To overcome this limitation, LVDM introduces a conditional latent diffusion mechanism that generates future latent codes autoregressively based on previous ones. Models such as Show-1, PixelDance, and SVD employ both pixel-level and latent-level diffusion techniques to produce high-resolution videos. Show-1 first generates low-resolution but temporally accurate keyframes using pixel-level VDMs, and then enhances them with latent-level VDMs to achieve high-quality and computationally efficient video output. PixelDance, built upon the latent diffusion paradigm, trains a denoising autoencoder in the latent space of a pretrained VAE to minimize computational costs. Its core is a 2D U-Net diffusion model extended into a 3D variant by incorporating temporal layers—including 1D convolutions and temporal attention—enabling it to handle video content while retaining effectiveness on image inputs. By jointly training on both image and video data, it ensures high-fidelity outputs with consistent spatial and temporal resolution. SVD enhances pretrained diffusion backbones by adding temporal convolution and attention layers, efficiently capturing temporal dynamics. Tune-A-Video extends 2D latent diffusion models (LDMs) into spatiotemporal domains for T2V generation by introducing temporal self-attention to capture inter-frame coherence. Temporal self-attention layers are inserted into each transformer block, allowing the model to preserve temporal consistency across frames. This design is further complemented by sparse spatiotemporal attention and selective fine-tuning strategies, updating only projection matrices within attention blocks to improve computational efficiency while maintaining pretrained T2I features.

In the area of video acceleration, the VideoLCM model was designed as a latent consistency model optimized through consistency distillation. It emphasizes reducing computational load and accelerating training. Leveraging large-scale pretrained video diffusion models, it improves efficiency and utilizes a DDIM solver with classifier-free guidance to synthesize high-quality content, enabling fast video generation with minimal sampling steps. VideoCrafter2 further enhances spatiotemporal consistency through an innovative data-level disentanglement strategy that separates motion dynamics from appearance attributes. This facilitates targeted fine-tuning of visual quality without sacrificing motion accuracy. Built upon VideoCrafter1—which integrates a video VAE with a latent video diffusion process—this design compresses videos into compact latent representations before applying diffusion-based generation.

Models such as Make-A-Video and Imagen Video extend text-to-image frameworks into the video domain. Make-A-Video leverages T2I advances without relying on paired text–video data. Its architecture con-

sists of three key components: a pretrained T2I model, spatiotemporal convolution and attention layers, and a frame interpolation network. The T2I backbone first generates static content, which is then temporally expanded via spatiotemporal layers and refined through interpolation for smooth motion and higher frame rates. Imagen Video employs a cascaded diffusion architecture specifically designed for T2V synthesis. It combines a base video diffusion model with subsequent spatial and temporal super-resolution modules, all conditioned on textual prompts to progressively enhance fidelity and resolution. This enables Imagen Video to generate high-quality, temporally consistent videos that align closely with textual descriptions. MotionDiffuse focuses on text-driven human motion generation, emphasizing diversity and fine-grained controllability. It employs a diffusion-based approach coupled with a cross-modal linear transformer that aligns text embeddings with motion representations, allowing independent control over body parts and temporal patterns to ensure realistic, diverse motion synthesis.

Text2Video-Zero, built upon Stable Diffusion, is designed for zero-shot T2V synthesis. Its core improvements introduce motion dynamics into latent encodings for temporal coherence and adopt cross-frame attention mechanisms to ensure consistent object identity and appearance across frames. These modifications allow the generation of high-quality, temporally consistent videos directly from text prompts without additional training or fine-tuning, leveraging the pretrained capabilities of existing T2I models.

NUWA-XL introduces a novel "diffusion-over-diffusion" architecture for ultra-long video generation, addressing the inefficiency and quality degradation of prior methods. It follows a coarse-to-fine strategy: a global diffusion model first produces keyframes outlining the overall structure, followed by local diffusion models that refine details between frames, enabling efficient synthesis of globally coherent and visually detailed long videos.

Unlike approaches that fine-tune pretrained models, Sora aims for the more challenging goal of training a diffusion model from scratch. Drawing inspiration from the scalability of Transformer architectures, OpenAI integrates the DiT framework into its core, replacing the traditional U-Net with a Transformer-based diffusion backbone. This design leverages the Transformer' s scalability to train on massive datasets and handle complex video generation tasks efficiently. Similarly, GenTron combines Transformers and diffusion models to improve training efficiency. Built on the DiT-XL/2 backbone, it tokenizes latent dimensions into non-overlapping patches processed through Transformer blocks. GenTron incorporates adaptive layer normalization and cross-attention for text conditioning, enhancing interaction between textual and visual features. Notably, GenTron-G/2 scales the model to over three billion parameters, deepening and widening Transformer and MLP blocks for large-scale training. W.A.L.T adopts a two-stage framework combining an autoencoder and a novel Transformer design. The autoencoder compresses both images and videos into low-dimensional latent representations, enabling efficient joint training. The Transformer alternates between spatial and spatiotemporal attention within windowed self-attention layers, significantly reducing computational cost while supporting unified image–video generation. Latte extends these innovations by applying a stack of Transformer blocks directly to latent video representations obtained from a pretrained VAE. This approach effectively models complex distributions across spatial and temporal dimensions.

### 2.2. Diffusion Models under the Divide-and-Conquer Paradigm [4]

Within video generation, the divide-and-conquer strategy has inspired the development of hierarchical architectures designed to first generate keyframes that outline the narrative structure, followed by frame-filling modules that complete the motion sequence. Global models specialize in producing story-defining keyframes, while local models focus on filling the tem-

poral gaps. Yin et al. (2023) proposed a 3D U-Net–based diffusion framework tailored for such segmented generation. Likewise, Ge et al. (2022) introduced a hierarchical Transformer architecture that enhances temporal sensitivity and interpolation capability in long-form video synthesis. This hierarchical decomposition allows distinct sub-models to handle different stages of the video creation pipeline, collectively simplifying the production of coherent, seamless long videos.

### 2.3. Diffusion Models under the Temporal Autoregressive Paradigm [4]

In the autoregressive paradigm, long video generation is simplified into sequentially creating temporally conditioned segments, thereby reducing the complexity of direct long-range synthesis. Diffusion models leveraging this paradigm operate in latent space to efficiently manage conditioning information and optimize their architectures for consistent future predictions. This combination of autoregressive modeling and diffusion refinement enables segment-by-segment synthesis of temporally coherent long videos driven by accumulated context from prior frames.

**Compressed Conditioning.** To efficiently handle the complexity of video data while optimizing computational and storage resources, recent studies have explored various compression strategies for diffusion-based models. Approaches include compressing video data into unified 3D latent spaces [**?** **?** ], preserving essential multi-dimensional features, or separating spatial and temporal information into distinct 2D spaces as proposed by Yu et al. (2023). These diverse compression schemes aim to balance representational fidelity with computational efficiency.

**Temporal Layer Integration for Long-Clip Generation.** Long-video generation tasks have been refined into producing individual video clips, prompting architectural adaptations for improved temporal modeling. Key developments include integrating temporal layers—such as attention and convolution modules—into diffusion backbones [**?** **?** ], enabling models to capture complex temporal dependencies. By leveraging conditioning signals and promoting iterative clip generation, these improvements transform latent diffusion models into powerful video generators capable of producing long, temporally coherent sequences.

**Enhanced Training and Generation Strategies.** Beyond architectural innovations, advanced training strategies have greatly improved the capacity of diffusion models to replicate long-form video dynamics. These include two-stage training procedures encompassing unconditional learning of data distributions followed by conditional generation based on specific prompts [**?** ]. Additionally, reuse strategies—iteratively applying and removing noise to mimic natural variability in video content—have emerged as novel techniques for boosting model robustness and performance [**?** ].

## 3. Model Optimization Techniques [1][8]

### 3.1. Quantization

The overall goal of quantization is to reduce the precision of model parameters $\theta$ and intermediate activation maps to lower-bit formats, such as 8-bit integers, while minimizing the loss of generalization performance. The process begins by defining a quantization function that maps continuous weights and activations to a discrete set of values, typically formulated as:

$$Q(r) = \text{Int}\left(\frac{r}{S}\right) - Z \tag{13}$$

where $Q$ denotes the quantization mapping function, $r$ represents the real-valued input (e.g., weights or activations), $S$ is the scaling factor, and $Z$ is the integer zero point. This mechanism, known as *uniform quantization*, ensures evenly spaced quantized levels, although non-uniform strategies also exist. The original real value $r$ can be approximately recovered from its quantized counterpart $Q(r)$ through a dequantiza-

tion process:

$$\tilde{r} = S(Q(r) + Z) \tag{14}$$

Because of inherent rounding errors, the approximated value $\tilde{r}$ may differ from $r$. A critical component of quantization is determining the optimal scaling factor $S$, which divides the range of $r$ into discrete intervals:

$$S = \frac{\beta - \alpha}{2^b - 1} \tag{15}$$

where $\alpha$ and $\beta$ denote the minimum and maximum real values, and $b$ is the number of quantization bits.

### 3.2. Knowledge Distillation

Knowledge distillation techniques, including *soft* and *hard* distillation, facilitate the transfer of learned knowledge from a large and complex "teacher" model to a simpler "student" model. Soft distillation minimizes the Kullback–Leibler (KL) divergence between softened logits (outputs) of the teacher and student models, as expressed by the following objective:

$$\begin{aligned} L_{\text{global}} = (1-\lambda) L_{\text{CE}}(\psi(Z_s), y) + \\ \lambda \tau^2 \text{KL}\left(\psi\left(\frac{Z_s}{\tau}\right) \| \psi\left(\frac{Z_t}{\tau}\right)\right) \end{aligned} \tag{16}$$

where $L_{\text{CE}}$ denotes the cross-entropy loss, $\psi$ represents the softmax function, $Z_t$ and $Z_s$ are the teacher and student logits, respectively, $\tau$ is the temperature parameter controlling distribution smoothness, and $\lambda$ balances the contributions of the KL divergence and cross-entropy loss.

In contrast, hard distillation uses the teacher model' s discrete predictions as labels for training the student model:

$$L_{\text{global, hardDistill}} = \frac{1}{2} L_{\text{CE}}(\psi(Z_s), y) + \frac{1}{2} L_{\text{CE}}(\psi(Z_s), y_t) \tag{17}$$

where $y_t$ represents the teacher' s hard-label decisions. The DeiT approach introduces a novel distillation method for Transformer architectures by adding a *distillation token* analogous to a class token but dedicated to mimicking the teacher' s output. This token interacts directly with other tokens via self-attention layers, demonstrating superior distillation performance. In our experimental setup, the DeiT framework was applied to the CIFAR dataset under computational constraints to evaluate its distillation effectiveness.

### 3.3. Pruning

Pruning in Vision Transformers (ViTs) primarily aims to reduce model complexity by removing redundant parameters, typically through the adjustment of weight kernel dimensions between hidden layers. This objective can be formulated as:

$$\min_{\alpha,\beta} \quad \text{loss}\left(\mathcal{L}\left(l^{(k)} W^{(k)} l^{(k+1)}\right)\right) - \text{loss}\left(\mathcal{L}\left(l^{(k)} \hat{W}^{(k)} l^{(k+1)}\right)\right) < \delta \tag{18}$$

where $\alpha$ and $\beta$ denote the original and pruned dimensions, respectively, and $\delta$ is the predefined threshold ensuring that the loss increase remains within acceptable limits. Determining which dimensions to prune typically involves computing *importance scores*, learned during pretraining or fine-tuning. Zhu et al. and Yang et al. derived these scores from gradient magnitudes of each weight and proposed integrating a *soft gating* layer that hardens during inference to zero out less important dimensions:

$$S_B(W) = \left(\sum_{\omega_b \in B} \frac{\partial \mathcal{L}}{\partial \omega_b}\right)^2 \tag{19}$$

Alternatively, Yu et al. computed importance scores using KL divergence, quantifying the performance difference between models with and without a given module on dataset $\Omega$:

$$S_B(W) = \sum_{i \in \Omega} \text{D}_{\text{KL}}(p_i || q_i) \tag{20}$$

where $p_i$ and $q_i$ represent the loss distributions of the complete and pruned models, respectively.

Recent advancements propose more refined scoring mechanisms. Tang et al. designed a theoretical metric estimating each image patch' s contribution to overall error, improving pruning efficiency. Rao et al. combined local and global features for a more comprehensive token-importance evaluation. Similarly, Yi et al. unified multiple scoring methods into a single loss function, further refining pruning accuracy.

### 3.4. Low-Rank Approximation

In ViTs, each self-attention block first projects the input sequence $X$ using weight matrices $W_Q$, $W_K$, and $W_V$ to obtain representations $Q = W_Q X$, $K = W_K X$, and $V = W_V X$. The attention mechanism is computed as $\text{softmax}(QK^T/\sqrt{d_q})V$, introducing a quadratic computational and memory complexity of $O(n^2)$, where $n$ is the sequence length. Low-rank approximation (LRA) emerges as a strategic approach to improve efficiency by approximating high-dimensional matrices while preserving performance. These methods can reduce both time and space complexity to approximately $O(n)$, even when integrated into pretrained or newly trained models.

It is important to note that LRA does not inherently reduce model size, as the original weights $W_Q$, $W_K$, and $W_V$ remain stored. However, it significantly decreases computation time and memory usage during fine-tuning and inference, since the approximations operate post-input projection.

Several LRA methods have been proposed. Nyström-based approaches such as Nyströmformer and SOFT linearize attention via the Nyström method. Other linearization techniques, including Linformer and Performer, combine low-rank and sparse attention mechanisms to further improve approximation accuracy.

## 4. Conclusion

This review has summarized key models and developments in video generation, with particular attention to the evolution of Diffusion Models and their applications. In addition, several optimization strategies for Transformer-based models were studied, including quantization, knowledge distillation, pruning, and low-rank approximation. Under the guidance of my advisor, I began from survey papers, progressed through conference publications, and gradually focused on core subfields, thereby gaining a clear understanding of the literature review process. Throughout this work, I developed practical skills in academic writing and technical documentation, including the use of proper formatting, equation and figure preparation, and citation management. The experience gained from this research practice has deepened my understanding of video generation, provided valuable insights into model optimization, and laid a solid foundation for my future research endeavors.

## References


[1] Feiyang Chen, Ziqian Luo, Lisang Zhou, Xueting Pan, and Ying Jiang. Comprehensive survey of model compression and speed up for vision transformers. *arXiv preprint arXiv:2404.10407*, 2024. 6

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

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

[4] Chengxuan Li, Di Huang, Zeyu Lu, Yang Xiao, Qingqi Pei, and Lei Bai. A survey on long video generation: Challenges, methods, and prospects. *arXiv preprint arXiv:2403.16407*, 2024. 5, 6

[5] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024. 3

[6] Rui Sun, Yumin Zhang, Tejal Shah, Jiahao Sun, Shuoying Zhang, Wenqi Li, Haoran Duan, Bo Wei, and Rajiv Ranjan. From sora what we can see: A survey of text-to-video generation. *arXiv preprint arXiv:2405.10674*, 2024. 1, 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[8] Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. A survey on efficient training of transformers. *arXiv preprint arXiv:2302.01107*, 2023. 6