CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer

Zhuoyi Yang, Jiayan Teng, Wendi Zheng Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, Jie Tang

Abstract

This paper presents CogVideoX, a large-scale text-to-video generation model based on a diffusion transformer, which can generate 10-second continuous videos aligned with a text prompt, with a frame rate of 16 fps and resolution of 768× 1360 pixels. Previous video generation models often had limited movement and short durations and it is difficult to generate videos with coherent narratives based on text. The authors propose several designs to address these issues. First, the authors propose a 3D Variational Autoencoder (VAE) to compress videos along both spatial and temporal dimensions, to improve both compression rate and video fidelity. Second, to improve text-video alignment, the authors propose an expert transformer with the expert adaptive LayerNorm to facilitate the deep fusion between the two modalities. Third, by employing a progressive training and multi-resolution frame-pack technique, CogVideoX is adept at producing coherent, long-duration, different shape videos characterized by significant motions. In addition, we develop an effective text-video data processing pipeline that includes various data preprocessing strategies and a video captioning method, which greatly contributes to the generation quality and semantic alignment. The results show that CogVideoX demonstrates state-of-the-art performance across both multiple machine metrics and human evaluations.

Keywords: Text-to-Video Generalization, Diffusion Model.

1 Introduction

The rapid development of text-to-video models has been phenomenal, driven by both the Transformer architecture [13] and diffusion model [2]. Early attempts to pretrain and scale Transformers to generate videos from text have shown great promise, such as CogVideo [3] and Phenaki [14]. Meanwhile, diffusion models have recently made exciting advancements in video generation [1, 10]. By using Transformers as the backbone of diffusion models, i.e., Diffusion Transformers (DiT) [7], text-to-video generation has reached a new milestone, as evidenced by the impressive Sora showcases.

Despite these rapid advancements in DiTs, it remains technically unclear how to achieve long-term consistent video generation with dynamic plots. For example, previous models had difficulty generating a video based on a prompt like ”a bolt of lightning splits a rock, and a person jumps out from inside the rock.”

In this work, the authors train and introduce CogVideoX, a set of large-scale diffusion transformer models designed for generating long-term, temporally consistent videos with rich motion semantics. The authors address the challenges mentioned above by developing a 3D Variational Autoencoder, an expert Transformer, a progressive training pipeline, and a video data filtering and captioning pipeline, respectively.

First, to efficiently consume high-dimension video data, the authors design and train a 3D causal VAE that compresses the video along both spatial and temporal dimensions. Compared to previous method(Blattmann et al., 2023) of fine-tuning 2D VAE, this strategy helps significantly reduce the sequence length and associated training compute and also helps prevent flicker in the generated videos, that is, ensuring continuity among frames.

Second, to improve the alignment between videos and texts, the authors propose an expert Transformer with expert adaptive LayerNorm to facilitate the fusion between the two modalities. To ensure the temporal consistency in video generation and capture largescale motions, we propose to use 3D full attention to comprehensively model the video along both temporal and spatial dimensions.

Third, as most video data available online lacks accurate textual descriptions, the authors develop a video captioning pipeline capable of accurately describing video content. This pipeline is used to generate new textual descriptions for all video training data, which significantly enhances CogVideoX's ability to grasp precise semantic understanding.

In addition, the authors adopt and design progressive training techniques, including multi-resolution frame pack and resolution progressive training, to further enhance the generation performance and stability of CogVideoX. Furthermore, the authors propose Explicit Uniform Sampling, which stablizes the training loss curve and accelerates convergence by setting different timestep sampling intervals on each data parallel rank.

To date, the authors have completed the CogVideoX training with two parameter sizes: 5 billion and 2 billion, respectively. Both machine and human evaluations suggest that CogVideoX-5B outperforms well-known public models and CogVideoX-2B is very competitive across most dimensions.

The contributions can be summarized as follows:

- The authors propose CogVideoX, a simple and scalable structure with a 3D causal VAE and an expert transformer, designed for generating coherent, long-duration, highaction videos. It can generate long videos with multiple aspect ratios, up to 768×1360 resolution, 10 seconds in length, at 16fps, without super-resolution or frame-interpolation.

- The authors evaluate CogVideoX through automated metric evaluation and human assessment, compared with openly-accessible top-performing text-to-video models. CogVideoX achieves state-of-the-art performance.

- The authors publicly release our 5B and 2B models, including text-to-video and image-tovideo versions, the first commercial-grade open-source video generation models. We hope it can advance the filed of video generation.

2 Method

The limitations of the Text-to-Video generation model include limited movement, short durations, and difficult to generate videos with narratives based on text.In order to tackle these problems, this paper proposes a 3D causal VAE to compress both spatial and temporal dimensions for the generation of long-term, temporally consistent videos. Reducing sequence length and the corresponding training computations to avoid flicker. Moreover, an expert Transformer with expert adaptive LayerNorm is proposed to facilitate the fusion between two modalities to improve text-video alignment. In this section, we introduce the details of CogVideoX.

2.1 3D Causal VAE

The structure of the 3D causal VAE is shown in Figure 1. To address the computational challenge of modeling video data, the authors propose the implementation of a video compression module utilizing 3D Variational Autoencoders [15]. The objective is to utilize three-dimensional convolutions to compress videos in both spatial and temporal dimensions. This can facilitate a superior compression ratio with significantly enhanced quality and continuity of video reconstruction.

Figure 1 (a) shows the architecture of the 3D VAE used in CogVideoX. It is composed of an encoder, a decoder, and a Kullback-Leibler (KL) regularizer for latent space regularization. The encoder and decoder consist of symmetrically arranged stages, respectively performing 2× downsampling and upsampling by the interleaving of ResNet block stacked stages. Some blocks perform spatial and temporal downsampling (upsampling), while others only perform spatial downsampling (upsampling), as shown in Figure 1 (a).

To preserve the causality of the video data within the temporal dimension, temporally causal convolution [15] is utilized, as illustrated in Figure 1 (b). The temporally causal convolution places all padding at the beginning of the convolutional space. This guarantees that future information does not affect present or past predictions.

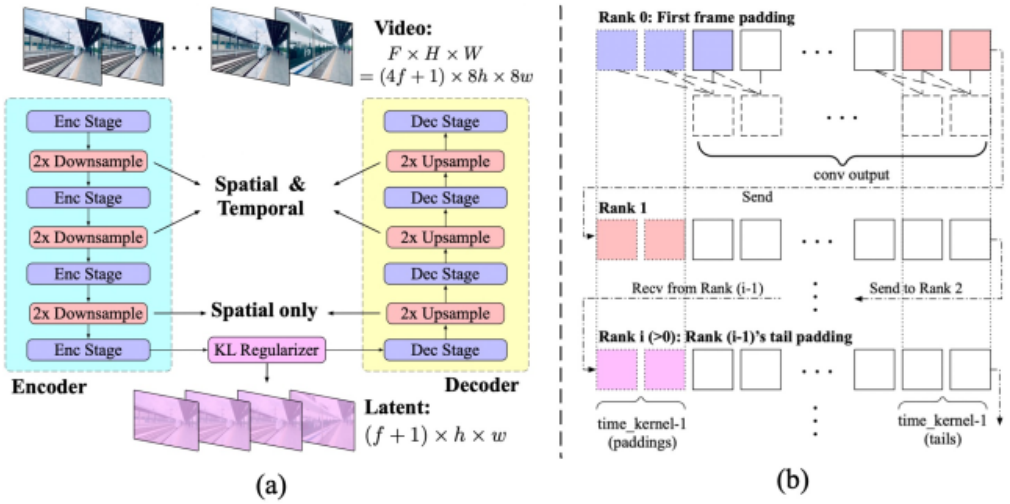

Figure 1. (a)The structure of the 3D VAE in CogVideoX. (b) The context parallel implementation on the temporally causal convolution.)

2.2 Expert Transformer

The expert transformer incorporates three design choices: patchify, 3D-ROPE, and the expert adaptive layernorm. This is helpful in enabling the integration of two modalities to enhance text-video alignment.

Patchify is employed to align the sequence length of video and image data. Patchify replicates the initial frame of videos or images at the beginning of the sequence. Aligning the sequence length and facilitating the joint training of videos and images.

3D-ROPE is used to encode the position information of video data. Rotary Position Embedding (RoPE) [11] is a relative positional encoding that captures inter-token relationships effectively in LLMs. It is utilized to process the image data. The authors extend ROPE to 3D-ROPE by employing a 3D coordinate to represent the latent in video tensors. The coordinate is subsequently concatenated along the channel dimension to achieve the final 3D-RoPE encoding.

Expert adaptive layernorm is used to align feature spaces across text and video data. The authors concatenate the embeddings of both text and video at the input stage to enhance the alignment of visual and semantic information. Nonetheless, the feature spaces of these two modalities exhibit significant differences, and their embeddings may possess distinct numerical scales. To enhance processing within the same sequence, the authors utilize the Expert Adaptive Layernorm to handle each modality independently. Specifically, the Vision Expert Adaptive Layernorm (Vison Expert AdaLN) and Text Expert Adaptive Layernorm (Text Expert AdaLN) process the hidden states of video and text data independently at the input stage. This strategy facilitates the alignment of feature spaces between two modalities while reducing supplementary parameters.

2.3 Overview of CogvideoX

Combining 3D causal VAE and expert transformer, the overall architecture of CogVideoX is presented in Figure 2. Given a pair of video and text input, 3D causal VAE is used to compress the video into the latent space, and the latents are then patchified and unfolded into a long sequence denoted as z_{vision}. Meanwhile, the text data is transformed into text embeddings z_{text} via a text encoder. In this paper, the text encoder employed is T5 [9]. Subsequently, z_{vision} and z_{text} are concatenated along the sequence dimension. The concatenated embeddings are then processed through a series of expert transformer blocks. Ultimately, the output is unpatchified to restore its original shape and decoded using a 3D causal VAE decoder to reconstruct the video.

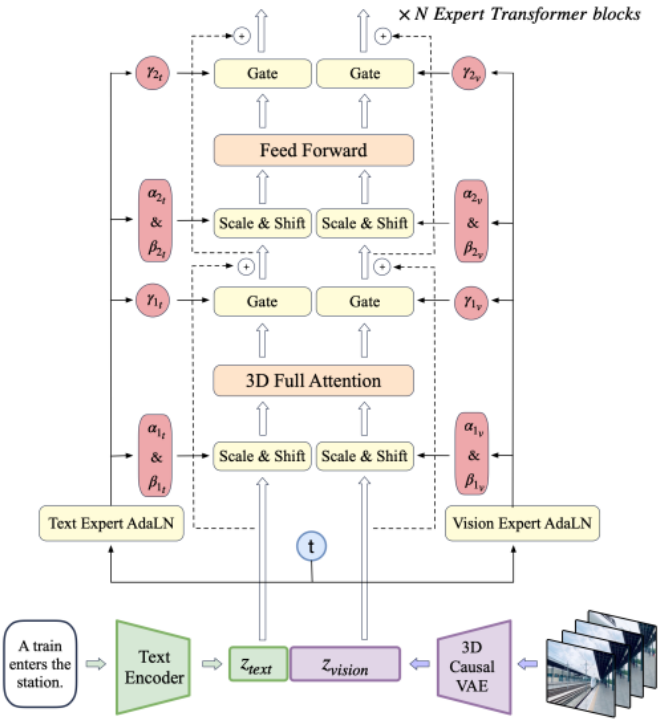

Figure 2. The overall architecture of CogvideoX

3 Implementation details

In this section, we will present the key codes of the implementation of the original paper.

We implemented CogVideoX in python. The source codes of encoding and decoding the video data are shown as follows:

```
def encode(
    self, x: torch.Tensor, return_dict: bool = True
) -> Union[AutoencoderKLOutput, Tuple[DiagonalGaussianDistribution]]:

    if self.use_slicing and x.shape[0] > 1:
        encoded_slices = [self._encode(x_slice) for x_slice in x.split(1)]
        h = torch.cat(encoded_slices)
    else:
        h = self._encode(x)

    posterior = DiagonalGaussianDistribution(h)

    if not return_dict:
        return (posterior,)
    return AutoencoderKLOutput(latent_dist=posterior)


def decode(self, z: torch.Tensor, return_dict: bool = True) -> Union[DecoderOutput, torch.Tensor]:

    if self.use_slicing and z.shape[0] > 1:
        decoded_slices = [self._decode(z_slice).sample for z_slice in z.split(1)]
        decoded = torch.cat(decoded_slices)
    else:
        decoded = self._decode(z).sample

    if not return_dict:
        return (decoded,)
    return DecoderOutput(sample=decoded)
```

Moreover, the implementation of the expert transformer is shown as follows:

```python
def forward(
    self,
    hidden_states: torch.Tensor,
    encoder_hidden_states: torch.Tensor,
    timestep: Union[int, float, torch.LongTensor],
    timestep_cond: Optional[torch.Tensor] = None,
    image_rotary_emb: Optional[Tuple[torch.Tensor, torch.Tensor]] = None,
    attention_kwargs: Optional[Dict[str, Any]] = None,
    return_dict: bool = True,
):

    batch_size, num_frames, channels, height, width = hidden_states.shape

    # 1. Time embedding
    timesteps = timestep
    t_emb = self.time_proj(timesteps)

    t_emb = t_emb.to(dtype=hidden_states.dtype)
    emb = self.time_embedding(t_emb, timestep_cond)

    # 2. Patch embedding
    hidden_states = self.patch_embed(encoder_hidden_states, hidden_states)
    hidden_states = self.embedding_dropout(hidden_states)

    text_seq_length = encoder_hidden_states.shape[1]
    encoder_hidden_states = hidden_states[:, :text_seq_length]
    hidden_states = hidden_states[:, text_seq_length:]
```

```python
    # 3. Transformer blocks
    for i, block in enumerate(self.transformer_blocks):
        if self.training and self.gradient_checkpointing:

            def create_custom_forward(module):
                def custom_forward(*inputs):
                    return module(*inputs)

                return custom_forward

            ckpt_kwargs: Dict[str, Any] = {"use_reentrant": False} if is_torch_version(">=", "1.11.0") else {}
            hidden_states, encoder_hidden_states = torch.utils.checkpoint.checkpoint(
                create_custom_forward(block),
                hidden_states,
                encoder_hidden_states,
                emb,
                image_rotary_emb,
                **ckpt_kwargs,
            )
        else:
            hidden_states, encoder_hidden_states = block(
                hidden_states=hidden_states,
                encoder_hidden_states=encoder_hidden_states,
                temb=emb,
                image_rotary_emb=image_rotary_emb,
            )

    # 4. Final block
    hidden_states = self.norm_out(hidden_states, temb=emb)
    hidden_states = self.proj_out(hidden_states)
    # 5. Unpatchify
    p = self.config.patch_size
    output = hidden_states.reshape(batch_size, num_frames, height // p, width // p, -1, p, p)
    output = output.permute(0, 1, 4, 2, 5, 3, 6).flatten(5, 6).flatten(3, 4)

    return Transformer2DModelOutput(sample=output)
```

3.1 Comparing with the released source codes

The source codes released with this paper do not include the implementation of the human perception evaluation metric on VBench [5]. To further evaluate the performance of CogVideoX, we have implemented this evaluation metric in Python. This addition allows us to better assess the alignment between the generated videos and human perceptual quality.

3.2 Experimental environment setup

In this section, we evaluate the text-to-video generation capabilities of the pretrained model CogVideoX-5B to evaluate the performance of CogVideoX. Several human perception metrics in Vbench [5] are employed, including human action, scene, dynamic degree, multiple objects, and appearance style.

VBench is a comprehensive benchmark for evaluating video generative models. It aims to decompose the overall ”video generation quality” into multiple well-defined evaluation dimensions. The evaluated videos are

generated using standard VBench prompt lists. Details of each evaluation metric are as follows.

Human action applying UMT [6] to evaluate the accuracy of human subjects in generated videos performing the actions specified in the text prompts. Scene using Tag2Text [4] to caption the generated scenes, and then verify their alignment with the scene descriptions in the text prompt. Dynamic degree utilising RAFT [12] to estimate the degree of dynamics in synthesized videos. Multiple objects detecting the success rate of generating all the objects specified in the text prompt within each video frame. Appearance style calculating the CLIP [8] feature similarity between synthesized frames and the corresponding style descriptions.

4 Results and analysis

Tabel 1 presents the performance comparison of CogVideoX and other models. The performance of the comparative models comes from the original paper. According to the experiment results, CogVideoX achieves the best performance in human action, dynamic degree, and appearance style. In Sence and appearance style, the performance gap between CogVideoX and the best method is small. These results indicate that CogVideoX outperforms prior methods for handling complex dynamic scenes and the quality of generative video.

Table 1. Evaluation results of CogVideoX

Models	Human Action	Scene	Dynamic Degree	Multiple Objects	Appearance Style
T2V-Turbo	95.2	**55.58**	49.17	54.65	24.42
AnimateDiff	92.6	50.19	40.83	36.88	22.42
VideoCrafter-2.0	95.0	55.29	42.50	40.66	**25.13**
OpenSora V1.2	85.8	42.47	47.22	58.41	23.89
Show-1	95.6	47.03	44.44	55.47	23.06
Gen-2	89.2	48.91	18.89	55.47	19.34
Pika	88.0	44.80	37.22	46.69	21.89
LaVie-2	96.4	49.59	31.11	64.88	25.09
CogVideoX	**96.75**	55.23	**61.89**	**70.15**	24.32

5 Conclusion and future work

In this report, we learned some knowledge of text-to-video generation and reproduced CogVideoX. We conducted experiments on human perception metrics to demonstrate the effectiveness of CogVideoX. Comprehensive experiments demonstrate that CogVideoX can handle complex dynamic scenes and generate high-quality video.

References

[1] 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*, 2022.

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

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

[4] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023.

[5] 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.

[6] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19948–19960, 2023.

[7] 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.

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[9] 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. *Journal of machine learning research*, 21(140):1–67, 2020.

[10] 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*, 2022.

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

[12] 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.

[13] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[14] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2022.

[15] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv*, 2023.