

TGT: Text-Grounded Trajectories for Locally Controlled Video Generation

Supplementary Material

A. Label Local Text and Motion

A.1. Point Description VLM

We adopt a teacher-student strategy to train a lightweight vision-language model capable of generating textual descriptions for arbitrary points within an image.

Training set construction. Following a procedure similar to [9], we build point-conditioned captioning supervision on the train 2017 split of COCO [3] dataset by first extracting candidate points of interest (POIs) from each image; the supervision signal is provided as localized captions paired with specific image coordinates. Concretely, we run Ultralytics SAM2.1-large [6] to obtain instance masks and compute the geometric centroid of every mask in image coordinates. To avoid redundant, tightly clustered centroids, we perform non-maximum suppression (NMS) in point space by treating each centroid as a disk of fixed radius $r = 32$ pixels and randomly retaining a single representative point when overlaps occur. The resulting set of de-duplicated centroids constitutes the POIs for that image.

For annotation, we assign each POI a unique integer identifier and render the identifiers onto the image for reference (see Figure 1). The annotated image is then provided to a teacher VLM (GPT-4o in our implementation), which is prompted to produce a concise localized description for every identifier, yielding one text string per point. Each supervision unit is recorded as an $\langle \text{image}, (x, y), \text{text} \rangle$ triplet, where (x, y) denotes the POI’s absolute pixel coordinates with the image’s native resolution. Collecting such triplets over the full split produces a large corpus of image–point–text pairs that we subsequently use to train the student model to generate local text given an image and a coordinate query.

Training details. We train a Qwen2.5-VL-3B model as the student model to map an input image and a pixel-coordinate query to a localized caption using the image-point-text triplets described above (see Figure 2). The training prompt is a single-line instruction: “ $\langle \text{image} \rangle$ Generate a detailed caption for the object at (x, y) ”. No visual markers are rendered on the image. We freeze the vision encoder and update the multimodal projection module and the language model decoder. Optimization uses AdamW [4] with learning rate 2×10^{-6} , cosine decay, warmup ratio 0.03, weight decay 0, gradient clipping at 1.0, and BF16 precision. We train for 5 epochs with per-device batch size = 4 and gradient accumulation = 4 on 16 80GB VRAM GPUs; the global batch size per update is $4 \times 4 \times 16 = 256$.

A.2. Label Text and Motion on Training Video

In order to construct spatiotemporally grounded supervision for video, we extend the static point–text annotations introduced in Appendix A.1 into dynamic trajectories that encode both motion and semantics. The pipeline consists of three major components: (i) representative point selection, (ii) localized text generation, and (iii) temporal propagation via TAP.

Representative point selection and localized text assignment. For each video frame, entity masks are first obtained using Grounded SAM [2, 6]. To summarize each entity with a small but informative set of anchor points, we apply an adaptive sampling strategy. If the number of foreground pixels in the mask is below a threshold (set to $0.01 \times H \times W$, where H and W are the frame height and width), the entity is represented by a single point: the center of its bounding box. For larger entities, the mask’s bounding box is partitioned into a grid of roughly square subregions, such that each subregion covers at most the threshold number of pixels. Within each subregion containing foreground pixels, we compute the bounding box of the local foreground and take its center as the representative point. This ensures that large entities with complex shapes are covered by multiple anchors, while small entities are efficiently represented by a single point. Each representative point is then paired with a coordinate-based query and passed to our point description VLM. This step yields concise, location-specific captions to these sets of representative points.

Trajectory propagation with TAP. Once static point to text pairs are obtained on the initial frame, we convert them into full trajectories using Tracking-Any-Point (TAP) [1]. TAP is a transformer-based video point tracker capable of following arbitrary query points over long temporal horizons. Given a sampled point (x_0, y_0) on frame 0, TAP predicts its corresponding coordinates (x_t, y_t) across subsequent frames t given a video. Importantly, TAP outputs both tracked positions and visibility flags, where the latter indicate occlusion or out-of-frame states. This allows each trajectory to encode not only motion but also reliability. Every propagated point inherits its associated localized description, producing a trajectory–text pair that maintains semantic grounding over time.

Final dataset. The result of this procedure is a collection of temporally consistent trajectories, each annotated with descriptive text. Formally, each unit is represented as a sequence

$$\langle (x_t, y_t, v_t)_{t=0}^T, m \rangle, \quad (1)$$

where (x_t, y_t) are pixel coordinates at frame t , $v_t \in \{0, 1\}$

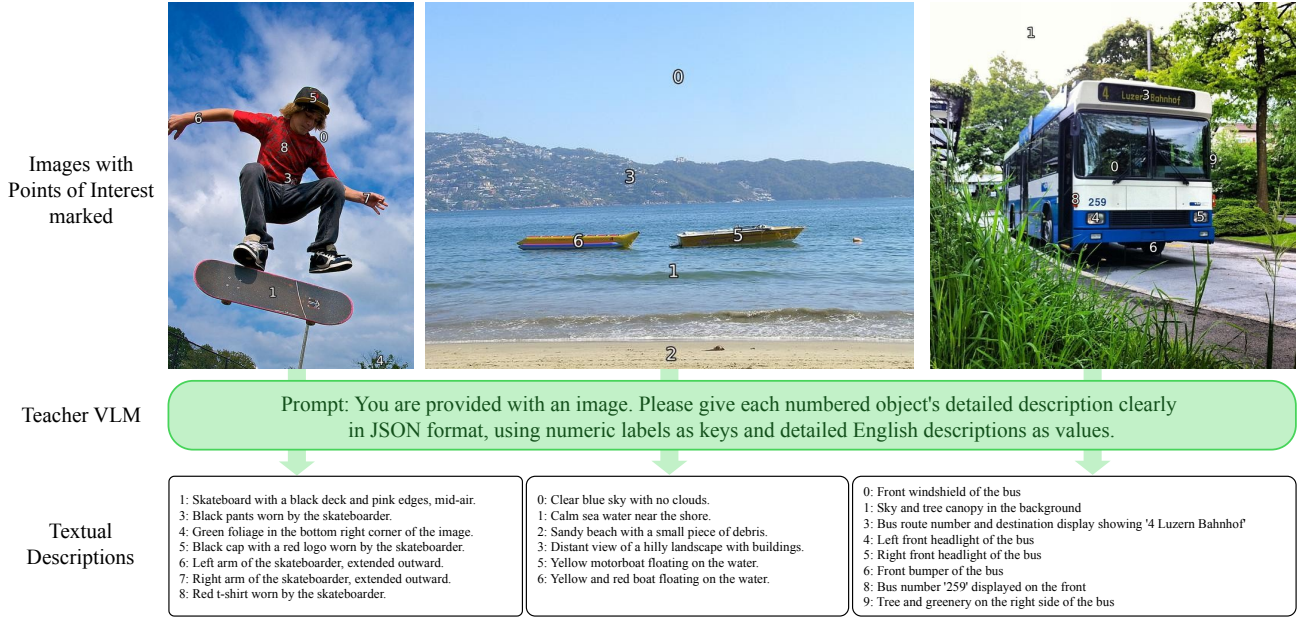


Figure 1. Data construction for distilling the local point caption VLM from an standard large VLM model. As the image shown, we prompt the teach model to generate labels via superimpose numbers onto images. Thereby the teacher model generate textual descriptions based on the numbered images.

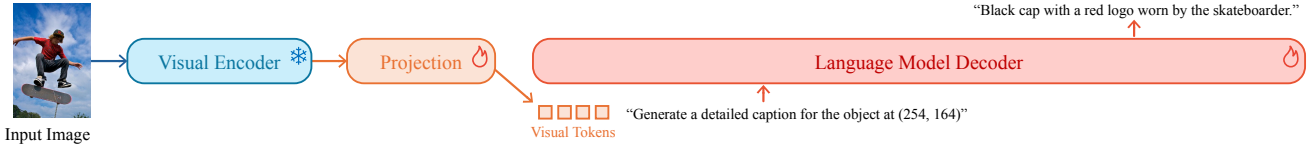


Figure 2. Illustration of finetuning the distilled VLM using the constructed training set. Specifically, we only train the token projector and the Language model decoder during finetuning.

denotes visibility, and m is the localized caption. Together, these labeled trajectories provide dense, multimodal supervision for training, capturing both the where and what information of entities throughout the video.

B. Baseline Implementation Details

B.1. WanT2V 2.2

We compare TGT against the Wan2.2 14B text-to-video model [7], under two baseline setups:

- Wan2.2 14B text-to-video with **global** prompt only.
- Wan2.2 14B text-to-video with **global + local** prompt.

In both cases, we use the official Wan2.2 T2V 14B model and its released implementation. For the global-only setup, the model is conditioned on the global caption extracted from the original reference video using the Qwen2.5-VL model, which is also the video prompt input to all evaluated methods unless specified otherwise. For the global + local setup, the model is conditioned on both the global caption

and the local prompts, where the latter are obtained using the same distilled VLM employed in our data pipeline. Figure 3 presents an example of generated outputs alongside their corresponding prompts for the Wan2.2 14B T2V baseline. Wan2.2 with extended prompt that describes motion can reconstruct high-level movement in original video better.

B.2. Tora & MotionCtrl

We evaluate TGT against Tora [10] and MotionCtrl [8] under a unified protocol. We first use the ground-truth segmentation mask on the initial frame to identify all entities and take the center representative point of each instance. Given these points, we run TAP to extract per-entity 2D trajectories across the sequence. Both Tora and MotionCtrl assume trajectories remain visible throughout; however, TAP marks some timesteps as "invisible" when tracking confidence falls below a threshold. For generation, we ignore TAP's visibility flags and feed the full continuous trajectory

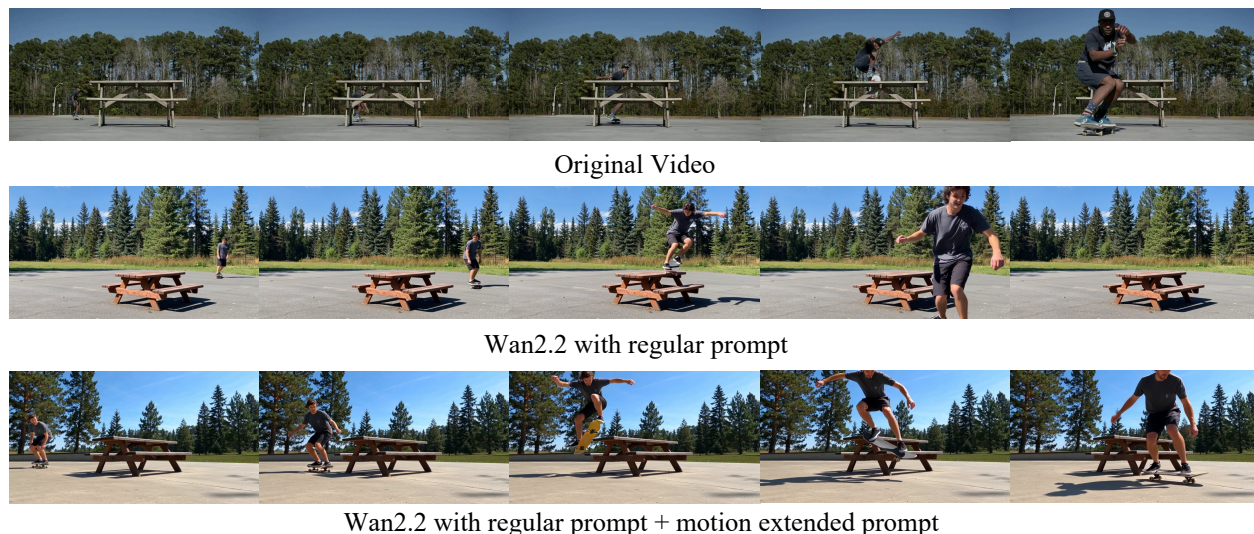


Figure 3. Examples of Wan2.2 T2V baseline models tested with regular prompt and with regular prompt plus motion-related extended prompt. The extended version can follow the direction the skateboarder comes better.

ries to these two methods to satisfy their input assumptions. For evaluation, we report trajectory error metrics only on visible timesteps after temporal alignment, ensuring a fair comparison

B.3. TrailBlazier

TrailBlazier [5] takes multi-frame bounding boxes with associated local text, builds a representation for each box (subject), and then fuses them into a single video. To form its inputs, we run grounded SAM on the video, guided by the dataset’s ground-truth segmentation masks, to obtain per-entity bounding boxes. We derive a short textual description for each box by querying our distilled VLM at the box center on the initial frame to produce a local prompt. Because TrailBlazier requires box locations over multiple frames, we follow its setup and uniformly sample boxes and descriptions at 1/4 of the sequence length to provide a sparse trajectory of boxes. TrailBlazier then generates a representation for each subject and fuses them to produce the final video.

C. Additional Qualitative Results

All videos are available in the uploaded supplementary materials. Figure 4, Figure 5, and Figure 6 illustrate qualitative results from TGT when conditioned on trajectories with different local prompt inputs. In all three examples, only basic scene-level information is given in the global prompt, without any explicit description of motion or interactions. The additional text-grounded trajectories are therefore responsible for shaping the observed behaviors. In Figure 4, an object is guided to float upwards, while Figure 5 shows a subject consistently descending on staircases. Figure 6

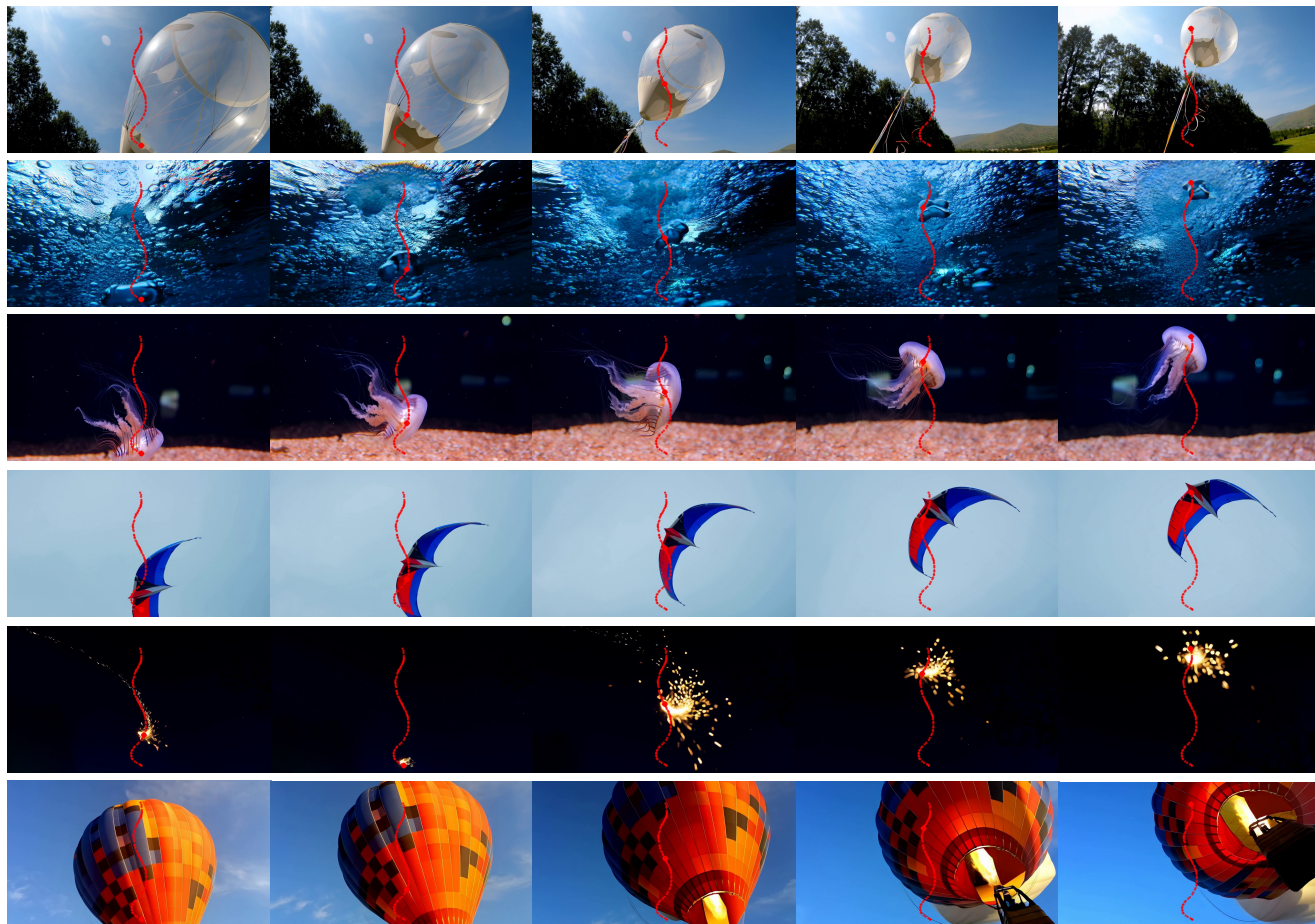
demonstrates more complex compositions involving multiple entities, each following its own trajectory. Across these cases, TGT produces correct items and coherent motions that align with the specified texts and trajectories, confirming the effectiveness of our method.

D. LLM Usage

We used large language models (LLMs) in two limited ways: (i) to help generate and refine example content such as candidate captions/local prompts for qualitative demonstrations, and (ii) to assist with wording, formatting, and editing during manuscript preparation. All model-suggested text and prompts were reviewed, edited, or discarded by the authors; no experimental design, implementation, or quantitative analysis depended on LLM output.

References

- [1] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *ICCV*, 2023. 1
- [2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 1
- [3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1
- [4] Ilya Loshchilov and Frank Hutter. Decoupled weight de-



a helium balloon / a stream of bubbles / a jellyfish / a kite / a glowing firefly / a hot-air balloon

Figure 4. Video generation results of TGT with a floating upwards trajectory under different local prompts.

- cay regularization. In *International Conference on Learning Representations*, 2019. 1
- [5] Wan-Duo Kurt Ma, J. P. Lewis, and W. Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia*, 2024. 3
- [6] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 1
- [7] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2
- [8] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2
- [9] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv:2310.11441*, 2023. 1
- [10] Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *CVPR*, 2025. 2



A child with a balloon / A cat / A monkey / A duck / A goat / A kangaroo / A squirrel / A suitcase / A ball of yarn / A ghost

Figure 5. Video generation results of TGT with a walking-down-stairs trajectory under different local prompts.



Red: A black van. Green: A helicopter. Blue: A house



Red: A wheelchair user. Green: A white balloon. Blue: A statue



Red: A child on a scooter. Green: A paper airplane. Blue: A shop window



Red: A street musician rolling a drum. Green: Soap bubbles. Blue: A picnic table



Red: A postal worker with a mail cart. Green: A shooting star. Blue: A small tent



Red: An astronaut hopping. Green: A small satellite. Blue: Sun in background



Red: A remote-control car. Green: A seagull. Blue: A rock in sea

Figure 6. Video generation results of TGT with multiple trajectories under different local prompts.