EXTENDING TEXT2VIDEO-ZERO FOR MULTI-CONTROLNET

by

BEN BACKEN

A THESIS

Presented to the Department of Computer Science
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science.

June 2023

An Abstract of the Thesis of

Ben Backen for the degree of Bachelor of Science
in the Department of Computer Science to be taken June 2023.

Title: Extending Text2Video-Zero for Multi-ControlNet

Approved: *Humphrey Shi, Ph.D.*
Primary Thesis Advisor

This research paper presents an extension to the Text2Video-Zero (T2V0) generative model, augmenting the synthesis of video from textual and video inputs. The project focuses on enhancing the functionality and accessibility of T2V0 by integrating Stable Diffusion's (SD) support for multiple ControlNets, implementing frame-wise masking for selective ControlNet application, and introducing memory optimizations to enable running the model on consumer-grade hardware. The paper also provides a high-level overview of SD, explores experimental features, and offers practical tips for generating videos using these tools. Additionally, we include a demonstration video showcasing T2V0 with Multi-ControlNet. The video highlights the early potential of text-to-video models for storytelling. Ultimately, the study strives to expand the capabilities and accessibility of T2V0, increasing users' control over their generated outputs while upholding the democratic principles of open-source AI.

Table of Contents

List of Figures

Introduction

Generative machine learning (ML) models are artificial intelligence systems that learn to produce novel data. In 2022, several research groups made enormous leaps in improving the quality of a particular type of generative model: text-to-image models. Text-to-image models take in a textual description and return an image depicting that text. The following figure is the output of a text-to-image model called Stable Diffusion (SD), given the prompt “frog holding a lightsaber.”

Frog Holding a Lightsaber

Outputs from Stable Diffusion given the prompt “frog holding a lightsaber”.

These models have immense potential as tools for visualizations and mockups. A creator can input an abstract idea and receive a unique image from which to draw inspiration. Pushing this concept further, generative models could serve as an alternative for actors and visual effects entirely.

In 2023, several research groups unveiled text-to-video models, which synthesize videos from textual input. Among these is Text2Video-Zero by Picsart AI Research (PAIR) (Khachatryan, Levon, et al.). T2V0 distinguishes itself from other text-to-video models by utilizing pre-trained SD weights (no additional training required) and by being completely open-source (Khachatryan, Levon, et al.). T2V0 is an enormous step towards democratizing AI.

The goal of our project is to extend the functionality of T2V0 and simultaneously increase its accessibility. In particular, we integrate SD's support for multiple ControlNets into T2V0, implement a method for applying different ControlNets to specific pixels in the video, and introduce memory optimizations to allow for the model to run on consumer-grade hardware (a GPU with 10GB of VRAM). These features increase the user's control over their model's outputs while preserving the democratic principles of PAIR's T2V0.

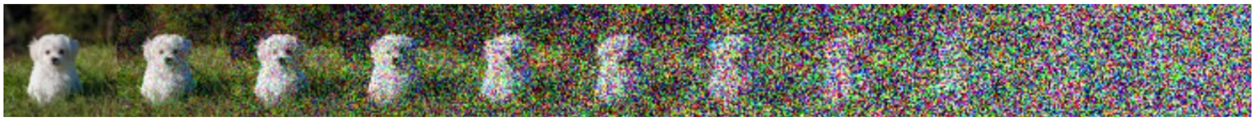
Diffusion Models

The basis of T2V0, SD, belongs to a family of ML models called "diffusion models." Before delving into SD, it is useful to gain a high-level understanding of how diffusion models generate images. The learning procedure entails two primary processes:

i. a forward diffusion process that adds Gaussian noise to an image over time until that image appears essentially random (pure noise)

ii. a reverse diffusion process where the model learns to gradually remove noise from the degraded image until returning to a recognizable image (Rogger, Niels, and Kashif Rasul).

The figure below demonstrates the forward diffusion process. We begin with a clear image of a dog. After each step, the image becomes increasingly corrupted until the photo seems to consist entirely of random pixels.



Forward Diffusion

Forward diffusion process on an image of a dog (Nichol, Alex, and Prafulla Dhariwal).

Intuitively, reversing the distortion process in a single step is infeasible; traversing from random pixels to a dog is an overwhelmingly complex task. However, if we follow the process incrementally in reverse, denoising the second image into the first image is reasonable. Similarly for the third image to the second image, and so on. The model must learn to predict the noise that was added to the image during a given step (i.e., it must learn the mean and covariance of the Gaussian distribution used to generate said noise). Then, it can gradually restore pure noise into an approximation of the original image.

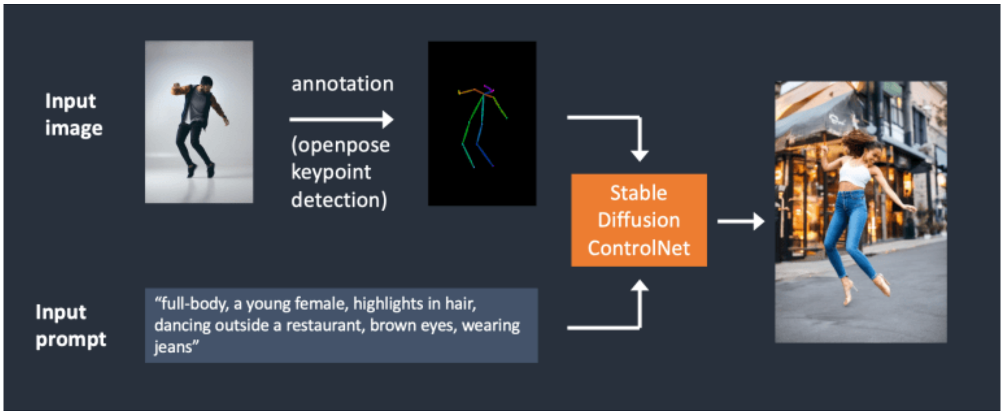
Once the model is proficient at this denoising process, one can construct a random image of Gaussian noise, feed it through the reverse diffusion process, and receive a unique image structurally similar to images in the training dataset. (For a more precise, mathematical description of diffusion models, see Sohl-Dickstein, Jascha, et al.)

Stable Diffusion

Reverse diffusion provides a method for generating images from noisy inputs. However, having a mechanism for guiding the output's contents would make the model far more versatile. Stable Diffusion (*Stable Diffusion 2*) is a latent diffusion model (Rombach, Robin, et al.) that utilizes Contrastive Language-Image Pre-training (CLIP) (Radford, Alec, et al.) to project text and images into the same embedding space. In other words, CLIP has learned a common representation between text and images. SD takes advantage of this ability to condition the denoising process towards the input text's "meaning" in the text-image embedding space. With CLIP-guided synthesis (Frans, Kevin, et al.), SD can transform textual descriptions into images.

ControlNet

ControlNet (Zhang, Lvmin, and Maneesh Agrawala) is a scheme that provides further control over the images generated by SD. Whereas standard, text-to-image SD supports conditioning on text (i.e., CLIP-guided synthesis), ControlNet allows for additional conditioning inputs. For example, the OpenPose network enables users to take an image of a person in a specific bodily position and generate a new image that retains the pose, but not necessarily the rest of the image's content. The following figure illustrates the process for OpenPose.



Input image

annotation

(openpose keypoint detection)

Stable Diffusion ControlNet

Input prompt

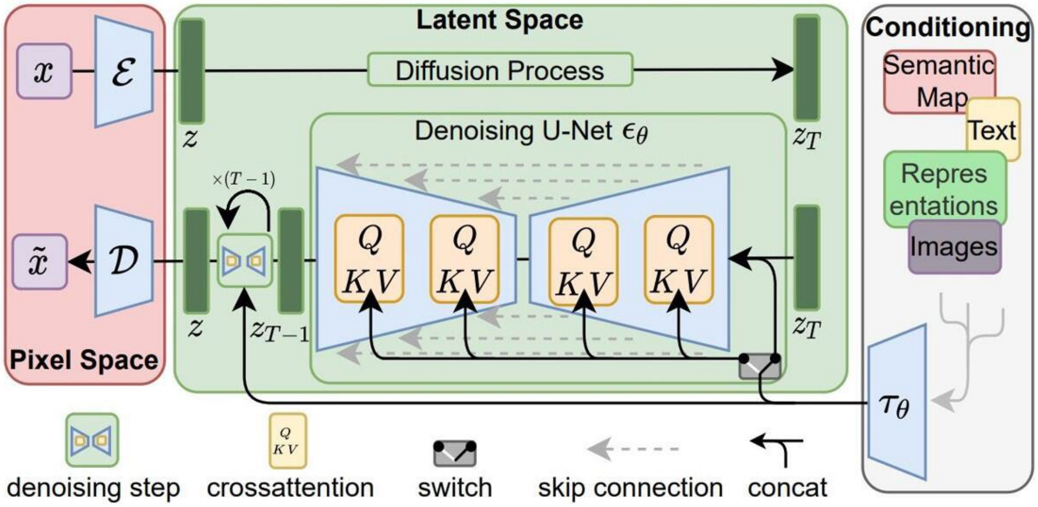
"full-body, a young female, highlights in hair, dancing outside a restaurant, brown eyes, wearing jeans"

OpenPose Diagram

An example of the OpenPose ControlNet being used to enforce the man's pose onto the woman in the new image (*ControlNet v1.1: A Complete Guide*).

We begin with an input image of a man whose feet are angled, knees bent, arms out, and head angled downward. This photo is passed through a separate model that has been trained to transform images of humans into skeleton-like annotations that represent the body's position. The armature image is then passed to the OpenPose ControlNet, which creates a conditional input for SD. Like with CLIP-guided synthesis, SD's output is pushed towards the "meaning" of this condition. In this case, the output is conditioned on the desired pose. The result is an image with an entirely different background and person, but the body position remains consistent.

ControlNet is an exceptionally powerful tool because of how it abstracts from SD. ControlNet does not directly modify SD's weights. To build a ControlNet network, one first creates a trainable copy of SD's U-Net encoder (for reference, a diagram of SD's architecture follows this paragraph). Each layer of the encoder-copy has a connection to its symmetric counterpart in the SD decoder. Then, during training, the entire SD model is frozen beside the encoder-copy. The encoder-copy learns to communicate between some domain (such as the skeleton-like pose images) and SD. This prevents the degradation of SD's text-to-image abilities while providing incredible control over outputs (Zhang, Lvmin, and Maneesh Agrawala).



Stable Diffusion Architecture

(Rombach, Robin, et al.)

Multiple ControlNets can be used in conjunction to perform different forms of control simultaneously. For example, one could use OpenPose and Canny edge detection to retain both the body position and general structure of the input image. The Diffusers SD pipeline supports

Multi-ControlNet functionality and that is what we will integrate into T2V0. Additionally, we will allow for the use of frame-wise masks to apply specific ControlNets to chosen sections of a video's frames.

Text2Video-Zero

T2V0 leverages SD's text-to-image capabilities for video generation. Unlike other text-to-video models such as Gen-2 (Esser, Patrick, et al.), T2V0 does not require expensive and resource-intensive training (Strubell, Emma, et al.) on enormous video datasets (Khachatryan, Levon, et al.). Instead, PAIR introduces two modifications to the SD pipeline: (i) enriching the latent codes of generated frames with motion dynamics and (ii) replacing SD's self-attention modules with cross-frame attention modules, which operate on the video's first frame for each subsequent frame (Khachatryan, Levon, et al.). For our Multi-ControlNet extension, only the latter mechanism is relevant.

When generating video, it is important that objects in the scene maintain a consistent appearance throughout. Because randomness is intrinsic to SD's forward diffusion process, video frames generated from the same prompt are unlikely to exhibit temporal consistency. To address this issue, PAIR implements cross-frame attention blocks.

In the original SD architecture, the U-Net contains self-attention modules and cross-attention modules. The cross-attention modules attend to conditions while the self-attention modules operate on the image being generated. In T2V0, PAIR swaps the self-attention modules with cross-frame attention blocks. These cross-frame attention blocks attend to the first frame for every subsequent frame. Consequently, "the appearance and structure of the objects and background as well as identities are carried over" (Khachatryan, Levon, et al.). T2V0 makes cohesive video generation possible on the basis of SD's text-to-image proficiency.

Methods

Multi-ControlNet

Our first task is integrating Multi-ControlNet into T2V0. Since the Diffusers SD pipeline supports Multi-ControlNet, these implementing these changes is straightforward. We add a new function "process_multi_controlnet" which takes in a list of ControlNet names and a list keyword-argument dictionaries for each net's preprocessor function, along with the usual text-to-video parameters. Each specified ControlNet is loaded with pre-trained weights and the input video is passed through the nets' preprocessing functions. The preprocessed videos are stored in a list, and we add cases to handle Multi-ControlNet input into PAIR's chunking optimization logic. The ControlNet models and preprocessed videos can then be passed as lists to the SD pipeline.

In order to utilize the Multi-ControlNet pipeline, we must update the Diffusers dependency from version "0.14.0" to version "0.16.1." This creates a bug in the "CrossFrameAttnProcessor" class relating to the attention block's "cross_attention_norm" attribute. Due to a change in the transformers module, we must first ensure that the attention object possesses said attribute before checking its value. After this slight adjustment, the pipeline runs as expected.

Multi-ControlNet Masking

To give users increased control over their outputs, we implement a technique for applying multiple ControlNets to different sections of video frames. For each ControlNet, the user must create a mask PNG file for each frame of the video. Pixels where the ControlNet should be applied are colored white (RGB 255, 255, 255) and all other pixels are colored black (RGB 0, 0, 0). It is important that these values are exact, as they are used directly for masking. After loading the ControlNets' preprocessed videos, the tensors are multiplied elementwise with the masks. Any pixels the user wishes for the ControlNet to operate on will be unchanged while pixels the user wants the ControlNet to ignore become 0.

This operation is valid for most ControlNet models. Since many ControlNets' preprocessing procedures use pure black as a background or "out of bounds" signal, masking excluded pixels with 0 amounts to ignoring the pixel. Later in the pipeline, all the ControlNet outputs are summed to produce a single conditioning tensor. Thus, if the masks do not overlap, the ControlNets' applied zones will not overlap. (Note that inpainting uses the value -1 to mark pixels where the ControlNet should focus, and therefore would not function properly with this masking method.)

Memory Optimization

Running large generative models can be a significant barrier for users with consumer-grade hardware. Fortunately, PAIR provides an excellent memory optimization option for T2V0 called "chunk_size". Adjusting the chunk size lets the user control how many video frames are processed at once. By setting this to the lowest possible value (2) and enabling token merging, T2V0 can run on a GPU with as little as 7 GB of VRAM (Khachatryan, Levon, et al.). Continuing PAIR's emphasis on accessibility, we submit further memory optimizations.

First, we add an option to the pipeline called "cuda_on_the_fly". This is an extension to the chunking option that stores all video frames and latent tensors on the CPU until they are ready to be processed by SD. Transferring tensors to the GPU on-the-fly has a negligible impact on inference time and eliminates video length as an obstacle for GPU memory. If a GPU can run inference with some chunk size, the possible video length is essentially only limited by RAM.

Additionally, when running T2V0 with ControlNets, the original input video never needs to exist on the GPU. All ControlNet preprocessing occurs on the CPU. Once preprocessing is complete, those tensors become the input for SD and the original video can be discarded.

These memory optimizations enable users to perform Multi-ControlNet inference on videos of arbitrary length using a GPU with 10 GB of VRAM.

Experimental Features

We implemented two extra features with limited success. The first is inpainting. The second is allowing the user to specify frame-wise prompts. Neither feature had a significant impact on the output, likely due to the overwhelming influence of the cross-frame attention blocks. Further experimentation is required.

Results

Demo

To demonstrate the capabilities of Text2Video-Zero with Multi-ControlNet, we created a simple animation in Blender and used the model to texture it. The video is available on YouTube here (https://www.youtube.com/watch?v=Y78TcPw9_js).

Video Generation Tips

Below is a short list of tips for generating videos with T2V0 and Multi-ControlNet:

1. All objects in the scene must be present in the first frame of the video. Due to the cross-frame attention blocks, new objects that enter into view will take on the colors and textures of objects generated in the first frame.

2. When using the depth and Canny ControlNets, low-poly objects tend to achieve the best results due to the low 512x512 resolution limit.

3. If objects overlap in 3D space, their edge and depth maps will be summed together and may result in the objects taking on colors and textures from elsewhere in the frame.

4. Do not specify too many colors in the prompt. SD does not handle this well and will begin coloring other pieces of the video incoherently.

5. There exist many models that have been fine-tuned by members of the online SD community. These models can often produce higher quality images and span a diverse range of visual styles. Models can be downloaded for free on websites such as Civitai. To use them with the Diffusers SD pipeline, you will need to convert the safetensors, CKPT, or PTH file into a Diffusers directory using one of the scripts here (https://github.com/huggingface/diffusers/tree/main/scripts). You can then call the SD pipeline's "from_pretrained" method to use the model.

6. Generating videos with many frames can take hours when using the most memory-efficient settings. Adding a hook in the inference function that saves the first frame to a file will give the user an opportunity to preview the quality of the entire video since the cross-frame attention blocks promote visual consistency based on frame 1.

Bibliography

ControlNet v1.1: A Complete Guide. 22 Feb. 2023, https://stable-diffusion-art.com/controlnet/.

Esser, Patrick, et al. Structure and Content-Guided Video Synthesis with Diffusion Models. arXiv, 6 Feb. 2023. arXiv.org, http://arxiv.org/abs/2302.03011.

Frans, Kevin, et al. "CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders." ArXiv.Org, 28 June 2021, https://doi.org/10.48550/arXiv.2106.14843.

Khachatryan, Levon, et al. Text2Video-Zero: Text-to-Image Diffusion Models Are Zero-Shot Video Generators. arXiv, 23 Mar. 2023. arXiv.org, https://doi.org/10.48550/arXiv.2303.13439.

Liu, Vivian, and Lydia B. Chilton. Design Guidelines for Prompt Engineering Text-to-Image Generative Models. arXiv, 18 Sept. 2021. arXiv.org, http://arxiv.org/abs/2109.06977.

Nichol, Alex, and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. Feb. 2021. arxiv.org, https://doi.org/10.48550/arXiv.2102.09672.

Radford, Alec, et al. "Learning Transferable Visual Models From Natural Language Supervision." ArXiv.Org, 26 Feb. 2021, https://doi.org/10.48550/arXiv.2103.00020.

Rogger, Niels, and Kashif Rasul. The Annotated Diffusion Model. https://huggingface.co/blog/annotated-diffusion.

Rombach, Robin, et al. "High-Resolution Image Synthesis with Latent Diffusion Models." ArXiv.Org, 20 Dec. 2021, https://doi.org/10.48550/arXiv.2112.10752.

Sohl-Dickstein, Jascha, et al. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. Mar. 2015. arxiv.org, https://doi.org/10.48550/arXiv.1503.03585.

Stable Diffusion 2. https://huggingface.co/docs/diffusers/api/pipelines/stable_diffusion_2.

Strubell, Emma, et al. *Energy and Policy Considerations for Deep Learning in NLP*. arXiv, 5 June 2019. *arXiv.org*, https://doi.org/10.48550/arXiv.1906.02243.

Zhang, Lvmin, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. arXiv, 10 Feb. 2023. arXiv.org, http://arxiv.org/abs/2302.05543.