TV2TV: A Unified Framework for Interleaved Language and Video Generation

Supplementary Material

7. Additional details: TV2TV

7.1. Architecture details

Modality-specific self-attention. Following the original MoT design [29], the text hidden states $\boldsymbol{h}_{\text{in}}^{\text{txt}}$ and video hidden states $\boldsymbol{h}_{\text{in}}^{\text{vid}}$ are transformed to their respective queries, keys, and values via separate Q, K, V matrices. The pre-attention layer normalization is also modality-specific and is folded into the QKV functions in the equations below for simplicity.

$$\boldsymbol{h}_{\text{Q}}^{\text{txt}} = \text{Q}_{\text{txt}}(\boldsymbol{h}_{\text{in}}^{\text{txt}}), \quad \boldsymbol{h}_{\text{K}}^{\text{txt}} = \text{K}_{\text{txt}}(\boldsymbol{h}_{\text{in}}^{\text{txt}}), \quad \boldsymbol{h}_{\text{V}}^{\text{txt}} = \text{V}_{\text{txt}}(\boldsymbol{h}_{\text{in}}^{\text{txt}}) \tag{3}$$

$$\boldsymbol{h}_{\text{Q}}^{\text{vid}} = \text{Q}_{\text{vid}}(\boldsymbol{h}_{\text{in}}^{\text{vid}}), \quad \boldsymbol{h}_{\text{K}}^{\text{vid}} = \text{K}_{\text{vid}}(\boldsymbol{h}_{\text{in}}^{\text{vid}}), \quad \boldsymbol{h}_{\text{V}}^{\text{vid}} = \text{V}_{\text{vid}}(\boldsymbol{h}_{\text{in}}^{\text{vid}}) \tag{4}$$

Attention is then computed across all tokens in the interleaving sequence. The attention-weighted values are projected back to the hidden state dimension using modality-specific O matrices.

$$\boldsymbol{h}_{\text{O}}^{\text{txt}} = \text{O}_{\text{txt}}\left(\text{softmax}\left(\frac{\text{mask}(\boldsymbol{h}_{\text{Q}}^{\text{txt}}\, \boldsymbol{h}_{\text{K}}^{\text{all}\,T})}{\sqrt{d}}\right) \boldsymbol{h}_{\text{V}}^{\text{all}}\right) \tag{5}$$

$$\boldsymbol{h}_{\text{O}}^{\text{vid}} = \text{O}_{\text{vid}}\left(\text{softmax}\left(\frac{\text{mask}(\boldsymbol{h}_{\text{Q}}^{\text{vid}}\, \boldsymbol{h}_{\text{K}}^{\text{all}\,T})}{\sqrt{d}}\right) \boldsymbol{h}_{\text{V}}^{\text{all}}\right) \tag{6}$$

where mask denotes a hybrid attention mask—applying a causal mask to the positions of text tokens and a block-causal mask to the positions of noisy and clean video tokens. An additional principle for masking is that noisy video tokens cannot be attended by any future tokens in the sequence. A global 1D RoPE is also applied here to all positions for all modalities.

Modality-specific feed-forward network. Again, following the original MoT [29] design, after self-attention, we use modality-specific FFNs to further transform text and video representations separately. The pre-FFN layer normalization is also modality-specific and is folded in the FFN function for simplicity.[3]

$$\boldsymbol{h}_{\text{FFN}}^{\text{txt}} = \text{FFN}_{\text{txt}}(\boldsymbol{h}_{\text{O}}^{\text{txt}}) \tag{7}$$

$$\boldsymbol{h}_{\text{FFN}}^{\text{vid}} = \text{FFN}_{\text{vid}}(\boldsymbol{h}_{\text{O}}^{\text{vid}}) \tag{8}$$

$$\tag{9}$$

[3] We also do not show residual connections for simplicity of notation.

7.2. Extending sequences beyond the trained context length

Because TV2TV is globally autoregressive, it can natively generate videos longer than its trained context length using sliding windows. To extend an interleaved text-video sequence $\oplus_{i=1}^{N}(\boldsymbol{x}_i^{\text{txt}}, \boldsymbol{x}_i^{\text{vid}})$, we retain the second half $\oplus_{i=N/2}^{N}(\boldsymbol{x}_i^{\text{txt}}, \boldsymbol{x}_i^{\text{vid}})$ and use it as a condition $\oplus_{i=1}^{N/2}(\boldsymbol{x}_i^{\text{txt}}, \boldsymbol{x}_i^{\text{vid}})$ for the next generation window.

8. Additional details: Controlled Experiments with Video Game Data

8.1. Model configuration details

See Table 2.

8.2. CS:GO actions as text

In Pearce and Zhu [36], each video frame is associated with a controller action, but as we group each 4 frames into a single video latent, we concatenate 4 controller actions, stringify them as text, and pass them to the model, e.g.:

(w, d, shift). 10, 0. 0, 0.		*(d). -100, 10. 0, 0.*
(d). 4, 0. 0, 0.	or	*(d). -100, 4. 0, 0.*
(d). 4, 0. 0, 0.		*(). -60, 0. 0, 0.*
(d). 0, 0. 1, 0.		*(). -4, 0. 0, 1.*

where the string includes: *(keyboard inputs). horizontal mouse move, vertical mouse move. left mouse click, right mouse click.* Example keyboard inputs are (w, a, s, d, space, ...): walk forward, left, backward, right, jump, etc. See Pearce and Zhu [36] for additional details on the CS:GO action space.

8.3. Human Evaluation Task

Annotators evaluate overall video quality within pairings according to the question:

> Which video has better visual quality? Examples of poor visual quality include cloudy/flickering generation quality, jumping through walls, static player movement (e.g. moving very slowly in one direction), and random jumps to elesewhere in the map. Do not penalize ghost-like or translucent characters.
>
> Answer choices:
>
> - Left has significantly better visual quality.
> - Left has marginally more visual quality.
> - Unsure or both seem equally good/bad.
> - Right has marginally more visual quality.
> - Right has significantly better visual quality.

Layers	28
Model Dimension	3072
FFN Dimension	8192
Attention Heads	24
Key/Value Heads	8
Activation Function	SwiGLU
Vocabulary Size	128K
Positional Embeddings – Interleaved Sequence	1D RoPE
Positional Embeddings – Video Only	2D APE
Training Steps	50K
Batch Size	128
Learning rate	3e-4
Max Context Length	15360
Tokens per Frame Chunk	240
Timestep t	$\text{logistic}(\mathcal{N}(0, 1.96))$
Text Dropout Rate $p_{\text{txt-drop}}$	0
Clean Video Flip Rate $p_{\text{clean-vid-flip}}$	0.5

Table 2. **Model configuration details** for TV2TV and baselines for experiments on video game data in §3. All model variants adopt a 3B-MoT Transfusion architecture.

Pointwise comparisons Annotators evaluate alignment between the generated video and intervention prompt according to the following instruction. In this instruction, 'Caption A' refers to the user-controllable intervention prompt included with the generated video, and 'START' and 'STOP' are assistive visual indicators added posthoc to the generated video corresponding to the intervention timestamp:

> Please watch the video. Does Caption A correctly reflect the clip shown between 'START' and 'STOP'? Consider primarily the period shown between START/STOP, although if the caption refers to jumping, reloading, or moving backwards you may also consider the period *immediately* following 'STOP'. Please use the rest of the video only for general context.
>
> Answer choices:
>
> - The caption correctly reflects the video.
> - The caption does not correctly reflect the video.
> - Unsure - the player is not actively playing (taken down by enemy and can't move).
> - Unsure - other (please specify in comments).

Additionally, visual quality is evaluated with the question:

> How is the overall visual quality of the video? Examples of poor visual quality include cloudy/flickering generation quality, jumping through walls, static player movement (e.g. moving very slowly in one direction), and random jumps to elesewhere in the map. Do not penalize ghost-like or translucent characters.
>
> Answer choices:
>
> - The video has strong visual quality.
> - The video has moderate visual quality.
> - The video has weak visual quality.
> - The video has no visual quality.

9. Additional details: Scaling TV2TV to Real World Data

9.1. Interleaved data augmentation pipeline details

An overview of the pipeline is shown in Figure 7.

Data sourcing We focus on building a dataset of sports content by filtering the YT-Temporal-1B dataset [60] using keyword-based filters (e.g. "*game highlights*"). We chose the sports domain for its high action density, which provides a strong testbed for interleaved reasoning capabilities. This yields 38K total hours of data.

Scene detection and segmentation We segment the data into scenes using TransNetV2 [41], a shot boundary detection model based on 3D convolutional networks. To identify clips containing interesting content, we employ a two-step approach. First, we use the Perception Encoder [8] to embed video frames and compute the cosine distance between consecutive frames, producing a time-series of semantic change. Peaks in this time-series may indicate moments where significant action likely occurs, so we refer to the frames associated with these peaks as *key frames*. We then apply an 8-second sliding window to each scene, extracting clips that are between 6 and 16 seconds long and contain the highest number of peaks. On average, these clips are 8.2 seconds in length.

Finally, we use a combination of key frames, hierarchical clustering of Perception Encoder embeddings [15], and heuristics to further segment the clips into chunks of frames suitable for interleaved captioning. Each clip is divided into an average of 4.3 chunks, though the number of chunks can range from as few as 2 to as many as 10 depending on the length of the video and the number of key frames detected. We impose a minimum chunk length of 1-second; the average chunk length is 1.9 seconds.

Filtering We apply several scene-level filters to further refine our selection:

- **VLM-based quality classifier:** To select high quality scenes we prompt Gemma-3-12B-Instruct [43] to select semantically-relevant content. For each scene, we sample consecutive frames from the start, center, and end of the video and ask the model to provide a score of 1-10 based on the perceived quality and relevance; see §9.2 for the full prompt.
- **Face bounding box filter:** We observed that a considerable portion of videos consisted of people talking directly to the camera without meaningful action or motion in the foreground. To remove such videos, we use RetinaFace [20] to obtain face bounding boxes and analyze both their coverage and temporal stability throughout the video. Clips with large, stable face bounding boxes are filtered out.
- **Motion filter:** We compute the optical flow for each clip [23] and calculate its average magnitude across frames as a motion score. Clips with low motion scores, indicating static or minimal movement, are filtered out.

After filtering, our final dataset comprises 8K hours of sports video data.

Interleaved captioning Finally, following recent work on differential video captioning [14–16, 37, 51, 55], we use Qwen2.5-VL-72B and Qwen3-VL-30B-A3B-Instruct [4] to generate (1) an overall *meta-caption* for the video and (2) differential captions describing action changes across subsequent frame chunks. Detailed prompts passed to the VLMs are provided in the Appendix §9.3. An example interleaved document is provided in Table 3.

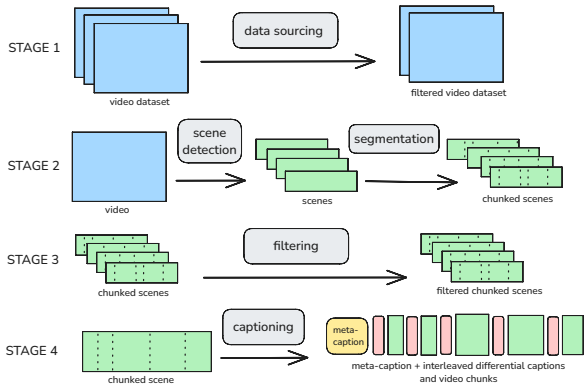


Figure 7. **Interleaved text and video data pipeline**. We present our methodology for using constructing an interleaved text and video training dataset from real world videos. The pipeline consists of several stages, including (1) data sourcing, (2) scene detection and segmentation, (3) filtering, and finally, (4) interleaved captioning with a VLM.

9.2. VLM-based quality filtering prompt

You are a capable model that can determine if the video is high quality or low quality, here are criteria to determine the quality:
1) Video is low quality if it has person talking to camera without any other motion, face in corners is OK.
2) Video is low quality if there is jittery motion due to camera movement.
3) Video is low quality if there is no motion with blank or static screen or image with just zoom in and zoom out.
4) Video is high quality if it has meaningful sports content like highlights of a game being played.

Now please rate video with a score between 1 and 10, where 1 is low quality and 10 is high quality, return the score in json format e.g.: {'quality_score': <predicted score>}.

Here are frames from new video sampled from start, middle and end of video:

9.3. VLM-based differential captioning prompt

You are a cautious video describer.
You will be shown multiple video segments from the same source video, shown in chronological order.
Describe what happens in EACH segment separately.
DO NOT reference 'the video' or 'the segment' in your descriptions.
DO NOT describe any text in the video.
Most important: Describe the actions or movements of the main characters or objects in each segment.
DO NOT anticipate future actions; only describe actions that are clearly visible in the current segment.
DO NOT repeat descriptions from previous segments.
If nothing meaningfully changes in the current segment compared to previous segments, use an empty string "" for the description of the current segment.
Keep each description concise (under 50 words). DO NOT hallucinate.

Format your output EXACTLY as follows:
Description of segment 1: [your description here]
Description of segment 2: [your description here]
...
Description of segment <N>: [your description here]
Here are the video segments in order: <VIDEO SEGMENTS>

9.4. Model configuration details

See Table 4.

9.5. Human Evaluation Task

For the human evaluation, all pairs are evaluated by a pool of professional external annotators via the Turing platform for increased robustness.

We include the evaluation questions answered by the annotators for the results discussed in §4:

- **Prompt alignment**: Which video is more aligned with the input language instruction?
  - Left is significantly more aligned with the text instruction.
  - Left is marginally more aligned with the text instruction.
  - Unsure or both seem equally good/bad.
  - Right is marginally more aligned with the text instruction.
  - Right is significantly more aligned with the text instruction.
- **Real world fidelity**: Which video has better fidelity to the real world?
  - Left has significantly better fidelity to the real world.
  - Left has marginally better fidelity to the real world.
  - Unsure or both seem equally good/bad.
  - Right has marginally better fidelity to the real world.
  - Right has significantly better fidelity to the real world.
- **Visual quality**: Which video has better visual quality?
  - Left has significantly better visual quality.
  - Left has marginally better visual quality.
  - Unsure or both seem equally good/bad.
  - Right has marginally better visual quality.
  - Right has significantly better visual quality.
- **Holisitic preference**: Which video do you prefer holistically?
  - I strongly prefer Left.
  - I somewhat prefer Left.
  - Unsure or both seem equally good/bad.
  - I somewhat prefer Right.
  - I strongly prefer Right.

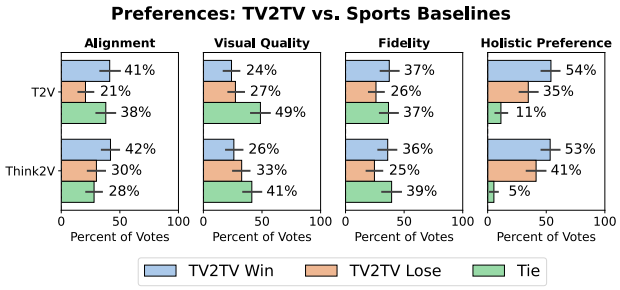


Figure 8. **Evaluation of TV2TV trained on real world interleaved sports data *vs.* T2V and Think2V in a controlled setup.** Compared to T2V, TV2TV shows strong win rates across alignment and holistic preference, with similar visual quality and fidelity. When compared to Think2V, TV2TV shows a similar pattern of improved alignment and overall preference, though it is not statistically significant.

9.6. Comparing TV2TV to sports baselines

In addition to comparisons with external models, we evaluate TV2TV with comparable T2V and Think2V baselines trained with the same real-world sports data. We find that when compared to T2V and Think2V baselines trained with the same real-world sports data as the TV2TV model, the TV2TV model shows strong win rates on alignment, visual quality, and holistic preference, while matching performance on real-world fidelity, as seen in Figure 8.

9.7. Steering the model with interleaved text

Table 5 shows an example of how interleaved captions can dynamically alter generation trajectories.

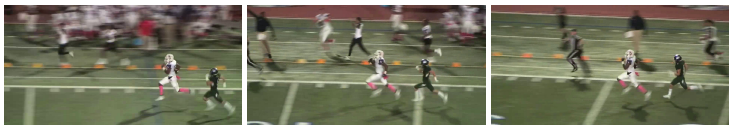
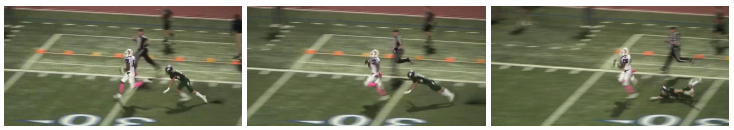
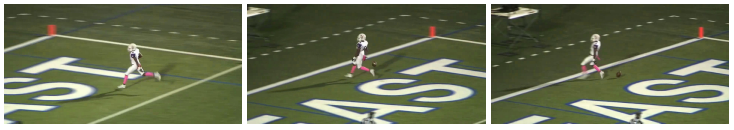
Interleaved caption	Subsampled frames
[0s - 2.2s] A player in a white uniform with pink socks runs with the ball, evading a defender in a black uniform. The player in white sprints towards the sideline, maintaining possession of the ball. The defender in black trails behind, attempting to catch up. Spectators are visible in the background.	
[2.2s - 4.3s] The player in the white uniform with pink socks continues running with the ball, moving further downfield. The defender in the black uniform remains in pursuit, closing the distance. Additional players in black uniforms join the chase, running towards the player with the ball. The player in white maintains possession and speed, evading the approaching defenders.	
[4.3s - 6.0s] The defender in the black uniform attempts a tackle but misses, falling to the ground. The player in the white uniform continues running unopposed towards the end zone.	
[6.0s - 8.6s] The player in the white uniform continues running towards the end zone, approaching the goal line.	
[8.6s - 10.1s] The player in the white uniform crosses the goal line, scoring a touchdown. The ball is now on the ground near the end zone.	
Overall meta-caption: A player in a white uniform with pink socks runs with the ball, evading defenders, and scores a touchdown.	

Table 3. **Example interleaved training document**. The source data is from YT-Temporal-1B [60]. Interleaved captions and the overall meta-caption are generated by Qwen3-VL-30B-A3B-Instruct [4].

Layers	32
Model Dimension	4096
FFN Dimension	14336
Attention Heads	32
Key/Value Heads	8
Activation Function	SwiGLU
Vocabulary Size	128K
Positional Embeddings – Interleaved Sequence	1D RoPE
Positional Embeddings – Video Only	2D APE
Training Steps	250K
Batch Size	512
Learning Rate	3e-4
Max Context Length	13056
Tokens per Frame Chunk	240
Timestep t	$\mathrm{logistic}(\mathcal{N}(0, 1.96))$
Text Dropout Rate $p_{\text{txt-drop}}$	0.05
Clean Video Flip Rate $p_{\text{clean-vid-flip}}$	0.2

Table 4. **Model configuration details** for TV2TV and baselines for experiments on real world sports data in §4. All model variants adopt an 8B-MoT Transfusion architecture.

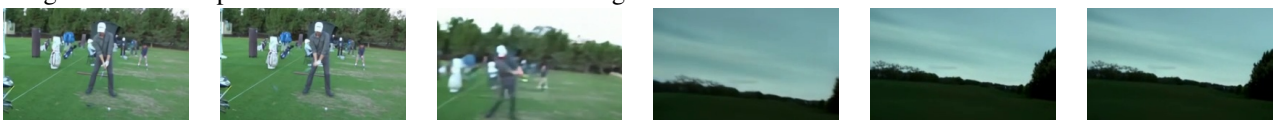
Subsampled frames					
t=0.0	t=1.19	t=2.38	t=3.62	t=4.81	t=6.06
Intervention 1 at t=1.56: "The man completes his golf swing, raising the club above his shoulder as he follows through. The golf ball is no longer visible, having been struck and sent flying forward. His body turns slightly to the left, maintaining balance after the powerful swing."					
					
Intervention 2 at t=1.56: "The man completes his golf swing, raising the club above his shoulder as he follows through. The camera pans to track the ball as it soars through the air."					
					
t=0.0	t=0.75	t=1.5	t=2.31	t=3.06	t=3.88
Intervention 1 at t=1.56: "The player in the white jersey takes control of the ball and runs towards the goal. He kicks the ball powerfully towards the net. The goalkeeper in the yellow jersey leaps to the left, extending his arms in an attempt to block the shot. The ball moves swiftly towards the goalpost as the goalkeeper's jump reaches its peak."					
					
Intervention 2 at t=1.56: "The player in the red jersey intercepts the ball near the center of the field and starts dribbling towards the right side of the frame, moving away from the goal. He evades an approaching defender in a white jersey by maneuvering the ball skillfully. As he advances, other players adjust their positions, preparing for the next phase of play. The goalkeeper remains near the goal, observing the developing action."					
					

Table 5. **Comparison of video rollouts steered by different interleaved captions.** We alter the interleaved caption at second 1.56s and observe how the generation trajectory is altered.