Goku: Flow Based Video Generative Foundation Models

Shoufa Chen[1*] Chongjian Ge[1*] Yuqi Zhang[2] Yida Zhang[2]
Fengda Zhu[2] Hao Yang[2] Hongxiang Hao[2] Hui Wu[2] Zhichao Lai[2] Yifei Hu[2]
Ting-Che Lin[2] Shilong Zhang[1] Fu Li[2] Chuan Li[2] Xing Wang[2] Yanghua Peng[2]
Peize Sun[1] Ping Luo[1] Yi Jiang[2] Zehuan Yuan[2] Bingyue Peng[2] Xiaobing Liu[2]

[1]The University of Hong Kong [2]Bytedance
[*] Equal Contribution
https://saiyan-world.github.io/goku/

Abstract

This paper introduces **Goku***, a state-of-the-art family of joint image-and-video generation models leveraging rectified flow Transformers to achieve industry-leading performance. We detail the foundational elements enabling high-quality visual generation, including the data curation pipeline, model architecture design, flow formulation, and advanced infrastructure for efficient and robust large-scale training. The Goku models demonstrate superior performance in both qualitative and quantitative evaluations, setting new benchmarks across major tasks. Specifically, Goku achieves 0.76 on GenEval and 83.65 on DPG-Bench for text-to-image generation, and 84.85 on VBench for text-to-video tasks. We believe that this work provides valuable insights and practical advancements for the research community in developing joint image-and-video generation models.*

1. Introduction

Video generation has garnered significant attention owing to its transformative potential across a wide range of applications, such media content creation [62], advertising [3, 93], video games [63, 77, 88], and world model simulators [1, 8, 31]. Benefiting from advanced generative algorithms [30, 34, 52, 53], scalable model architectures [59, 78], vast amounts of internet-sourced data [14, 45, 57], and ongoing expansion of computing capabilities [18–20], remarkable advancements have been achieved in the field of video generation [7, 8, 35, 36, 42, 44, 47, 49, 62, 69, 89].

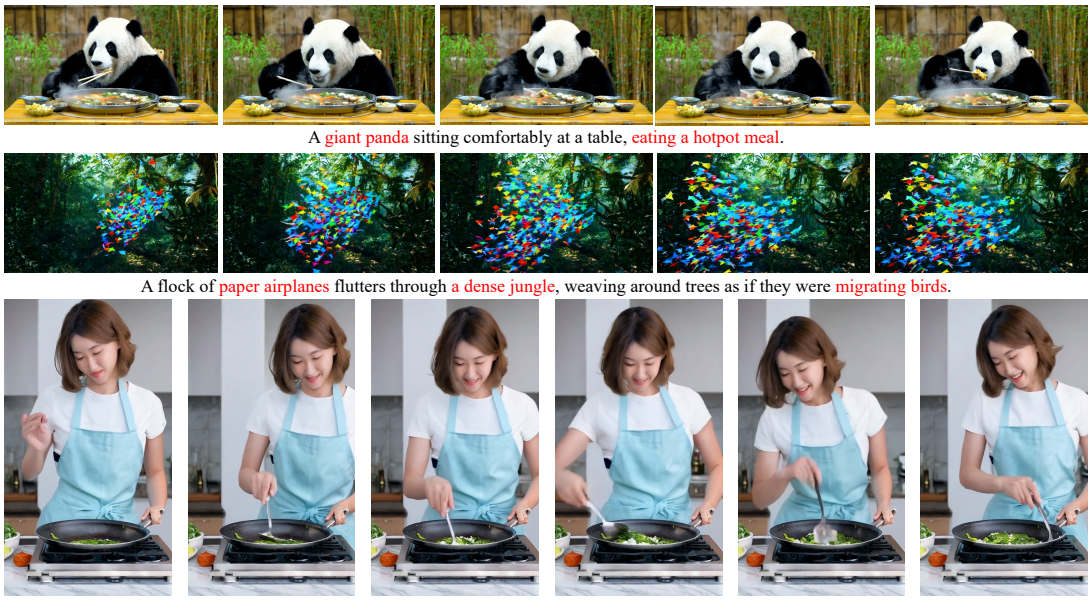
In this work, we introduce **Goku**, a family of rectified flow [52, 53] transformer models designed for joint image and video generation, paving the way toward industry-grade performance. Our approach emphasizes four key aspects: data curation, model architecture, flow formulation, and optimized training infrastructure—each refined for high-quality, large-scale video generation.

First, we present a comprehensive data processing pipeline designed to construct large-scale, high-quality image and video-text datasets. The pipeline integrates multiple advanced techniques, including video and image filtering based on aesthetic scores, OCR analysis, and subjective evaluations, to ensure exceptional visual and contextual quality. Furthermore, we employ multimodal large language models (MLLMs) [90] to generate dense and contextually aligned captions, which are subsequently refined using an additional Qwen2 [87] to enhance their accuracy and descriptive richness. As a result, we have curated a robust training dataset comprising approximately 36M video-text pairs and 160M image-text pairs, which are proven sufficient for training industry-level generative models.

Secondly, we take a pioneering step by applying rectified flow formulation [52] for joint image and video generation, implemented through the Goku model family, which comprises Transformer architectures with 2B and 8B parameters. At its core, the Goku framework employs a 3D joint image-video variational autoencoder (VAE) to compress image and video inputs into a shared latent space, facilitating unified representation. This shared latent space is coupled with a full-attention [78] mechanism, enabling seamless joint training of image and video. This architecture delivers high-quality, coherent outputs across both images and videos, establishing a unified framework for visual generation tasks.

Furthermore, to support the training of Goku at scale, we have developed a robust infrastructure tailored for large-scale model training. Our approach incorporates advanced parallelism strategies [41, 94] to manage memory efficiently during long-context training. Additionally, we employ ByteCheckpoint [79] for high-performance checkpointing and integrate fault-tolerant mechanisms from MegaS-

(a) Text-to-Image Samples



(b) Text-to-Video Samples

Figure 1. **Generated samples from Goku.** Key components are highlighted in **RED**.

cale [43] to ensure stability and scalability across large GPU clusters. These optimizations enable Goku to handle the computational and data challenges of generative modeling with exceptional efficiency and reliability.

We evaluate Goku on both text-to-image and text-to-video benchmarks to highlight its competitive advantages. For text-to-image generation, Goku-T2I demonstrates strong performance across multiple benchmarks, including T2I-CompBench [39], GenEval [28], and DPG-Bench [38], excelling in both visual quality and text-image alignment. In text-to-video benchmarks, Goku-T2V achieves state-of-the-art performance on the UCF-101 [71] zero-shot generation task. Additionally, Goku-T2V attains an impressive score of **84.85** on VBench [40], securing the top position on the leaderboard (as of 2025-01-25) and surpassing several leading commercial text-to-video models. Qualitative results, illustrated in Figure 1, further demonstrate the superior quality of the generated samples. These findings underscore Goku's effectiveness in videoi generation and its potential as a solution for both research and commercial applications.

2. Goku: Generative Flow Models

In this section, we present three core components of Goku, the image-video joint VAE [89], the Goku Transformer architecture, and the rectified flow formulation. These components are designed to work synergistically, forming a cohesive and scalable framework for joint image and video generation. During training, each raw video input $x \in \mathbb{R}^{T\times H\times W\times 3}$ (with images treated as a special case where $T = 1$) is encoded from the pixel space to a latent space using a 3D image-video joint VAE (Section 2.1). The encoded latents are then organized into mini-batches containing both video and image representations, facilitating the learning of a unified cross-modal representation. Subsequently, the rectified flow formulation (Section 2.3) is applied to these latents, leveraging a series of Transformer blocks (Section 2.2) to model complex temporal and spatial dependencies effectively.

2.1. Image-Video Joint VAE

Earlier research [25, 33, 65] demonstrates that diffusion and flow-based models can significantly improve efficiency and performance by modeling in latent space through a Variational Auto-Encoder (VAE) [25, 46]. Inspired by Sora [8], the open-source community has introduced 3D-VAE to explore spatio-temporal compression within latent spaces for video generation tasks [50, 89, 95]. To extend the advantages of latent space modeling across multiple media formats, including images and videos, we adopt a jointly trained Image-Video VAE [89] that handles both image and video data within a unified framework. Specifically, for videos, we apply a compression stride of $8 \times 8 \times 4$ across height, width, and temporal dimensions, respectively, while for images, the compression stride is set to 8×8 in spatial dimensions.

2.2. Transformer Architectures

The design of the Goku Transformer block builds upon GenTron [12], an extension of the class-conditioned diffusion transformer [59] for text-to-image/video tasks. It includes a self-attention module for capturing inter-token correlations, a cross-attention layer to integrate textual conditional embeddings (extracted via the Flan-T5 language model [16]), a feed-forward network (FFN) for feature projection, and a layer-wise adaLN-Zero block that incorporates timestep information to guide feature transformations. Additionally, we introduce several recent design enhancements to improve model performance and training stability, as detailed below.

Plain Full Attention. In Transformer-based video generative models, previous approaches [7, 12, 69, 85] typically combine temporal attention with spatial attention to extend text-to-image generation to video. While this method reduces computational cost, it is sub-optimal for modeling complex temporal motions, as highlighted in prior work [62, 89]. In Goku, we adopt full attention to model multi-modal tokens (image and video) within a unified network. Given the large number of video tokens remaining after VAE processing—particularly for high-frame-rate, long-duration videos—we leverage `FlashAttention` [21, 68] and sequence parallelism [51] to optimize both GPU memory usage and computational efficiency.

Model	Layer	Model Dim.	FFN Dim.	Attention Heads
Goku-1B	28	1152	4608	16
Goku-2B	28	1792	7168	28
Goku-8B	40	3072	12288	48

Table 1. **Architecture configurations for Goku Models.** Goku-1B model is only used for pilot experiments in Section 2.3

Patch n' Pack. To enable joint training on images and videos of varying aspect ratios and lengths, we follow the approach from NaViT [23], packing both modalities into a single minibatch along the sequence dimension. This method allows flexible mixing of training instances with different sequence lengths into a single batch, eliminating the need for data buckets [61].

3D RoPE Position Embedding. Rotary Position Embedding (RoPE) [72] has demonstrated effectiveness in LLMs by enabling greater sequence length flexibility and reducing inter-token dependencies as relative distances increase. During joint training, we apply 3D RoPE embeddings to image and video tokens. In our joint training framework, we extend 3D RoPE embeddings to image and video tokens, leveraging their extrapolation capability to accommodate varying resolutions. This adaptability makes RoPE particularly suited for handling diverse resolutions and video lengths. Furthermore, our empirical analysis revealed that RoPE converges faster than sinusoidal positional embeddings during transitions across different training stages

Q-K Normalization. Training large-scale Transformers can occasionally result in loss spikes, which may lead to model corruption, manifesting as severe artifacts or even pure noise in generated images or videos. To mitigate this issue, we incorporate query-key normalization [22] to stabilize the training process. Specifically, we apply RMSNorm [92] to each query-key feature prior to attention computation, ensuring smoother and more reliable training dynamics.

The overall Transformer model is constructed by stacking a sequence of blocks as described above. To address varying computational demands and performance requirements, we design three model variants, summarized in Table 1. The

Goku-1B model serves as a lightweight option for pilot experiments. The Goku-2B variant consists of 28 layers, each with a model dimension of 1792 and 28 attention heads, providing a balance between computational efficiency and expressive capacity. In contrast, the larger Goku-8B variant features 40 layers, a model dimension of 3072, and 48 attention heads, delivering superior modeling capacity aimed at achieving high generation quality.

2.3. Flow-based Training

Our flow-based formulation is rooted in the rectified flow (RF) algorithm [2, 52, 53], where a sample is progressively transformed from a prior distribution, such as a standard normal distribution, to the target data distribution. This transformation is achieved by defining the forward process as a series of linear interpolations between the prior and target distributions. Specifically, given a real data sample $\mathbf{x}_1$ from the target distribution and a noise sample $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$ from the prior distribution, a training example is constructed through linear interpolation:

$$\mathbf{x}_t = t \cdot \mathbf{x}_1 + (1 - t) \cdot \mathbf{x}_0, \tag{1}$$

where $t \in [0, 1]$ represents the interpolation coefficient. The model is trained to predict the velocity, defined as the time derivative of $\mathbf{x}_t$, $\mathbf{v}_t = \frac{d\mathbf{x}_t}{dt}$, which guides the transformation of intermediate samples $\mathbf{x}_t$ towards the real data $\mathbf{x}_1$ during inference. By establishing a direct, linear interpolation between data and noise, RF simplifies the modeling process, providing improved theoretical properties, conceptual clarity, and faster convergence across data distributions.

Goku takes a pioneering step by adopting a flow-based formulation for joint image-and-video generation. We conduct a pilot experiment to validate the rapid convergence of flow-based training by performing class-conditional generation with Goku-1B a model specifically designed for these proof-of-concept experiments, on ImageNet-1K (256×256) [24]. The model is configured with 28 layers, an attention dimension of 1152, and 16 attention heads. To evaluate performance, we compare key metrics, such as FID-50K and Inception Score (IS), for models trained using the denoising diffusion probabilistic model (DDPM) [34] and rectified flow. As shown in Table 2, RF demonstrates faster convergence than DDPM. For instance, Goku-1B (RF) achieves a lower FID-50K after 400k training steps compared to Goku-1B (DDPM), which requires 1000k steps to reach a similar level of performance.

2.4. Training Details

We propose a multi-stage training strategy. It starts with text-semantic pairing, advances to joint image-video training and cascaded resolutions, and concludes with modality-specific fine-tuning for optimal visual and temporal quality. Please refer to Appendix A for more details.

Loss	Steps	FID ↓	sFID ↓	IS ↑	Precision ↑	Recall ↑
DDPM	200k	3.0795	4.3498	226.4783	0.8387	0.5317
DDPM	400k	2.5231	4.3821	265.0612	0.8399	0.5591
DDPM	1000k	2.2568	4.4887	286.5601	0.8319	0.5849
Rectified Flow	200k	2.7472	4.6416	232.3090	0.8239	0.5590
Rectified Flow	400k	2.1572	4.5022	261.1203	0.8210	0.5871

Table 2. **Proof-of-concept experiments of class-conditional generation on ImageNet 256×256.** Rectified flow achieves faster convergency compared to DDPM.

2.5. Image-to-Video

To extend Goku for adapting an *image* as an additional condition for video generation, we employ a widely used strategy by using the first frame of each clip as the reference image [6, 29, 89]. The corresponding image tokens are broadcasted and concatenated with the paired noised video tokens along the channel dimension. To fully leverage the pretrained knowledge during fine-tuning, we introduce a single MLP layer for channel alignment, while preserving the rest of the model architecture identical to Goku-T2V.

3. Infrastructure Optimization

To achieve scalable and efficient training of Goku, we first adopt advanced parallelism strategies (Section 3.1), to handle the challenges of long-context, large-scale models. To further optimize memory usage and balance computation with communication, we implement fine-grained Activation Checkpointing (Section 3.2). Additionally, we integrate robust fault tolerance mechanisms from MegaScale, enabling automated fault detection and recovery with minimal disruption (Section 3.3). Finally, ByteCheckpoint is utilized to ensure efficient and scalable saving and loading of training states, supporting flexibility across diverse hardware configurations (Section 3.4). The details of these optimizations are introduced below.

3.1. Model Parallelism Strategies

The substantial model size and the exceptionally long sequence length (exceeding 220K tokens for the longest sequence) necessitate the adoption of multiple parallelism strategies to ensure efficient training. Specifically, we employ 3D parallelism to achieve scalability across three axes: input sequences, data, and model parameters.

Sequence-Parallelism (SP) [41, 48, 51] slices the input across the sequence dimension for independent layers (*e.g.*, LayerNorm) to eliminate redundant computations, reduce memory usage, and support padding for non-conforming input. We adopt *Ulysses* [41] as our implementation, which shards samples across the sequence parallel group from the start of the training loop. During attention computation, it

uses all-to-all communication to distribute query, key, and value shards, allowing each worker to process the full sequence but only a subset of attention heads. After parallel computation of attention heads, another all-to-all communication aggregates the results, recombining all heads and the sharded sequence dimension.

Fully Sharded Data Parallelism (FSDP) [94] partitions all parameters, gradients and optimizer states across the data parallel ranks. Instead of all-reduce in Distributed Data Parallelism, FSDP performs all-gather for parameters and reduce-scatter for gradients, enabling overlap with forward and backward computations to potentially reduce communication overhead. In our case, we adopt the `HYBRID_SHARD` strategy, which combines `FULL_SHARD` within a *shard group* and parameter replication across such groups, which effectively implements data parallelism (DP). This approach minimizes communication costs by limiting all-gather and reduce-scatter operations.

3.2. Activation Checkpointing

While the parallelism methods discussed in Section 3.1 provide significant memory savings and enable large-scaling training with long sequences, they inevitably introduce communication overhead among ranks, which can lead to suboptimal overall performance. To address this issue and better balance the computation and communication by maximizing their overlap in the profiling trace, we designed a fine-grained Activation Checkpointing (AC) [13] strategy. Specifically, we implemented selective activation checkpointing to minimize the number of layers requiring activation storage while maximizing GPU utilization.

3.3. Cluster Fault Tolerance

Scaling Goku training to large-scale GPU clusters inevitably introduces fault scenarios, which can reduce training efficiency. The likelihood of encountering failures increases with the number of nodes, as larger systems have a higher probability of at least one node failing. These disruptions can extend training time and increase costs. To enhance stability and efficiency at scale, we adopted fault tolerance techniques from MegaScale [43], including self-check diagnostics, multi-level monitoring, and fast restart/recovery mechanisms. These strategies effectively mitigate the impact of interruptions, enabling Goku to maintain robust performance in large-scale generative modeling tasks.

3.4. Saving and Loading Training Stages

Checkpointing training states—such as model parameters, exponential moving average (EMA) parameters, optimizer states, and random states—is crucial for training large-scale models, particularly given the increased likelihood of cluster faults. Reloading checkpointed states ensures reproducibility, which is essential for model reliability and debugging potential issues, including those caused by unintentional errors or malicious attacks.

To support scalable large-scale training, we adopt ByteCheckpoint [79] as our checkpointing solution. It not only enables parallel saving and loading of partitioned checkpoints with high I/O efficiency but also supports resharding distributed checkpoints. This flexibility allows seamless switching between different training scales, accommodating varying numbers of ranks and diverse storage backends. In our setup, checkpointing an 8B model across over thousands of GPUs blocks training for less than 4 seconds, which is negligible compared to the overall forward and backward computation time per iteration.

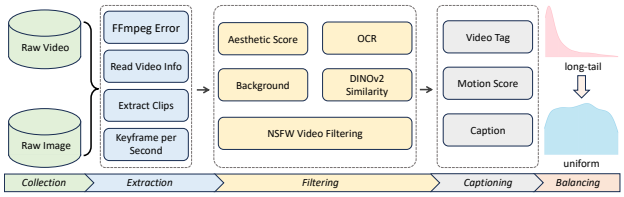


Figure 2. **The data curation pipeline in Goku.** Given a large volume of video/image data collected from Internet, we generate high-quality video/image-text pairs through a series of data filtering, captioning and balancing steps.

4. Data Curation Pipeline

Stage	Amount	Resolution	DINO-Sim.	Aesthetic	OCR	Motion
480p	36M	$\geq 480\times864$	$\geq$0.85	$\geq$ 4.3	<= 0.02	0.3 - 20.0
720p	24M	$\geq 720\times1280$	$\geq$0.90	$\geq$ 4.5	<= 0.01	0.5 - 15.0
1080p	7M	$\geq 1080\times1920$	$\geq$0.90	$\geq$ 4.5	<= 0.01	0.5 - 8.0

Table 3. **Overview of multi-stage training data**.This table summarizes the thresholds for each filtering criterion, including resolution, DINO similarity, aesthetic score, OCR text coverage, motion score, and the corresponding data quantities.

We unblock the data volume that is utilized for industry-grade video/image generation models. Our data curation pipeline, illustrated in Figure 2, consists of five main stages: (1) image and video collection, (2) video extraction and clipping, (3) image and video filtering, (4) captioning, and (5) data distribution balancing. We describe the details of data curation procedure below.

4.1. Data Overview

We collet raw image and video data from a variety of sources, including publicly available academic datasets, internet resources, and proprietary datasets obtained through partnerships with collaborating organizations. After rigorous filter-

ing, the final training dataset for Goku consists of approximately 160M image-text pairs and 36M video-text pairs, encompassing both publicly available datasets and internally curated proprietary datasets. The detailed composition of these resources is outlined as follows:

- **Text-to-Image Data.** Our text-to-image training dataset includes 100M public samples from LAION [67] and 60M high-quality, internal samples. We use public data for pre-training and internal data for fine-tuning.
- **Text-to-Video Data.** Our T2V training dataset includes 11M public clips and 25M in-house clips. The former include Panda-70M [14], InternVid [83], OpenVid-1M [57], and Pexels [50].

4.2. Data Preprocessing

We build a high-quality video dataset through preprocessing raw videos by standardizing duration, resolution, bitrate, and frame rate. Videos are segmented using PySceneDetect and refined with DINOv2. Clips undergo visual aesthetic evaluation, OCR-based text filtering, and RAFT-based motion filtering, with criteria adjusted by resolution. Semantic tags balance the dataset, emphasizing human-related content by adjusting representation across categories. More details are provided in Appendix C.

4.3. Captioning

Detailed captions are essential for enabling the model to generate text-aligned images/videos precisely. For images, we use InternVL2.0 [15] to generate dense captions for each sample. To caption video clips, we start with InternVL2.0 [15] for keyframe captions, followed by Tarsier2 [90] for video-wide captions. Note that the Tarsier2 model can inherently describe camera motion types (*e.g.*, *zoom in*, *pan right*) in videos, eliminating the need for a separate prediction model and simplifying the overall pipeline compared to previous work such as [62]. Next, we utilize Qwen2 [87] to merge the keyframe and video captions. Besides, we also empirically found that adding the motion score (calculated by RAFT [75]) to the captions improves motion control for video generation. This approach enables users to specify different motion scores in prompts to guide the model in generating videos with varied motion dynamics.

5. Experiments

5.1. Text-to-Image Results

we conduct a comprehensive quantitative evaluation of Goku-T2I on widely recognized image generation benchmarks, including GenEval [28], T2I-CompBench [39], and DPG-Bench [38]. Details of these benchmarks could be found in Appendix B. The results are summarized in Table 4.

Method	Text Enc.	GenEval	T2I-CompBench			DPG
		Overall	Color	Shape	Texture	Avg.
SDv1.5 [65]	CLIP ViT-L/14	0.43	0.3730	0.3646	0.4219	63.18
DALL-E 2 [64]	CLIP ViT-H/16	0.52	0.5750	0.5464	0.6374	-
SDv2.1 [65]	CLIP ViT-H/14	0.50	0.5694	0.4495	0.4982	-
SDX [61]	CLIP ViT-bigG	0.55	0.6369	0.5408	0.5637	74.65
PixArt-α [11]	Flan-T5-XXL	0.48	0.6886	0.5582	0.7044	71.11
DALL-E 3 [5]	Flan-T5-XXL	0.67†	0.8110†	0.6750†	0.8070†	83.50†
GenTron [12]	CLIP T5XXL	-	0.7674	0.5700	0.7150	-
SD3 [26]	Flan-T5-XXL	0.74	-	-	-	-
Show-o [86]	Phi-1.5	0.53	-	-	-	-
Transfusion [96]	-	0.63	-	-	-	-
Chameleon [54]	-	0.39	-	-	-	-
LlamaGen [73]	FLAN-T5 XL	0.32	-	-	-	-
Emu 3 [81]	-	0.66†	0.7913†	0.5846†	0.7422†	80.60
Goku-T2I (2B)	FLAN-T5 XL	0.70	0.7521	0.4832	0.6691	83.65
Goku-T2I (2B)†		0.76†	0.7561†	0.5759†	0.7071†	

Table 4. **Comparison with state-of-the-art models on image generation benchmarks.** We evaluate on GenEval [28]; T2I-CompBench [39] and DPG-Bench [38]. Following [81], we use † to indicate the result with prompt rewriting.

Performance on GenEval. To assess text-image alignment comprehensively, we employ the GenEval benchmark, which evaluates the correspondence between textual descriptions and visual content. Since Goku-T2I is primarily trained on dense generative captions, it exhibits a natural advantage when handling detailed prompts. To further explore this, we expand the original short prompts in GenEval with ChatGPT-4o, preserving their semantics while enhancing descriptive detail. As shown in Table 4, Goku-T2I achieves strong performance with the original short prompts, surpassing most state-of-the-art models. With the rewritten prompts, Goku-T2I attains the highest score (0.76), demonstrating its exceptional capability in aligning detailed textual descriptions with generated images.

Performance on T2I-CompBench. We further evaluate the alignment between generated images and textual conditions using the T2I-CompBench benchmark, which focuses on various object attributes such as color, shape, and texture. As illustrated in Table 4, Goku-T2I consistently outperforms several strong baselines, including PixArt-α [11], SDXL [61], and DALL-E 2 [56]. Notably, the inclusion of prompt rewriting leads to improved performance across all attributes, further highlighting Goku-T2I's robustness in text-image alignment.

Performance on DPG-Bench. While the aforementioned benchmarks primarily evaluate text-image alignment with short prompts, DPG-Bench is designed to test model performance on dense prompt following. This challenging benchmark includes 1,000 detailed prompts, providing a rigorous test of a model's ability to generate visually accurate outputs for complex textual inputs. As shown in the last column of

Method	Resolution	FVD (↓)	IS (↑)
CogVideo (Chinese) [37]	480×480	751.34	23.55
CogVideo (English) [37]	480×480	701.59	25.27
Make-A-Video [69]	256×256	367.23	33.00
VideoLDM [7]	-	550.61	33.45
LVDM [33]	256×256	372.00	-
MagicVideo [97]	-	655.00	-
PixelDance [91]	-	242.82	42.10
PYOCO [27]	-	355.19	47.76
Emu-Video [29]	256×256	317.10	42.7
SVD [6]	240×360	242.02	-
Goku-2B (ours)	256×256	246.17	45.77 ± 1.10
Goku-2B (ours)	240×360	254.47	46.64 ± 1.08
Goku-2B (ours)	128×128	217.24	42.30 ± 1.03

Table 5. **Zero-shot text-to-video performance on UCF-101**. We generate videos of different resolutions, including 256×256, 240×360, 128×128, for comprehensive comparisons.

(a) Model Scaling

(b) Joint Training

Figure 3. **Ablation Studies of Model Scaling and Joint Training.** Fig. (a) shows the comparison between Goku-T2V(2B) and Goku-T2V(8B). Fig. (b) shows the comparisons between whether joint training is adopted or not.

Table 4, Goku-T2I achieves the highest performance with an average score of 83.65, surpassing PixArt-α [11] (71.11), DALL-E 3 [5] (83.50), and EMU3 [81] (80.60). These results highlight Goku-T2I's superior ability to handle dense prompts and maintain high fidelity in text-image alignment.

5.2. Text-to-Video Results

Performance on UCF-101. We conduct experiments on UCF-101 [71] using zero-shot text-to-video setting. As UCF-101 only has *class* labels, we utilize an video-language model, Tarsier-34B [80], to generate detailed captions for all UCF-101 videos. These captions are then used to synthesize videos with Goku. Finally, we generated 13,320 videos at different resolutions with Goku-2B model for evaluation, including 256×256, 240×360 and 128×128. Following standard practice [70], we use the I3D model, pre-trained on Kinetics-400 [9], as the feature extractor. Based on the extracted features, we calculated Fréchet Video Distance (FVD) [76] to evaluate the fidelity of the generated videos. The results in Table 5 demonstrate that Goku consistently generates videos with lower FVD and higher IS. For instance, at a resolution of 128×128, the FVD of videos generated by Goku is 217.24, achieving state-of-the-art performance and highlighting significant advantages over other methods.

Performance on VBench. As presented in Table 6, we evaluate Goku-T2V against state-of-the-art models on VBench [40], a comprehensive benchmark designed to assess video generation quality across 16 dimensions. Goku-T2V achieves state-of-the-art overall performance on VBench, showcasing its ability to generate high-quality videos across diverse attributes and scenarios.

Among the key metrics, Goku-T2V demonstrates notable strength in human action representation, dynamic degree, and multiple object generation, reflecting its capacity for handling complex and diverse video content. Additionally, it achieves competitive results in appearance style, quality score, and semantic alignment, highlighting its balanced performance across multiple aspects.

For detailed results on all 16 evaluation dimensions, we refer readers to Table 8 in the Appendix. This comprehensive analysis underscores Goku-T2V's superiority in video generation compared to prior approaches.

5.3. Image-to-Video

We fine-tune Goku-I2V from the T2V initialization with 4.5M text-image-video triplets, sourced from diverse domains to ensure robust generalization. Despite the relatively small number of fine-tuning steps (10k), our model demonstrates remarkable efficiency in animating reference image while maintaining strong alignment with the accompanying text. As illustrated in Figure 4, the generated videos exhibit high visual quality and temporal coherence, effectively capturing the semantic nuances described in the text.

5.4. Image and Video Qualitative Visualizations

For intuitive comparisons, we conduct qualitative assessments and present sampled results in Figure 12. The evaluation includes open-source models, such as CogVideoX [89] and Open-Sora-Plan [95], alongside closed-source commercial products, including DreamMachine [55], Pika [60], Vidu [4], and Kling [49]. The results reveal that some commercial models struggle to generate critical video elements when handling complex prompts. For instance, models like Pika, DreamMachine, and Vidu (rows 3–5) fail to render the

Models	Human Action	Scene	Dynamic Degree	Multiple Objects	Appear. Style	Quality Score	Semantic Score	**Overall**
AnimateDiff-V2	92.60	50.19	40.83	36.88	22.42	82.90	69.75	80.27
VideoCrafter-2.0	95.00	55.29	42.50	40.66	**25.13**	82.20	73.42	80.44
OpenSora V1.2	85.80	42.47	47.22	58.41	23.89	80.71	73.30	79.23
Show-1	95.60	47.03	44.44	45.47	23.06	80.42	72.98	78.93
Gen-3	96.40	54.57	60.14	53.64	24.31	84.11	75.17	82.32
Pika-1.0	86.20	49.83	47.50	43.08	22.26	82.92	71.77	80.69
CogVideoX-5B	99.40	53.20	70.97	62.11	24.91	82.75	77.04	81.61
Kling	93.40	50.86	46.94	68.05	19.62	83.39	75.68	81.85
Mira	63.80	16.34	60.33	12.52	21.89	78.78	44.21	71.87
CausVid	**99.80**	56.58	**92.69**	72.15	24.27	**85.65**	78.75	84.27
Luma	96.40	**58.98**	44.26	**82.63**	24.66	83.47	**84.17**	83.61
HunyuanVideo	94.40	53.88	70.83	68.55	19.80	85.09	75.82	83.24
Goku (ours)	97.60	57.08	76.11	79.48	23.08	85.60	81.87	**84.85**

Table 6. **Comparison with leading T2V models on VBench.** Goku achieves state-of-the-art overall performance. Detailed results across all 16 evaluation dimensions are provided in Table 8 in the Appendix.

Figure 4. **Samples of Goku-I2V.** Reference images are presented in the leftmost columns. We omitted redundant information from the long prompts, displaying only the key details in each one. Key words are highlighted in **RED**.

skimming drone over water. While certain models succeed in generating the target drone, they often produce distorted subjects (rows 1–2) or static frames lacking motion consistency (row 6). In contrast, Goku-T2V (8B) demonstrates superior performance by accurately incorporating all details of the prompt, creating a coherent visual output with smooth motion. Additional comparisons are provided in the appendix for a more comprehensive evaluation. Furthermore, more video examples are available at the Goku homepage.

5.5. Ablation Studies

Model Scaling. We compared Goku-T2V models with 2B and 8B parameters. Results in Figure 3a indicate that model scaling helps mitigate the generation of distorted object structures, such as the arm in Figure 3a (row 1) and the wheel in Figure 3a (row 2). This aligns with findings observed in large multi-modality models.

Joint Training. We further examine the impact of joint image-and-video training. Starting from the same pretrained Goku-T2I (8B) weights, we fine-tuned Goku-T2V (8B) on 480p videos for an equal number of training steps, with and without joint image-and-video training. As shown in Figure 3b, Goku-T2V without joint training tends to generate low-quality video frames, while the model with joint training more consistently produces photorealistic frames.

6. Conclusion

In this work, we introduced Goku, a novel joint image-and-video generation model designed for industry-standard performance. By combining meticulous data curation and a robust model architecture, Goku effectively integrates image and video modalities to produce high-quality outputs. Empirical evaluations demonstrate Goku's superior capability in commercial-grade visual content generation.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[2] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

[3] Ivan Bacher, Hossein Javidnia, Soumyabrata Dev, Rahul Agrahari, Murhaf Hossari, Matthew Nicholson, Clare Conran, Jian Tang, Peng Song, David Corrigan, et al. An advert creation system for 3d product placements. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*, pages 224–239. Springer, 2021.

[4] Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

[5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[6] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[7] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[8] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators, 2024.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[10] Brandon Castellano. PySceneDetect, 2024.

[11] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alphaalpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[12] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6441–6451, 2024.

[13] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[14] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024.

[15] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024.

[16] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[17] Image Hash Contributors. Image hash, 2013.

[18] NVIDIA Corporation. Nvidia h100 tensor core gpu architecture, 2022.

[19] NVIDIA Corporation. Nvidia announces dgx gh200 ai supercomputer, 2023.

[20] NVIDIA Corporation. Nvidia h200 nvl pcie gpu accelerates ai and hpc applications, 2024.

[21] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

[22] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[23] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36, 2024.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[25] 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.

[26] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[27] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22930–22941, 2023.

[28] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[29] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023.

[30] 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.

[31] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[32] Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. Venhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024.

[33] 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*, 2(3):4, 2022.

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

[35] 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.

[36] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[37] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

[38] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[39] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

[40] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[41] Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv preprint arXiv:2309.14509*, 2023.

[42] Yatai Ji, Jiacheng Zhang, Jie Wu, Shilong Zhang, Shoufa Chen, Chongjian GE, Peize Sun, Weifeng Chen, Wenqi Shao, Xuefeng Xiao, et al. Prompt-a-video: Prompt your video diffusion model via preference-aligned llm. *arXiv preprint arXiv:2412.15156*, 2024.

[43] Ziheng Jiang, Haibin Lin, Yinmin Zhong, Qi Huang, Yangrui Chen, Zhi Zhang, Yanghua Peng, Xiang Li, Cong Xie, Shibiao Nong, et al. Megascale: Scaling large language model training to more than 10,000 gpus. *arXiv preprint arXiv:2402.15627*, 2024.

[44] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

[45] Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. *arXiv preprint arXiv:2407.06358*, 2024.

[46] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[47] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[48] Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023.

[49] Kuaishou. Kling ai. https://klingai.com/, 2024.

[50] PKU-Yuan Lab and Tuzhan AI etc. Open-sora-plan, 2024.

[51] Shenggui Li, Fuzhao Xue, Chaitanya Baranwal, Yongbin Li, and Yang You. Sequence parallelism: Long sequence training from system perspective. *arXiv preprint arXiv:2105.13120*, 2021.

[52] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[53] Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

[54] Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Luma. Luma ai. https://lumalabs.ai/dream-machine, 2024.

[56] Pamela Mishkin, Lama Ahmad, Miles Brundage, Gretchen Krueger, and Girish Sastry. Dall· e 2 preview-risks and limitations. *Noudettu*, 28(2022):3, 2022.

[57] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[58] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[59] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[60] pika. Pika ai. https://pika.art/try, 2024.

[61] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[62] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

[63] Julian Quevedo, Quinn McIntyre, Spruce Campbell, and Robert Wachen. Oasis: A universe in a transformer, 2024.

[64] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3, 2022.

[65] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[66] Runway. Gen-2: Generate novel videos with text, images or video clips. https://runwayml.com/research/gen-2/, 2023.

[67] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[68] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *arXiv preprint arXiv:2407.08608*, 2024.

[69] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023.

[70] 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.

[71] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. Technical report, Center for Research in Computer Vision, Orlando, FL 32816, USA, 2012. CRCV-TR-12-01.

[72] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[73] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

[74] Vchitect Team. Vchitect-2.0: Parallel transformer for scaling up video diffusion models. https://github.com/Vchitect/Vchitect-2.0, 2024.

[75] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[76] 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.

[77] Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. *arXiv preprint arXiv:2408.14837*, 2024.

[78] 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.

[79] Borui Wan, Mingji Han, Yiyao Sheng, Zhichao Lai, Mofan Zhang, Junda Zhang, Yanghua Peng, Haibin Lin, Xin Liu, and Chuan Wu. Bytecheckpoint: A unified checkpointing system for llm development. *arXiv preprint arXiv:2407.20143*, 2024.

[80] Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024.

[81] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need, 2024.

[82] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023.

[83] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[84] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[85] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023.

[86] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[87] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[88] Mingyu Yang, Junyou Li, Zhongbin Fang, Sheng Chen, Yangbin Yu, Qiang Fu, Wei Yang, and Deheng Ye. Playable game generation. *arXiv preprint arXiv:2412.00887*, 2024.

[89] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

[90] Liping Yuan, Jiawei Wang, Haomiao Sun, Yuchen Zhang, and Yuan Lin. Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding. *arXiv preprint arXiv:2501.07888*, 2025.

[91] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8850–8860, 2024.

[92] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

[93] Juan Zhang, Jiahao Chen, Cheng Wang, Zhiwang Yu, Tangquan Qi, Can Liu, and Di Wu. Virbo: Multimodal multilingual avatar video generation in digital marketing. *arXiv preprint arXiv:2403.11700*, 2024.

[94] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Proc. VLDB Endow.*, 16 (12):3848–3860, 2023.

[95] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all, 2024.

[96] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

[97] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.