Show Me What and Tell Me How: Video Synthesis via Multimodal Conditioning

Ligong Han\textsuperscript{1,2*} Jian Ren\textsuperscript{1} Hsin-Ying Lee\textsuperscript{1} Francesco Barbieri\textsuperscript{1}
Kyle Olszewski\textsuperscript{1} Shervin Minaee\textsuperscript{1} Dimitris Metaxas\textsuperscript{2} Sergey Tulyakov\textsuperscript{1}
\textsuperscript{1}Snap Inc. \textsuperscript{2}Rutgers University

Abstract

Most methods for conditional video synthesis use a single modality as the condition. This comes with major limitations. For example, it is problematic for a model conditioned on an image to generate a specific motion trajectory desired by the user since there is no means to provide motion information. Conversely, language information can describe the desired motion, while not precisely defining the content of the video. This work presents a multimodal video generation framework that benefits from text and images provided jointly or separately. We leverage the recent progress in quantized representations for videos and apply a bidirectional transformer with multiple modalities as inputs to predict a discrete video representation. To improve video quality and consistency, we propose a new video token trained with self-learning and an improved mask-prediction algorithm for sampling video tokens. We introduce text augmentation to improve the robustness of the textual representation and diversity of generated videos. Our framework can incorporate various visual modalities, such as segmentation masks, drawings, and partially occluded images. It can generate much longer sequences than the one used for training. In addition, our model can extract visual information as suggested by the text prompt, e.g., “an object in image one is moving northeast”, and generate corresponding videos. We run evaluations on three public datasets and a newly collected dataset labeled with facial attributes, achieving state-of-the-art generation results on all four.

1. Introduction

Generic video synthesis methods generate videos by sampling from a random distribution \cite{56, 57}. To get more control over the generated content, conditional video synthesis works utilize input signals, such as images \cite{9, 19}, text or language \cite{7, 30}, and action classes \cite{62}. This enables synthesized videos containing the desired objects as specified by visual information or desired actions as specified by textual information.

Existing works on conditional video generation use only one of the possible control signals as inputs \cite{8, 30}. This limits the flexibility and quality of the generative process. For example, given a screenplay we could potentially generate several movies, depending on the decisions of the director, set designer, and visual effect artist. In a similar way, a video generation model conditioned with a text prompt should be primed with different visual inputs. Additionally, a generative video model conditioned on a given image should be able to learn to generate various plausible videos, which can be defined from various natural language instructions. For example, to generate object-centric videos with objects moving \cite{70}, the motion can be easily defined through a text prompt, e.g., “moving in a zig-zag way,” while the objects can be defined by visual inputs. Thus, an interesting yet challenging question arises: Can we learn a video generation model that can support such behavior?

We tackle the question in this work and propose a new video synthesis model supporting diverse, multimodal conditioning signals. Our method consists of two phases. The first phase obtains discrete representations from images. We employ an autoencoder with a quantized bottleneck, inspired by the recent success of two-stage image generation using quantized feature representations \cite{21, 38, 47, 76}. The second phase learns to generate video representations that are conditioned on the input modalities, which can then be decoded into videos using the decoder from the first stage. We leverage a bidirectional transformer, i.e., BERT \cite{17}, trained with a masked sequence modeling task, that uses tokens from multimodal samples and predicts the latent representation for videos. Building such a framework requires solving several challenging problems. First, video consistency is a common problem among video generation methods. Second, it is necessary to ensure that the correct textual information is learned. Third, training a transformer for image synthesis is computationally demanding \cite{12}, an issue that is even more severe in the time domain, as a longer sequence of tokens needs to be learned. To solve these challenges, we propose the following contributions:

\footnote{*Work done during an internship at Snap Inc.}
\footnote{\textsuperscript{1}Code: https://github.com/snap-research/MMVID and Webpage.}
• We introduce a bidirectional transformer with several new techniques to improve video generation: For training, we propose the video token VID, which is trained via self-learning and video attention, to model temporal consistency: For inference, we improve mask-predict to generate videos with improved quality.
• We introduce text augmentation, including text dropout and pretrained language models for extracting textual embeddings, to generate diverse videos that are correlated with the provided text.
• We explore long sequence synthesis with the transformer model to generate sequences with lengths that are much longer than the one used for training (Fig. 5).

We name our framework MMVID and show that a MultiModal VIDEo generator can enable various applications. The user can show what to generate using visual modalities and tell how to generate with language. We explore two settings for multimodal video generation. The first involves independent multimodalities, such that there is no relationship between textual and visual controls (Fig. 3a and Fig. 4). The second one targets dependent multimodal generation, where we use text to obtain certain attributes from given visual controls (Fig. 3b and Fig. 4). The latter case allows for more potential applications, in which language is not able to accurately describe certain image content that the user seeks to generate, but images can efficiently define such content. We also show our model can use diverse visual information, including segmentation masks, drawings, and partially observed images (Fig. 4).

To validate our approach extensively, we conduct experiments on four datasets. In addition to three public datasets, we collect a new dataset, named Multimodal VoxCeleb, that includes 19,522 videos from VoxCeleb [37] with 36 manually labeled facial attributes.

2. Related Works

Video Generation. For simplicity, previous works on video generation can be categorized into unconditional and conditional generation, where most of them apply similar training strategies: adversarial training with image and video discriminators [14, 24, 50]. Research on unconditional video generation studies how to synthesize diverse videos with input latent content or motion noise [1, 28, 49, 57, 62, 66, 75]. Recent efforts in this direction have achieved high resolution and high quality generation results for images and videos [14, 29, 56]. On the other hand, conditional video generation utilizes given visual or textual information for video synthesis [9, 19, 39, 48, 54, 64, 67]. For example, the task of video prediction uses the provided first image or a few images to generate a sequence of frames [3–6, 11, 15, 16, 26, 60, 61, 63]. Similarly, text-to-video generation applies the conditional signal from text, captions, or natural language descriptions [31, 36, 40]. TFGAN [7] proposes a multi-scale text-filter conditioning scheme for discriminators. TiVGAN [30] proposes to generate a single image from text and synthesizes consecutive frames through further stages. In this work, we study conditional video synthesis. However, we differ from existing methods since we address a more challenging problem: multimodal video generation. Instead of using a single modality, such as textual guidance, we show how multiple modalities can be input within a single framework for video generation. With multimodal controls, i.e., textual and visual inputs, we further enhance two settings for video generation: independent and dependent multimodal inputs, in which various applications can be developed.

Transformers for Video Generation. Transformer-based networks have shown promising and often superior performance not only in natural languages processing tasks [10, 43, 59], but also in computer vision related efforts [12, 20, 27, 41, 42]. Recent works provide promising results on conditional image generation [13, 21], text-to-image generation [18, 45], video generation [44, 68, 70, 72], and text-to-video synthesis [69] using transformers. Unlike existing transformer-based video generation works that focus on autoregressive training, we apply a non-autoregressive generation pipeline with a bidirectional transformer [22, 23, 25, 35, 65]. Our work is inspired by M6-UFC [76], which utilizes the non-autoregressive training for multimodal image generation and produces more diverse image generation with higher quality. Building upon M6-UFC, we further introduce training techniques for multimodal video synthesis.

3. Methods

Our framework for multimodal video generation is a two-stage image generation method. It uses discrete feature representations [21, 38, 47, 76]. During the first stage we train an autoencoder (with encoder $E$ and decoder $D$) that has the same architecture as the one from VQGAN [21] to obtain a quantized representation for images. Given a real video clip defined as $v = \{x_1, x_2, \cdots, x_T\}$ with $x_t \in \mathbb{R}^{H \times W \times 3}$, we get a quantized representation of the video defined as $z = \{z_1, z_2, \cdots, z_T\}$, where $z_t = q(E(x_t)) \in \mathbb{N}^{h \times w}$. $q(\cdot)$ denotes the quantization operation and $\mathbb{N}$ indicates a set of positive integers.

During the second stage we learn a bidirectional transformer for modeling the correlation between multimodal controls and the learned vector quantization representation of a video. Specifically, we concatenate the tokens from the multimodal inputs and the target video as a sequence to train the transformer. Tensors obtained from an image and video must be vectorized for concatenation. We do so using the reshape operation (Reshape). Therefore, we have a video tensor $z$ reshaped into a single-index tensor as $\text{Reshape}(z) = [z^{(1)}, \cdots, z^{(hwT)}]$. For simplicity
of notation, we define $z \equiv \text{Reshape}(\mathbf{z})$. To train the non-autoregressive transformer (BERT) on video tokens, we employ three tasks: Masked Sequence Modeling (MSM), RELevance estimation (REL), and VIDeo consistency estimation (VID). During inference, samples are generated via an iterative algorithm based on mask-predict [23], which is simulated by the MSM task during training. The REL and VID tasks regularize the model to synthesize videos that are relevant to the multimodal signals and are temporally consistent. In the following two sections we present each task.

3.1. Masked Sequence Modeling with Relevance

**Masked Sequence Modeling.** The MSM task is similar to a conditional masked language model [23]. It is essential for the non-autoregressive model to learn bidirectional representations and enables parallel generation (mask-predict). Inspired by M6-UFC [76] and VIMPAC [55], we consider five masking strategies: (I) i.i.d. masking, i.e., randomly masking video tokens according to a Bernoulli distribution; (II) masking all tokens; (III) block masking [55], which masks continuous tokens inside spatio-temporal blocks; (IV) the negation of block masking, which preserves the spatio-temporal block and masks the rest of the tokens; (V) randomly keeping some frames (optional). Strategies I and II are designed to simulate mask-predict sampling (the strategy chosen most of the time). Strategy II helps the model learn to generate from a fully masked sequence in the first step of mask-predict. Strategies III - V can be used as Preservation Control (PC) for preservation tasks, which enable the use of partial images as input (Figs. 3a, 4) and performing long sequence generation (Fig. 5). The MSM task minimizes the softmax cross-entropy loss $\mathcal{L}_{\text{MSM}}$ as follows:

$$\mathcal{L}_{\text{MSM}} = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log P(z^{(1)} | z^m, c),$$

where $\mathcal{M}$ is the masking indices, $z^m$ is the masked sequence, and $c$ denotes the control sequence.

**Relevance Estimation.** To encourage the transformer to learn the correlation between multimodal inputs and target videos, we prepend a special token REL that is similar to the one used in M6-UFC [76] to the whole sequence, and learn a binary classifier to classify positive and negative sequences. The positive sequence is the same as the sequence used in the MSM task so that we can reuse the same transformer in the forward pass. The negative sequence is constructed by swapping the condition signals along the batch dimension. This swapping does not guarantee constructing strictly negative samples. Nevertheless, we still find it is good enough to make the model learn relevance in practice. The loss function $\mathcal{L}_{\text{REL}}$ for the relevance task is given by:

$$\mathcal{L}_{\text{REL}} = -\log P(1 | z^m, c) - \log P(0 | z^m, \bar{c}),$$

where $\bar{c}$ denotes the swapped control sequence.

3.2. Video Consistency Estimation

To further regularize the model to generate temporally consistent videos, we introduce the video consistency estimation task. Similar to REL, we use a special token VID to classify positive and negative sequences.
**Video Attention.** The VID task focuses on video token sequences. Thus, we place the VID token between the control and target sequences. We apply a mask to BERT to blind the scope of the VID token from the control signals so it only calculates attention from the tokens of the target videos. The positive sequence is the same one used in MSM and REL tasks. The negative sequence is obtained by performing negative augmentation on videos to construct samples that do not have temporally consistent motion or content.

**Negative Video Augmentation.** We employ four strategies to augment negative video sequences: (I) frame swaping – a random frame is replaced by using a frame from another video; (II) frame shuffling – frames within a sequence are shuffled; (III) color jittering – randomly changing the color of one frame; (IV) affine transform – randomly applying an affine transformation on one frame. All augmentations are performed in image space. With \( \tilde{z} \) denoting the video sequence after augmentation, the loss \( L_{VID} \) for the VID task is:

\[
L_{VID} = -\log P(1|z^m, c) - \log P(0|\tilde{z}^m, c). \tag{3}
\]

Overall, the full objective is \( L = \lambda_{MM}L_{MM} + \lambda_{REL}L_{REL} + \lambda_{VID}L_{VID} \), where \( \lambda \) balances the losses.

### 3.3. Improved Mask-Predict for Video Generation

We employ mask-predict [23] during inference, which iteratively remasks and repredicts low-confidence tokens by starting from a fully-masked sequence. We chose it because it can be used with our bidirectional transformer, as the length of the target sequence is fixed. In addition, mask predict provides several benefits. First, it allows efficient parallel sampling of tokens in a sequence. Second, the unrolling iterations from mask-predict enable direct optimization on synthesized samples, which can reduce exposure bias [46]. Third, information comes from both directions, which makes the generated videos more consistent.

We build our sampling algorithm based on the original mask-predict [23] with two improvements: (I) noise-anneling multinomial sampling, i.e., adding noise during remasking; (II) a new scheme for mask annealing, i.e., using a piecewise linear annealing scheme to prevent the generated motion from being washed out after too many steps of mask-predict. We also apply a beam search from M6-UFC [76]. In Alg. 1, the transformer (BERT) takes input tokens \( z_{in} \) and outputs score \( s \) and the logits \( \tilde{p} \) for all target tokens. At each mask-predict iteration, we sample tokens with SampleToken that returns a predicted token \( z_{out} \) and a vector \( y \) containing its probabilities (unnormalized). SampleToken also accepts a scalar \( \sigma \) that indicates the noise level to be added during the token sampling process. SampleMask \((y, m, N - n)\) remasks \( n \) tokens from a total of \( N \) tokens according to the multinomial defined by the normalized \( y \), while ensuring tokens with \( m = 1 \) are always preserved. \( z_0 \) denotes the fully-masked sequence. The functions SampleToken, SampleMask and the schedules of \( \sigma^{(i)} \) and \( \lambda^{(i)} \) are shown in Appendix.

**Algorithm 1 Improved Mask-Predict for Video Generation**

**Require:** Initial PC mask \( m_{PC} \) and initial token \( z_{in} \).

1. \( \tilde{p}, s \leftarrow \text{BERT}(z_{in}) \)
2. \( z_{out}, y \leftarrow \text{SampleToken}(\tilde{p}, \sigma^{(1)}) \)
3. \( z_{out} \leftarrow m_{PC} \odot z_{in} + (1 - m_{PC}) \odot z_{out} \quad \triangleright \text{PC} \)
4. for \( i \in \{2, ..., L\} \) do \quad \triangleright \text{main loop}
   5. for \( b \in \{1, ..., B\} \) do \quad \triangleright \text{beam search}
      6. \( m^b \leftarrow \text{SampleMask}(y, m_{PC}, N - n^{(i)}) \)
      7. \( z_{in}^b \leftarrow m^b \odot z_{out} + (1 - m^b) \odot z_\phi \quad \triangleright \text{remask} \)
      8. \( \tilde{p}^b, s^b \leftarrow \text{BERT}(z_{in}^b) \quad \triangleright \text{repredict} \)
   9. end for
10. \( b^* \leftarrow \text{arg max}_b(s^b) \)
11. \( z_{out} \leftarrow \text{SampleToken}(\tilde{p}^{b*}, \sigma^{(i)}) \)
12. end for
13. return \( z_{out} \)

### 3.4. Text Augmentation

We study two augmentation methods. First, we randomly drop sentences from the input text to avoid memorizing certain word combinations. Second, we apply a fixed pretrained language model, i.e., RoBERTa [33], rather than learning text token embeddings in a lookup table from scratch, to let the model be more robust for input textual information. The features of text tokens are obtained from an additional multilayer perceptron (MLP) appended after the language model that matches the vector dimension with BERT. The features are converted to a weighted sum to get the final embedding of the input text. With the language model, the video generation framework is more robust for out-of-distribution text prompts. When using the tokenizer from an existing work [42], we observed that it might not properly handle synonyms without a common root (Fig. 6).

### 3.5. Long Sequence Generation

Due to the inherent preservation control mechanism during training (strategy V in the MSM task), we can generate sequences with many more frames than the model is trained with via interpolation or extrapolation. **Interpolation** is conducted by generating the intermediate frames between the given frames. As illustrated by Fig. 1 (d), we place \( z_1 \) and \( z_2 \) at the positions of frames 1 and 3 to serve as preservation controls, i.e., they are kept the same during mask-predict iterations, and we can interpolate a frame between them. **Extrapolation** is similar to interpolation, except we condition the model on previous frames to generate the next frames. As illustrated in Fig. 1 (c), this process can be iterated a number of times to generate longer videos.
4. Experiments

Datasets. We show experiments on the following datasets.

- **Shapes** is proposed by TFGAN [8] for text-to-video generation. Each video shows one object (a geometric shape with specified color and size) displayed in a textured moving background. The motion of an object is described by a text and the background is moving in a random way. There are 30K videos with size 64 × 64.

- **MUG** [2] contains 52 actors performing 6 different facial expressions. We also provide gender labels for the actors. For a fair comparison, we obtain text descriptions by following TiVGAN [30]. We run experiments on 1039 videos with resolution 128 × 128.

- **iPER** [32] consists of 206 videos of 30 subjects wearing different clothes performing an A-pose and random actions. Experiments are conducted with size 128 × 128.

- **Multimodal VoxCeleb** is a new dataset for multimodal video generation. We first obtain 19,522 videos from VoxCeleb [37] after performing pre-processing [53]. Second, we manually label 36 facial attributes described in CelebA [34] for each video. Third, we use a probabilistic context-free grammar to generate language descriptions [71]. Finally, we run APDrawingGAN [73] to get artistic portrait drawings and utilize face-parsing [74] to produce segmentation masks.

Baseline Methods. We run TFGAN [8] on Shapes, MUG, and Multimodal VoxCeleb datasets for comparison of text-to-video synthesis. We also compare our approach with TiVGAN [30] on MUG. Additionally, we unify the autoregressive transformer of DALL-E [45] and the autoencoder from VQGAN (the same one used in our method) in a progressive transformer of DALL-E [45] and the autoencoder from VQGAN (the same one used in our method) in a multimodal video generative model. We name the strong baseline as AutoRegressive Transformer for Video generation (ART-V) and compare it with our bidirectional transformer for predicting video tokens. We train ART-V with the next-token-prediction objective on concatenated token sequences obtained from input controls and target videos.

Evaluation Metrics. We follow the metrics from existing works on Shapes and MUG to get a fair comparison. Specifically, we compute classification accuracy on Shapes and MUG and Inception Score (IS) [52] on MUG. On Multimodal VoxCeleb and iPER datasets, we report Fréchet Video Distances (FVD) [58] that is computed from 2048 samples, and Precision-Recall Distribution (PRD) ($F_8$ and $F_{1/8}$) for diversity [51]. We further perform human evaluation using Amazon Mechanical Turk to verify the quality and diversity of synthesized videos.

4.1. Text-to-Video Generation

Shapes. We report the classification accuracy in Tab. 1 (top four rows) for the Shapes dataset. ART-V and MMVID are trained for 100K iterations. Compared with TFGAN [8], our model achieves significantly higher classification accuracy for Shape, Size, and Average (Avg) categories. Compared with ART-V, we perform better in all the categories. Note that our method has slightly lower accuracy on Color, Motion, and Direction (Dir) than TFGAN. The reason might be the VQGAN introduces errors in reconstruction, since the background is diverse and moving, and it needs to encode these small translations by only $4 \times 4$ codes. Note that to have a fair comparison, we do not apply text augmentation when performing comparison with other works.

MUG. We follow the experimental setup in TiVGAN [30] for MUG.
Table 2. Inception Score (IS) and classification accuracy (%) on MUG for video generation. We mark ‘*’ to IS values reported in TiVGAN. Our model achieves highest accuracy and IS.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Methods</th>
<th>Gender (%) ↑</th>
<th>Expression (%) ↑</th>
<th>IS ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Only</td>
<td>TGAN [40]</td>
<td>-</td>
<td>-</td>
<td>*4.63</td>
</tr>
<tr>
<td></td>
<td>MoCoGAN [57]</td>
<td>-</td>
<td>-</td>
<td>*4.92</td>
</tr>
<tr>
<td>ART-V</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*4.65</td>
</tr>
<tr>
<td>TiVGAN [30]</td>
<td>-</td>
<td>-</td>
<td>- *5.34</td>
<td></td>
</tr>
<tr>
<td>TFGAN [8]</td>
<td>99.22</td>
<td>100.00</td>
<td>5.53</td>
<td></td>
</tr>
<tr>
<td>ART-V</td>
<td>93.46</td>
<td>99.12</td>
<td>5.72</td>
<td></td>
</tr>
<tr>
<td>MMVID (Ours)</td>
<td>99.90</td>
<td>100.00</td>
<td>5.94</td>
<td></td>
</tr>
<tr>
<td>Multimodal</td>
<td>ART-V</td>
<td>89.16</td>
<td>98.54</td>
<td>5.59</td>
</tr>
<tr>
<td></td>
<td>MMVID (Ours)</td>
<td>98.14</td>
<td>100.00</td>
<td>5.85</td>
</tr>
</tbody>
</table>

Table 3. Evaluation metrics for text-to-video generation on iPER and Multimodal VoxCeleb datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Methods</th>
<th>Human (%) ↑</th>
<th>FVD ↓</th>
<th>$F_8$ ↑</th>
<th>$F_{1/8}$ ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPER</td>
<td>ART-V</td>
<td>-</td>
<td>277.604</td>
<td>0.936</td>
<td>0.806</td>
</tr>
<tr>
<td></td>
<td>MMVID (Ours)</td>
<td></td>
<td>289.127</td>
<td>0.944</td>
<td>0.924</td>
</tr>
<tr>
<td>Multimodal</td>
<td>ART-V</td>
<td>46.0</td>
<td>70.955</td>
<td>0.946</td>
<td>0.940</td>
</tr>
<tr>
<td>VoxCeleb</td>
<td>MMVID (Ours)</td>
<td>54.0</td>
<td>59.464</td>
<td>0.965</td>
<td>0.961</td>
</tr>
</tbody>
</table>

for experiments on the MUG expression dataset. We train models with a temporal step size of 8 due to the memory limit of GPU. Note TiVGAN is trained with a step size of 4 and generates 16-frame videos, while our model generates 8-frame videos in a single forward. We also train a 3D ConvNet as described in TiVGAN to evaluate the Inception Score and perform classification on Gender and Expression. Results are shown in Fig. 2a and Tab. 2 (top 8 rows). Our model achieves the best performance.

**iPER**. We show the results of the dataset in Tab. 3 (top 3 rows), demonstrating the advantages of our method. Long sequence generation results are shown in Fig. 5.

**Multimodal VoxCeleb**. We train ART-V and our model at a spatial resolution of $128 \times 128$ and a temporal step of 4 to generate 8 frames. Our method shows better results than ART-V on all the metrics, as shown in Tab. 3 (bottom two rows). We notice ART-V can also generate video samples with good visual quality and are aligned well with the text descriptions. However, ART-V often produces samples that are not temporally consistent. For example, as shown in Fig. 2b, the frame generated by ART-V at $t = 1$ is sharp and clear, but frames at $t = 5$ or $t = 8$ are blurry. The reason might be the exposure bias in autoregressive models becomes more obvious as the sequence length is long, i.e., an 8-frame video at resolution $128 \times 128$ has 512 tokens. Thanks to bidirectional information during training and inference, our MMVID is able to produce temporally consistent videos.

### 4.2. Multimodal Video Generation

Multimodal conditions can evolve in two cases: independent and dependent, and we show experiments on both.

**Independent Multimodal Controls.** This setting is similar to conventional conditional video generation, except the condition is changed to multimodal controls. We conduct experiments on Shapes and MUG datasets with the input condition as the combination of text and image. The bottom two rows in Tab. 1 and Tab. 2 demonstrate the advantages of our method over ART-V on all metrics. Additionally, we provide generated samples in Fig. 3a, where only a partial image is given as the visual condition. As can be seen, ART-V cannot satisfy the visual constraint well and the generated video is not consistent. The quality degradation for multimodal video synthesis of ART-V can also be verified in Tab. 1 as it shows lower classification accuracy than text-only generation, while our method is able to generate high quality videos for different condition signals.

We also conduct extensive experiments of video generation under various combinations of textual and image controls on Multimodal VoxCeleb, as shown in Fig. 4. We apply three different image controls, including segmentation mask (Fig. 4 row (a) - (b)), drawing (Fig. 4 row (c) - (d)), and partial image (Fig. 4 row (d) - (f)). In Fig. 4 row (b), our
This man has mustache, straight hair and beard. He has arched eyebrows. He wears eyeglasses and goatee.

A person with draw in image one and mask in image two is talking.

A person with mask in image two and appearance in image one is talking.

She has high cheekbones, pale skin and arched eyebrows. She has big lips, straight hair and heavy makeup. She has rosy cheeks and blond hair. She is young. She wears lipstick.

He has pointy nose, black hair and double chin. He has straight hair, arched eyebrows and big nose. This man is chubby.

A person with appearance in image one and motion in the following frames is talking.

Figure 4. Independent and Dependent multimodal video generation on Multimodal VoxCeleb with textual control (TC), image control (IC), and video control (VC). Row (a) - (b): TC + IC (segmentation mask); Row (c): TC (null) + IC (drawing); Row (d) - (e): dependent TC + IC; Row (f) - (h): TC + IC (partial image) and the TC of (g) is obtained from the TC of (f) by replacing “blond” with “black”; Row (i): dependent TC + VC and the VC includes content and motion information.

Our model can synthesize frames with eyeglasses even though eyeglasses are not shown in segmentation mask. In Fig. 4 row (f) - (g), we show that using the same image control while replacing the “blond” with “black” in the text description, we can generate frames with similar content except the hair color is changed. Such examples demonstrate that our method has a good understanding of multimodal controls.

Dependent Multimodal Controls. Furthermore, we introduce a novel task for multimodal video generation where textual and visual controls are dependent, such that the actual control signals are guided by the textual description. For example, Fig. 3b illustrates how the text informs from which image the model should query color, shape, and background information. More synthesized examples on Multimodal VoxCeleb are given in Fig. 4. For Fig. 4 row (d) - (e), our model learns to combine detailed facial features
Figure 5. Extrapolation and Interpolation. Row (a) - (b): long sequence generation via extrapolation. Row (c): interpolating a real sequence. Frames in dotted red boxes are fixed as preservation control. Textual controls for each row are: (a) “Person 024 dressed in 2 is performing random pose, normal speed.”; (b) “Person 024 dressed in 1 is performing A-pose, normal speed.”; and (c) “Person 028 dressed in 2 is performing A-pose, normal speed.”

from drawing or image and coarse features (i.e., pose) from mask. For Fig. 4 row (i), our method successfully retargets the subject with an appearance from the given image control (IC1) and generates frames with the motion specified by consecutive images that provide motion control (VC1).

Table 4. Analysis on video augmentation strategies.

<table>
<thead>
<tr>
<th>Video Augmentation</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swap Shuffle Color Align</td>
<td>Shape Color Size Motion Dir Avg</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>90.43</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>91.02</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>92.48</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>90.62</td>
</tr>
<tr>
<td>✓ ✓ ✓ ✓</td>
<td>93.36</td>
</tr>
</tbody>
</table>

4.3. Long Sequence Generation and Ablation

Long Sequence Generation. Our approach enables temporal extrapolation of videos. We show samples of video extrapolation and interpolation in Fig. 5. Samples from Fig. 5 row (a) - (b) are generated by being iteratively conditioned on previous 6 frames to generate the following 2 frames. Fig. 5 row (c) shows an example of synthesizing one frame by interpolating two consecutive real frames.

Analysis on VID Task. We perform analysis for different VID strategies on the Shapes dataset. Tab. 4 shows that the highest average accuracy is achieved when all augmentation is used (sampled uniformly). Also note that accuracy for color is the highest when we only apply color augmentation.

Analysis on Language Embedding. Analysis of using a pretrained language model is shown in Fig. 6. The method with a language model (w/ RoBERTa) is more robust to various text inputs than the one without it (w/o RoBERTa).

5. Limitation and Conclusion

This paper targets a new problem, which is video generation using multimodal inputs. To tackle the problem, we utilize a two-stage video generation framework that includes an autoencoder for quantized representation of images and videos and a non-autoregressive transformer for predicting video tokens from multimodal input signals. Several techniques are proposed, including the special VID token, textual embedding, and improved mask prediction, to help generate temporally consistent videos. On the other hand, the proposed method also contains some limitations, including temporal consistency issues for high-resolution videos, generating diverse motion patterns for longer sequences, and further improving the diversity of non-autoregressive transformers. More details can be found in the Appendix. Besides improving the limitation, a future direction might be to leverage more control modalities, such as audio, to generate videos with a much higher resolution.
References


[31] Yitong Li, Martin Min, Dinghan Shen, David Carlson, and Lawrence Carin. Video generation from text. In *AAAI*, 2018. 2


[57] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 1, 2, 6

[58] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. To


