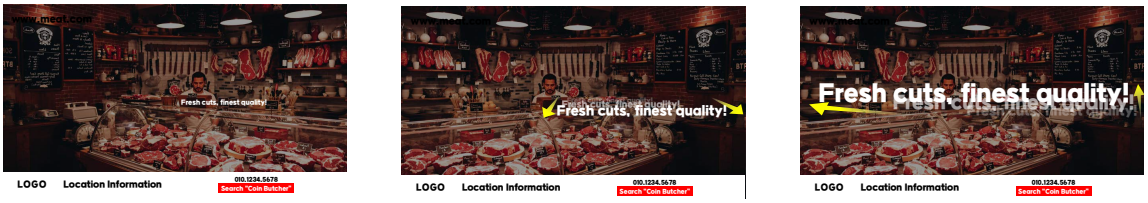
Generating Animated Layouts as Structured Text Representations

Yeonsang Shin[1*] Jihwan Kim[1*] Yumin Song[1]
Kyungseung Lee[2] Hyunhee Chung[2] Taeyoung Na[2]
[1]Seoul National University [2]SK telecom

(a) **Input:** The advertisement features text overlays on a house exterior … The first text says 'Sleek design,' followed by 'Your need first.' The final call to action reads 'Consult now!' … The top-right banner features 'GS Design.'



(b) **Input:** The video features text displaying "Fresh cuts, finest quality!" … The bottom banner shows a logo and text: "Location Information", "Search "Coin Butcher" and "010.1234.5678."

Figure 1. **Results generated by VAKER.** Given text prompts that specify the content and style, VAKER creates animated layouts for video advertisements. Each row illustrates the sequential animation of elements layered over the base video content.

Abstract

Despite the remarkable progress in text-to-video models, achieving precise control over text elements and animated graphics remains a significant challenge, especially in applications such as video advertisements. To address this limitation, we introduce Animated Layout Generation, a novel approach to extend static graphic layouts with temporal dynamics. We propose a Structured Text Representation for fine-grained video control through hierarchical visual elements. To demonstrate the effectiveness of our approach, we present VAKER (Video Ad maKER), a text-to-video advertisement generation pipeline that combines a three-stage generation process with Unstructured Text Reasoning for seamless integration with LLMs. VAKER fully automates video advertisement generation by incorporating dynamic layout trajectories for objects and graphics across specific video frames. Through extensive evaluations, we demonstrate that VAKER significantly outperforms existing methods in generating video advertisements.

[*]Equal contribution.
Project page: https://yeonsangshin.github.io/projects/Vaker

1. Introduction

Recent advances in text-to-video generation have demonstrated impressive capabilities in creating high-quality videos from text descriptions. While current diffusion-based approaches [4, 29, 34] excel at generating diverse videos with realistic scenes and movements, they struggle with readable text and precise control over animated elements. This limitation is particularly evident in videos that combine dynamic graphical elements with base video content, such as social media videos with animated captions, promotional content with moving visual elements, or informational videos with synchronized text overlays. The core challenge stems from two fundamental constraints: the difficulty in generating high-quality readable text within videos, and the lack of precise control over animated graphics that must be precisely positioned and timed on video content.

To address these challenges, we present *"Animated Layout Generation"*, a generation task that extends traditional static graphic layouts [6, 16] to incorporate temporal dynamics. Whereas conventional layouts organize visual objects through static bounding boxes, animated layouts add

temporal control to these objects—managing how they appear, move, and transform over time. This approach is particularly effective for creating videos with graphical elements that demand precise control over both spatial composition and temporal evolution of multiple visual elements.

The implementation focuses on *Structured Text (ST) Representation*, a novel representation that enables fine-grained control over video generation by organizing visual elements in a hierarchical structure. By transforming videos into text sequences, this representation format preserves essential spatial and temporal relationships while enabling consistent mapping from text to visual output. A key advantage of this text-based representation is its natural compatibility with Large Language Models (LLMs), which have demonstrated remarkable capabilities across various domains. Furthermore, the format excels at capturing on-screen text and spatial positioning, making it particularly effective for applications requiring detailed control over animated elements throughout video sequences.

In this paper, we present our approach through VAKER, a text-to-video advertisement generation pipeline that realizes Animated Layout Generation. VAKER decomposes the generation process into three stages (Banner, Mainground, and Animation), each leveraging ST-Representation to control spatial layout, visual attributes, and temporal dynamics. Additionally, we incorporate Unstructured Text (UT) Reasoning to effectively leverage LLM capabilities in translating user prompts into a structured format. Extensive evaluations through qualitative assessments, quantitative metrics, and user studies demonstrate that VAKER significantly outperforms existing approaches in generating video advertisements. Our work shows that the proposed approach successfully addresses the core challenges of precise text rendering and graphical control in video generation, contributing a robust framework for creating videos with dynamic visual elements.

The main contributions of this paper are summarized below:

- We introduce *Animated Layout Generation*, a novel approach to extend traditional layout generation to the temporal domain. By incorporating time-based controls into layout composition, our approach enables precise manipulation of dynamic graphics and ensures text readability in videos.
- We develop *ST-Representation*, a novel structured video representation that effectively captures temporal and spatial information in controllable text sequences. This design enables natural integration with LLMs while preserving detailed control over essential visual components.
- We present VAKER, a comprehensive text-to-video advertisement generation pipeline that demonstrates our approach in practice. Through rigorous evaluations, we show that VAKER generates superior results compared to existing approaches.

2. Related work

2.1. Text-to-video diffusion models

Recent advancements in text-to-video diffusion models have demonstrated substantial capabilities in generating visually diverse and realistic videos from textual prompts, underscoring their potential in general video synthesis tasks. Despite these strengths, current diffusion models face considerable challenges in handling specialized video content that requires embedded textual elements or fine-grained control of animated overlays. This limitation stems partly from design choices in state-of-the-art video generation models [4, 29, 34] that often exclude videos containing substantial text to mitigate visual artifacts in the dataset, consequently restricting the models' ability to generate integrated and readable text within videos. Furthermore, most approaches rely on text encoders that transform text into embeddings used to condition the diffusion process, an architecture inherently unsuitable for precise text rendering and spatial control. In response to these challenges, we introduce a Structured Text (ST) Representation for animated layouts to enable the generation of video content with embedded text, ensuring clarity, spatial precision, and temporal consistency.

2.2. Graphic layout generation

Graphic layout generation is central to automated design, supporting efficient, visually compelling content across various media types. Existing approaches in layout generation can be divided into two main categories based on their input format: constraint-based methods [6, 7] and Text-to-layout methods [19, 20].

Constraint-based. Constraint-based layout generation relies on predefined rules including element categories, sizes, relationships, as well as refinement and completion requirements. The field has evolved from early approaches using GANs [36, 37] and VAEs [1, 32] to recent advances in Transformer-based models [9, 15] and diffusion approaches [6, 7, 13, 16]. Recent works have also explored LLM applications [19, 20, 26]. While these methods excel at creating structured layouts, their rigid constraint-based input format struggles to capture complex requirements for specialized content such as readable text overlays or synchronized visual elements, motivating the need for more flexible natural language specifications.

Text-to-layout. Text-to-layout models [19, 20] generate layouts directly from natural language descriptions. Recent approaches leverage LLMs: LayoutPrompter [20] employs in-context learning without fine-tuning, while Parse-then-Place [19] utilizes a fine-tuned model to convert text into intermediate representations. These methods enhance accessibility by enabling natural language specifications rather than technical constraints. While these approaches handle

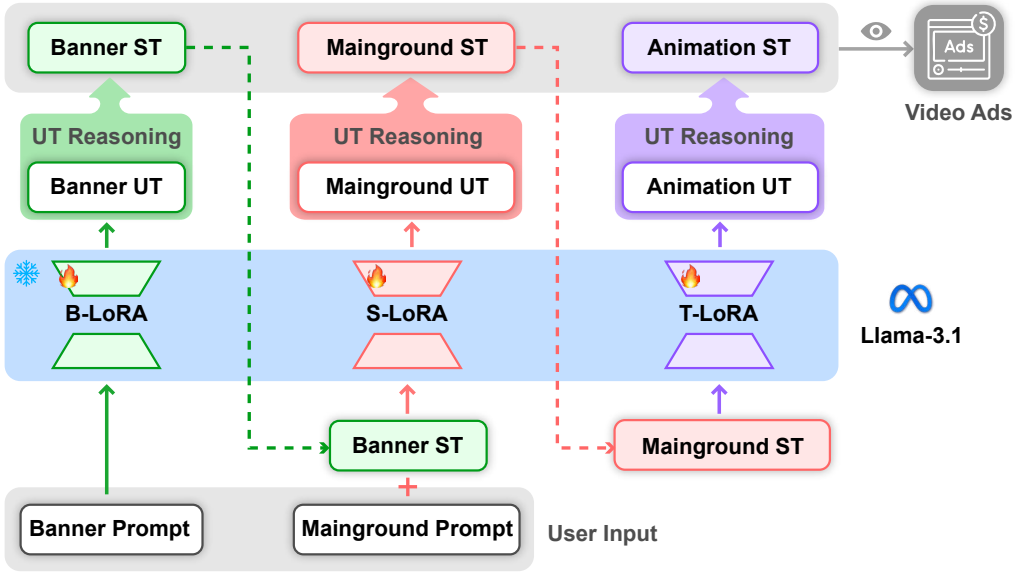


Figure 2. **Overview of VAKER**. VAKER takes text input from users to generate video advertisements. It does so by producing Structured Text (ST) Representations, a format we propose for encoding video layouts as structured text. The pipeline breaks down video advertisement creation into three components: Banner, Mainground, and Animation. Each component uses a LoRA-adapted language model that first interprets the design in natural language before generating structured specifications for the final video advertisement.

static layouts effectively, they lack the capability to manage dynamic layouts where elements change over time—a limitation that our Animated Layout Generation specifically addresses.

2.3. Large Language Models

Large Language Models (LLMs) [23, 27] have transformed natural language processing by understanding and generating human-like text. Beyond basic text generation, LLMs demonstrate sophisticated reasoning capabilities through various prompting techniques, including Chain-of-Thought prompting [30] for step-by-step problem solving. Recent studies have revealed that LLMs possess inherent visual understanding capabilities. Despite being trained solely on text data, they can effectively reason about spatial relationships and visual concepts [24]. These capabilities, combined with their proven success across diverse applications, have motivated our exploration of LLMs' potential for Animated Layout Generation.

2.4. Video advertisement analysis

Video advertisement analysis has primarily focused on understanding viewer behavior and commercial effectiveness. Prior works have examined advertising rhetoric [14], narrative structures [35], and multimodal elements [5]. However, the automated generation of video advertisements, particularly considering temporal dynamics and element positioning, remains largely unexplored.

3. ST-Representation for videos

The Structured Text (ST) Representation forms the foundation of our approach to video generation, transforming animated layouts into a structured textual format with multiple advantages. First, encoding video layouts as text sequences enables leveraging Large Language Models (LLMs), which excel at processing and generating text-based data. Second, this text-based representation directly preserves on-screen text, making it particularly effective for applications requiring precise text rendering. Third, the hierarchical key-value structure of ST-Representation effectively organizes complex visual information, maintaining essential spatial and temporal relationships within the content. Finally, this structured format facilitates a reliable mapping from text to visual output, ensuring consistent visualization of text-based representations in the generated videos.

Structured video representation. To efficiently manage diverse video components, we define a video clip $\mathbf{V}$ as a 4-tuple

$$\mathbf{V} = (\mathbf{B}, \mathbf{FG}, \mathbf{BG}, \mathbf{A}), \tag{1}$$

where $\mathbf{B}$, $\mathbf{FG}$, $\mathbf{BG}$, and $\mathbf{A}$ represent the Banner, Foreground, Background, and Animation information, respectively. We define each component as a structured form, which can be represented as structured text, such as JSON, XML, or YAML object.

The Banner

$$\mathbf{B} = \{B_j\}_{j \in \mathcal{J}} \tag{2}$$

denotes specific fixed regions and the objects inside them, with each B_j defined as

$$B_j = \left(b_j^{\text{ban}}, y_j^{\text{ban}}, \{o_m^{\text{ban}}\}_{m=1}^{M}\right). \tag{3}$$

Here, b_j^{ban} specifies the bounding box coordinates, y_j^{ban} encodes banner attributes, and $\{o_m^{\text{ban}}\}_{m=1}^{M}$ represents the set of M objects within the banner. We define attributes y as supplementary properties, such as content and color, which are critical for accurate rendering within the layout.

The Foreground

$$\mathbf{FG} = \{o_n^{\text{fg}}\}_{n=1}^{N} \tag{4}$$

comprises movable objects, such as images and text, that appear prominently within the scene and interact dynamically with other elements.

The Background

$$\mathbf{BG} = (c^{\text{bg}}, y^{\text{bg}}) \tag{5}$$

represents type of the background c^{bg} and its associated attributes y^{bg}.

The Animation

$$\mathbf{A} = \left(l, \{a_n^{\text{fg}}\}_{n=1}^{N}\right) \tag{6}$$

describes scene dynamics, where l represents total scene duration in frame count, and each animation a_n consists of a sequence of K keyframes for the n^{th} object. Each animation a_n is defined as

$$a_n = \{(f_k, b_k, t_k)\}_{k=1}^{K}, \tag{7}$$

where f_k is the keyframe index within the sequence, b_k specifies the bounding box coordinates of the object at that frame, and t_k represents additional transformation parameters (*e.g.*, rotation, transparency, scaling) that can be straightforwardly integrated to enable richer motion effects beyond positional changes.

Object information. Each object o_i within a scene is represented as a 3-tuple

$$o_i = (c_i, b_i, y_i), \tag{8}$$

where c_i denotes the object class (e.g., text, logo, image, etc.), b_i specifies the bounding box coordinates as $b_i = [x_1, y_1, w, h]$, with (x_1, y_1) representing the top-left corner coordinates and (w, h) denoting the object's width and height, and y_i encapsulates the object's specific attributes.

4. VAKER: Video Ad maKER

This section describes how we leverage ST-Representation and LLMs to generate realistic video advertisements.

Algorithm 1 Three-stage generation procedure of VAKER

Require: B_θ, S_ϕ, T_ψ
Input: $p^{\text{b}}, p^{\text{m}}$
Output: $(\mathbf{B}, \mathbf{FG}, \mathbf{BG}, \mathbf{A})$

$\mathbf{B} \leftarrow B_\theta(p^{\text{b}})$ ▷ B-LoRA
$\mathbf{FG}, \mathbf{BG} \leftarrow S_\phi(p^{\text{m}}, \mathbf{B})$ ▷ S-LoRA
$\mathbf{A} \leftarrow T_\psi(\mathbf{FG})$ ▷ T-LoRA

return $(\mathbf{B}, \mathbf{FG}, \mathbf{BG}, \mathbf{A})$

4.1. ST-Representation for video advertisements

We adapt the Structured Text (ST) Representation introduced in Sec. 3 to address the specific requirements of video advertisements.

In video advertisements, banners are typically positioned at the bottom, top-left, or top-right of the frame, indicating:

$$\mathcal{J} = \{\text{bottom}, \text{top-left}, \text{top-right}\}, \tag{9}$$

for $\mathcal{J}$ in Eq. (2). The attribute y_j^{ban} in Eq. (3) represents the banner color in this context.

The Background can be either a solid color or an image with varying attributes such that

$$(c^{\text{bg}}, y^{\text{bg}}) = (\text{solid color}, \text{color}) \tag{10}$$

or

$$(c^{\text{bg}}, y^{\text{bg}}) = (\text{image}, \text{text caption}). \tag{11}$$

We focus on two primary object classes: text and logo. Each object o_i in Eq. (8) is defined as either:

$$c_i = \text{text}, y_i = \{\text{raw text}, \text{text color}, \text{textbox color}\} \tag{12}$$

or

$$c_i = \text{logo}, y_i = \text{None} \tag{13}$$

For animation, we simplify to $a_n = \{(f_k, b_k)\}_{k=1}^{K}$, omitting transformation parameters t_k to focus on positional changes in this first exploration of animated layouts.

Importantly, our ST-Representation incorporates color schemas, enabling VAKER to generate both layouts and visual attributes cohesively. Detailed information about each component of the ST-Representation can be found in the supplementary materials.

4.2. Three-stage generation

Training an LLM to generate dynamic content with precise spatial positioning and temporal movement would require extensive training data and computational resources. To enable effective layout generation even with limited data, we decompose the process into three distinct stages, each handled by a specialized expert model fine-tuned with LoRA [12].

Our pipeline sequentially generates three Structured Text (ST) objects—Banner ST, Mainground ST, and Animation

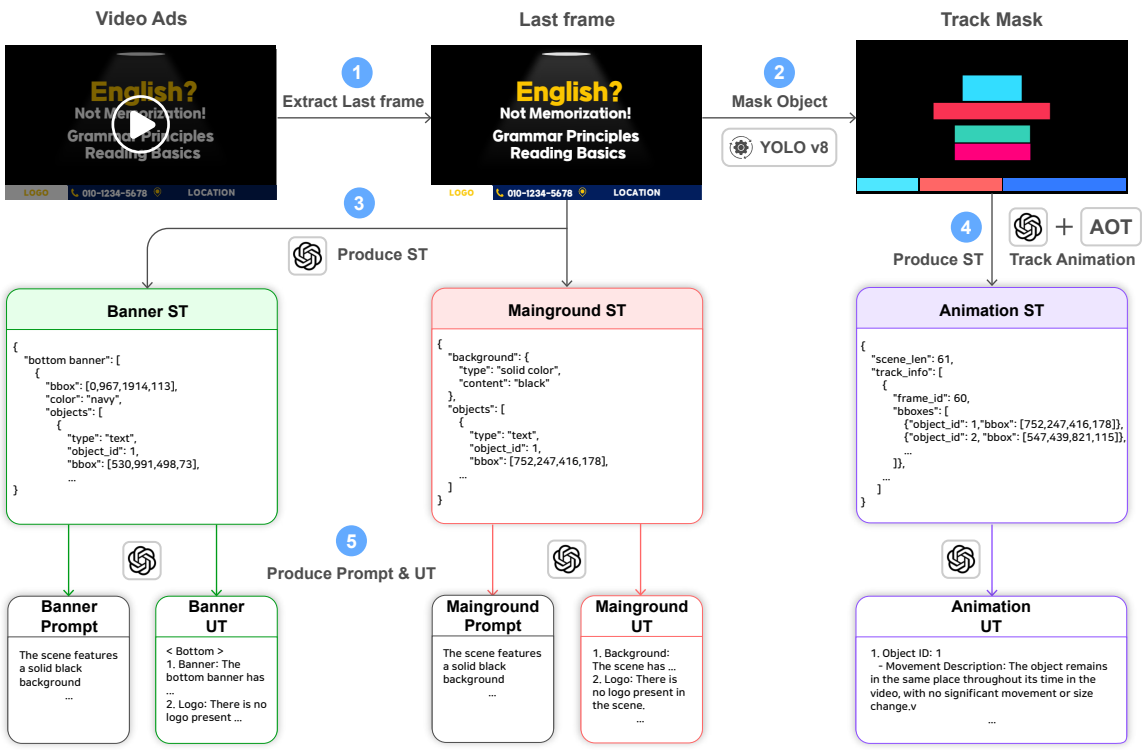


Figure 3. **Video Advertisement Dataset Construction Pipeline.** Our automated pipeline converts video advertisements into three types of data for training VAKER: Structured Text (ST), Unstructured Text (UT) Reasoning, and natural language prompts. Given a video advertisement, the pipeline performs the following steps: (1) extracts the last frame, (2) detects and classifies objects using fine-tuned detection models [28], and (3) generates Banner ST and Mainground ST from the spatial layout, (4) tracks object movements using tracking models [33] to generate Animation ST, and finally, (5) uses template-based prompting with LLMs to convert these ST-Representations into UT Reasonings and natural language prompts.

ST—based on user input prompts for the banner (p^{b}) and mainground (p^{m}). Each ST object is encoded in JSON format. We integrate both Foreground and Background into a single Mainground ST to effectively contextualize background information within the overall layout.

The generation process follows a sequential architecture. Initially, the Banner $\mathbf{B}$ is generated by the banner expert B_θ (B-LoRA), using the banner prompt p^{b}. Subsequently, the spatial expert S_ϕ (S-LoRA) generates Mainground ST based on the mainground prompt p^{m} and the previously generated Banner $\mathbf{B}$. Lastly, the temporal expert T_ψ (T-LoRA) produces the animation $\mathbf{A}$, defining the movements of the foreground objects. The complete generation process is presented in Algorithm 1.

4.3. UT Reasoning

Generating structured text containing numerous bounding box coordinates directly from natural language prompts places a substantial computational burden on the model. To address this challenge, we propose an Unstructured Text (UT) Reasoning approach, inspired by the Chain-of-Thought methodology [30], which first generates detailed natural language descriptions before converting them to structured text.

As illustrated in Fig. 2, VAKER incorporates an explicit reasoning phase that produces comprehensive specifications prior to generating the Banner, Mainground, and Animation ST. This intermediate reasoning step facilitates more precise and contextually appropriate content generation, effectively bridging the gap between natural language prompts and structured layout specifications.

4.4. Dataset construction

Manually converting video advertisements into ST-Representations would require a time-consuming and labor-intensive process, necessitating frame-by-frame annotation of object coordinates, text content, and color attributes. To build a comprehensive dataset efficiently, we develop an automated pipeline Fig. 3 that converts video advertisements into three types of data: Structured Text (ST), Unstructured Text (UT) Reasoning, and natural language prompts. This pipeline extracts key visual information from videos

using computer vision models and converts it into structured representations, which are then transformed into natural language descriptions using language models. The complete pipeline processes each video in under a minute, with technical details provided in the supplementary materials. Our dataset comprises 2,224 real video advertisements, provided specifically for research purposes, originally aired on commercial television broadcasts. Each sample represents a single scene at 1920×1080 resolution and 30 fps, with an average length of 50 frames. The dataset encompasses diverse commercial categories, including restaurants, services, and retail products, capturing a wide spectrum of advertising layouts and visual design patterns.

5. Experiment

5.1. Experiment setting

Implementation details. To train each expert, we finetune the Llama-3.1 70B-Instruct model [21] using LoRA. Our dataset, consisting of 2,224 samples, is split into training (90%) and validation (10%) sets, with each expert trained until validation loss is minimized. Each expert is guided by a task-specific system prompt that includes persona, task description, context, and format specifications, following the approach outlined in [8]. The additional details and complete prompts used for training are provided in the supplementary materials.

For qualitative visualization, we use OpenCV to render the generated ST-Representation, which is formatted as a JSON object. Note that the background images shown in our results are generated from captions in **BG** using off-the-shelf image models: FLUX 1.1 dev [3] and FLUX 1.1 Pro Ultra [2].

Baselines. As VAKER represents the first attempt at text-to-video advertisement generation, no directly comparable baselines exist for the complete framework. To evaluate the two integrated stages of the framework (Banner ST and Mainground ST), we compare against established graphic layout generation models: LayoutPrompter [20] for text-to-layout and constraint-explicit tasks, Desigen [31], and Parse-Then-Place [19]. Additionally, we compare VAKER against state-of-the-art diffusion models, including two text-to-video models (Luma Dream Machine v1.5 [22] and Haiper 2.0 [10]) and two text-to-image models (FLUX 1.1 Pro Ultra [2] and Stable Diffusion 3.5 Large [25]). For the text-to-image models, we evaluate a two-stage approach (T2I + I2V) for video advertisement generation.

Metric for motion. Since our work is the first to generate animated layouts, we propose the Fréchet Motion Distance (FMD) metric. FMD is based on the assumption that high-quality motion should resemble the distribution of motions in the training data. It quantifies this resemblance by comparing relative motion vectors $\Delta b(t) = b(t) - b(0)$ normalized by frame dimensions. Lower FMD values indicate more realistic motion. The formal equation and implementation details are provided in the supplementary material.

Model	FMD ↓	Overlap ↓	mIoU ↑	Failure ↓
LP (T2L) [20]	0.0273	0.5990	0.3172	19.28%
PTP [19]	0.0273	0.4343	0.2600	35.24%
Ours	**0.0103**	**0.4221**	**0.3376**	**5.41%**

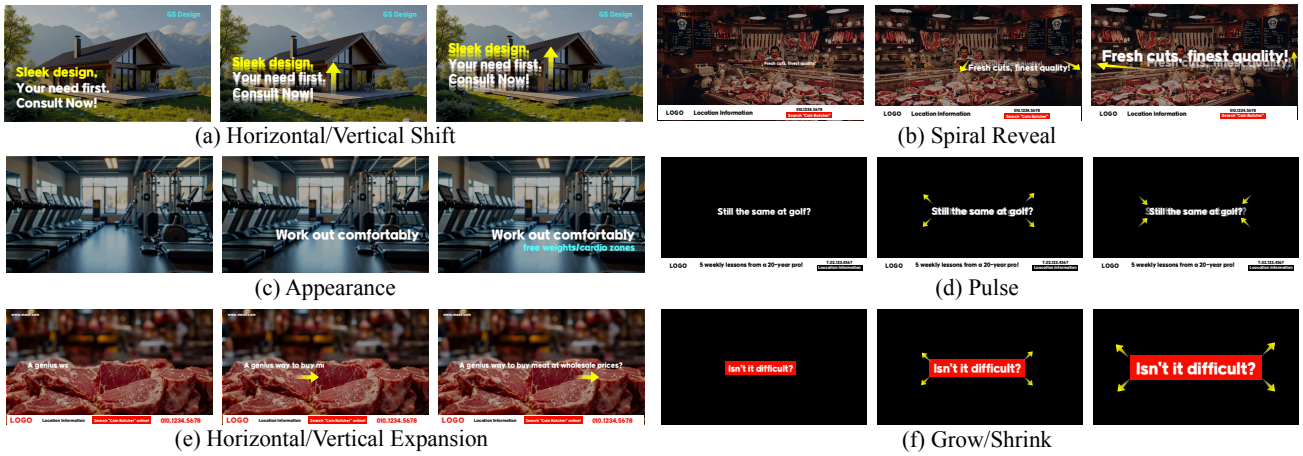
Table 1. **Comparison with text-to-layout baseline models.** Bold indicates best performance.

Layout evaluation metrics. To quantitatively assess layout quality and compare performance with baseline models, we evaluate our framework using three additional metrics: (1) Overlap [18] measures the average Intersection over Union (IoU) between element pairs in a generated layout. Lower values indicate better spacing between elements, ensuring text and visual components remain distinct and readable throughout the video. (2) Maximum IoU (mIoU) [17] calculates the highest IoU between elements in generated and real layouts derived from the same prompt. Higher values indicate better alignment with real-world design practices. (3) Failure rate represents the percentage of generated layouts with invalid structures. This includes JSON parsing failures (malformed JSON syntax), missing required keys (e.g., omitting attributes), incorrect structural hierarchies, missing requested elements, etc.

5.2. Qualitative results

Qualitative analysis. Figs. 1 and 4 demonstrate that VAKER generates high-quality layouts that accurately align with natural language input prompts. As shown in Fig. 1, VAKER successfully converts text descriptions into video advertisements with appropriate element placement and aesthetically pleasing color schemes. Fig. 4 further showcases VAKER's diverse animation capabilities. Interestingly, several of these animation patterns emerged without explicit training examples, demonstrating the model's exceptional generative capabilities.

Comparison with diffusion models. Fig. 5 presents a qualitative comparison of our proposed method against existing diffusion-based models renowned for their typography generation, evaluating their capabilities in generating videos with embedded text and graphical overlays. As shown in the third and fourth columns, T2V models [10, 22] consistently fail to generate clear, legible text, often producing distorted or illegible text within the scene. Similarly, T2I models [2, 25], despite claiming enhanced typography



(a) Horizontal/Vertical Shift (b) Spiral Reveal (c) Appearance (d) Pulse (e) Horizontal/Vertical Expansion (f) Grow/Shrink

Figure 4. **Qualitative examples of animations generated by VAKER.** VAKER demonstrates diverse animation capabilities including (a) horizontal/vertical shift, (b) spiral reveal, (c) appearance, (d) pulse, (e) horizontal/vertical expansion, and (f) grow/shrink.

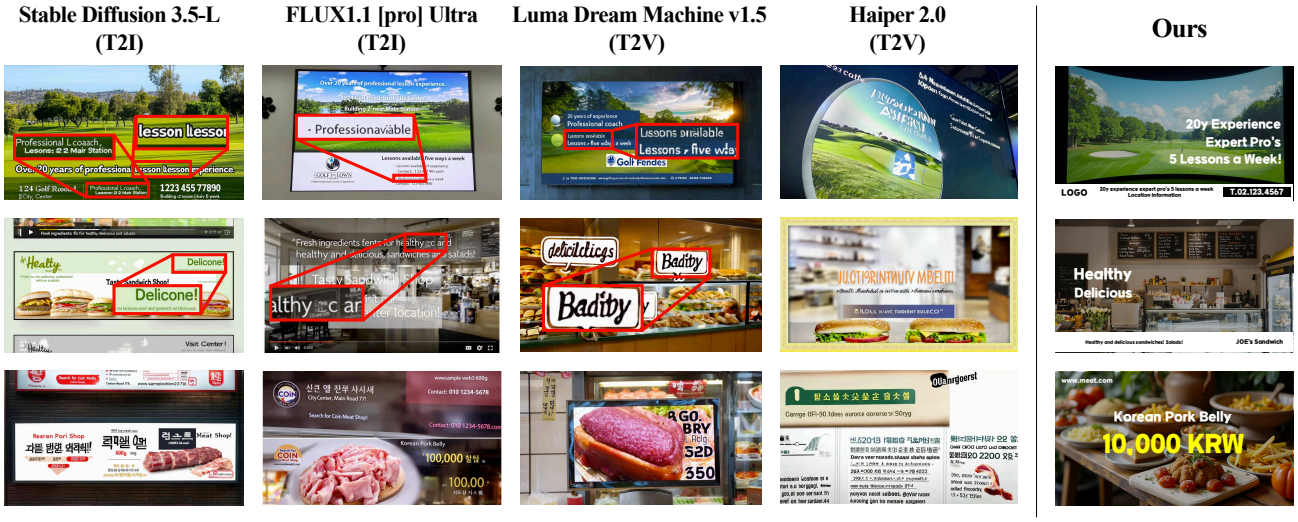


Figure 5. **Qualitative comparisons with diffusion models.** We compare VAKER with text-driven diffusion models (T2I and T2V) for generating videos with the same prompts.

generation capabilities, struggle to produce clear text elements when extensive textual content is required, as demonstrated in the first two rows. In contrast, VAKER excels in producing clear, precise typography by directly encoding text elements within the ST-Representation.

Comparison with layout generation models. Since our work is the first to introduce *dynamic* layout generation, we compare our model against *static* layout generation approaches. Fig. 6 presents a qualitative comparison with state-of-the-art graphic layout generation model baselines. For text-to-layout models, LayoutPrompter (Text-to-Layout) [20] and Parse-Then-Place [19], we provide input prompts identical to those used for our proposed model. For Desigen [31] and LayoutPrompter (Constraint-Explicit), which accept only object class information as input, we provide inputs consisting solely of object classes that align with our prompts. To facilitate direct comparison in visualization, we uniformly render input text in white to match bounding box information. As shown in Fig. 6, our approach demonstrates significant advantages over competing models by effectively incorporating text information. Results indicate superior control over text length, refined color use, and precise layout placement for banner-related elements, highlighting our method's unique strengths in generating well-structured and visually cohesive layouts.

5.3. Quantitative results

Quantitative analysis. Tab. 1 demonstrates that VAKER outperforms all baseline models across all metrics, achieving the lowest FMD and Overlap, highest mIoU, and low-

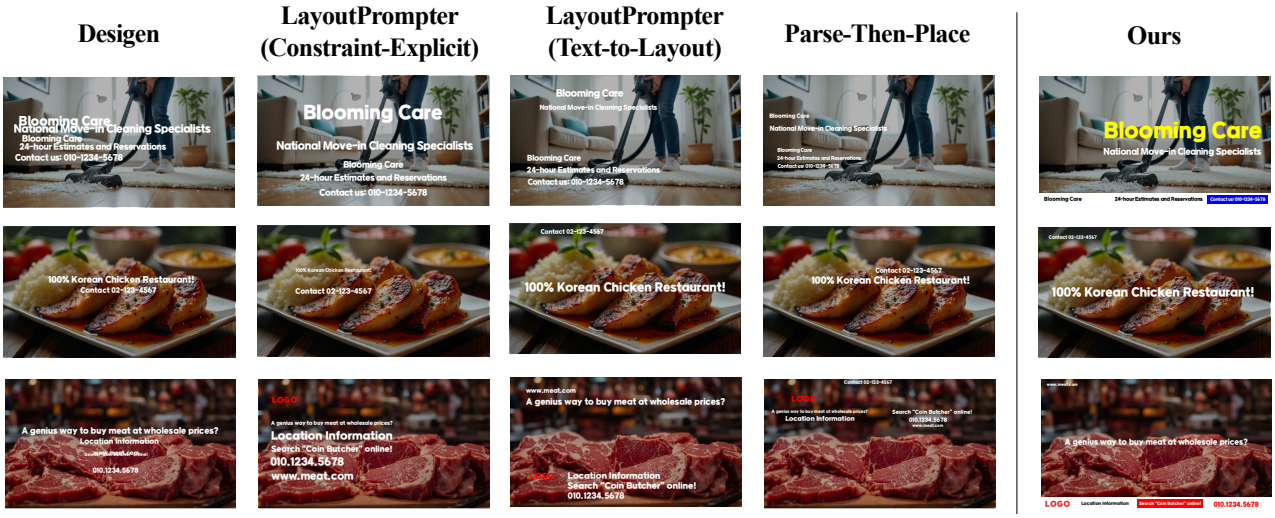


Figure 6. **Qualitative comparison with GLGs.** We compare VAKER with four baseline graphic layout generation models.

est failure rate. Notably, the failure rate of VAKER shows a significant reduction compared to baseline models, with approximately 3 to 6 times fewer instances of invalid layout generation. As expected, LP and PTP are unable to produce motion, resulting in their low (and identical) FMD scores. This substantial improvement highlights VAKER's enhanced ability to maintain proper structure, generate dynamic motion, and consistently produce layouts that faithfully adhere to input prompts.

Setting	FMD ↓	Overlap ↓	mIoU ↑	Failure ↓
Ours	**0.0103**	**0.4221**	**0.3376**	**5.41%**
w/o CoT	0.0272	0.4391	0.2285	8.11%
w/o MoE	0.0196	0.7565	0.2368	7.61%
w/o both	0.0273	0.6401	0.2330	23.33%

Table 2. **Ablation study for layout generation.** Bold indicates best performance.

5.4. Ablation studies

To evaluate the design choices in the proposed pipeline, we conduct ablation experiments on two key components: Chain-of-Thought (CoT) implemented via UT Reasoning (Sec. 4.3), and Mixture-of-Experts (MoE) implemented through the three-stage generation process (Sec. 4.2). We assess their impact using four metrics: FMD, Overlap, mIoU, and Failure rate.

The results presented in Tab. 2 highlight the contributions of both components. The exclusion of CoT results in higher failure rates, reduced layout quality, and degraded motion quality, underscoring its importance in the generation process. Similarly, the absence of MoE notably affects element spacing, as reflected in the Overlap metric, and diminishes motion quality. Removing both components leads to a substantial decline in performance across most metrics, with a particularly significant increase in failure rates and motion quality equivalent to static baselines (Tab. 1). These findings indicate the complementary roles of CoT and MoE in ensuring stable and high-quality layout generation with effective motion.

6. Conclusion

In this work, we introduced Animated Layout Generation, extending traditional layout generation into the temporal domain through our novel Structured Text (ST) Representation. We demonstrated its effectiveness through VAKER, a comprehensive system that employs a three-stage generation pipeline for creating video advertisements with precise control over text, visual attributes, and animated objects. Despite current limitations—such as supporting only simple movements and partial information loss during the conversion from videos to STs—our experimental results demonstrate superior performance compared to both Video Diffusion Models and Graphic Layout Generation approaches. This work establishes a new direction for video generation that enables precise control over both text rendering and animated graphics, opening new possibilities for automated content creation while maintaining control over spatial and temporal elements. Future work could explore more complex animation patterns, dynamic visual effects, and improved information preservation during conversion, further expanding the capabilities of animated layout generation.

References

[1] Diego Martin Arroyo, Janis Postels, and Federico Tombari. Variational transformer networks for layout generation. In *CVPR*, 2021. 2

[2] Black Forest Labs. Flux 1.1 [pro] Ultra. https://blackforestlabs.ai/flux-1-1-ultra, 2024. Last accessed 15 November, 2024. 6

[3] Black Forest Labs. FLUX.1 [dev]. https://huggingface.co/black-forest-labs/FLUX.1-dev, 2024. Last accessed 15 November, 2024. 6

[4] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 2

[5] Digbalay Bose, Rajat Hebbar, Tiantian Feng, Krishna Somandepalli, Anfeng Xu, and Shrikanth Narayanan. Mm-au:towards multimodal understanding of advertisement videos. In *ACM MM*, 2023. 3

[6] Shang Chai, Liansheng Zhuang, and Fengying Yan. Layoutdm: Transformer-based diffusion model for layout generation. In *CVPR*, 2023. 1, 2

[7] Jian Chen, Ruiyi Zhang, Yufan Zhou, Rajiv Jain, Zhiqiang Xu, Ryan Rossi, and Changyou Chen. Towards aligned layout generation via diffusion model with aesthetic constraints. In *ICLR*, 2024. 2

[8] Google. Gemini for Google Workspace Prompting Guide. https://services.google.com/fh/files/misc/gemini-for-google-workspace-prompting-guide-101.pdf, 2024. Last accessed 15 November, 2024. 6

[9] Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S. Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layouttransformer: Layout generation and completion with self-attention. In *ICCV*, 2021. 2

[10] Haiper AI. Haiper 2.0. https://haiper.ai/blog/haiper-2-0, 2024. Last accessed 15 November, 2024. 6

[11] HsiaoYuan Hsu, Xiangteng He, Yuxin Peng, Hao Kong, and Qing Zhang. Posterlayout: A new benchmark and approach for content-aware visual-textual presentation layout. In *CVPR*, 2023. 4

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 4

[13] Mude Hui, Zhizheng Zhang, Xiaoyi Zhang, Wenxuan Xie, Yuwang Wang, and Yan Lu. Unifying layout generation with a decoupled diffusion model. In *CVPR*, 2023. 2

[14] Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. Automatic understanding of image and video advertisements. In *CVPR*, 2017. 3

[15] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Towards flexible multi-modal document models. In *CVPR*, 2023. 2

[16] Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Layoutdm: Discrete diffusion model for controllable layout generation. In *CVPR*, 2023. 1, 2

[17] Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. Constrained graphic layout generation via latent optimization. In *ACM MM*, 2021. 6

[18] Jianan Li, Jimei Yang, Jianming Zhang, Chang Liu, Christina Wang, and Tingfa Xu. Attribute-conditioned layout gan for automatic graphic design. In *TVCG*, 2020. 6

[19] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Weijiang Xu, Ting Liu, Jian-Guang Lou, and Dongmei Zhang. A parse-then-place approach for generating graphic layouts from textual descriptions. In *ICCV*, 2023. 2, 6, 7

[20] Jiawei Lin, Jiaqi Guo, Shizhao Sun, Zijiang James Yang, Jian-Guang Lou, and Dongmei Zhang. Layoutprompter: Awaken the design ability of large language models. In *NeurIPS*, 2023. 2, 6, 7

[21] Meta AI Llama Team. The llama 3 herd of models, 2024. 6, 3

[22] Luma Labs. Dream Machine. https://lumalabs.ai/dream-machine, 2024. Last accessed 15 November, 2024. 6

[23] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024. 3

[24] Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. A vision check-up for language models. In *CVPR*, 2024. 3

[25] Stability AI. Stable Diffusion 3.5 Large. https://huggingface.co/stabilityai/stable-diffusion-3.5-large, 2024. Last accessed 15 November, 2024. 6

[26] Zecheng Tang, Chenfei Wu, Juntao Li, and Nan Duan. Layoutnuwa: Revealing the hidden layout expertise of large language models. In *ICLR*, 2024. 2

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, and Shruti Bhosale et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3

[28] Rejin Varghese and Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *ADICS*, 2024. 5, 3

[29] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, Yingli Zhao, Yulong Ao, Xuebin Min, Tao Li, Boya Wu, Bo Zhao, Bowen Zhang, Liangdong Wang, Guang Liu, Zheqi He, Xi Yang, Jingjing Liu, Yonghua Lin, Tiejun Huang, and Zhongyuan Wang. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 1, 2

[30] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny

Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022. 3, 5

[31] Haohan Weng, Danqing Huang, Yu Qiao, Zheng Hu, Chin-Yew Lin, Tong Zhang, and C. L. Philip Chen. Desigen: A pipeline for controllable design template generation. In *CVPR*, 2024. 6, 7

[32] Kota Yamaguchi. Canvasvae: Learning to generate vector graphic documents. In *ICCV*, 2021. 2

[33] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. In *NeurIPS*, 2021. 5, 3

[34] 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, and Jie Tang. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 1, 2

[35] Keren Ye, Kyle Buettner, and Adriana Kovashka. Story understanding in video advertisements. In *BMVC*, 2018. 3

[36] Xinru Zheng, Xiaorul Qiao, Ying Cao, and Rynson W. H. Lau. Content-aware generative modeling of graphic design layouts. *TOG*, 2019. 2

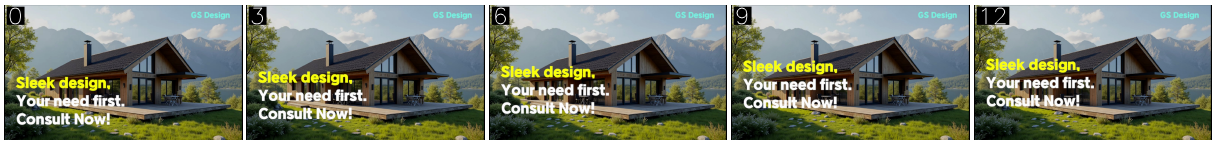
[37] Min Zhou, Chenchen Xu, Ye Ma, Tiezheng Ge, Yuning Jiang, and Weiwei Xu. Composition-aware graphic layout gan for visual-textual presentation designs. In *IJCAI*, 2022. 2

Generating Animated Layouts as Structured Text Representations

Supplementary Material

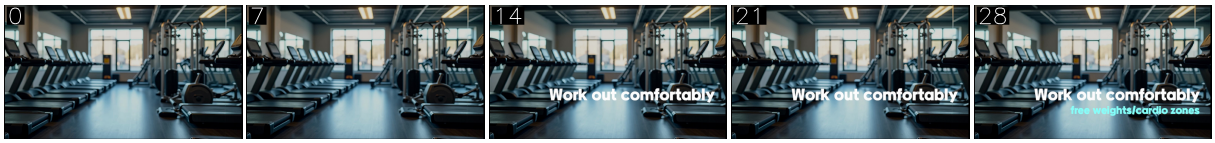
1. Qualitative results

In Figs. 1 and 2, we provide video results generated by VAKER. To effectively visualize the generated animations, we sample frames at intervals of 2 to 5 frames.



(a) The top-right banner features the text “GS Design.” There are no banners at the bottom. The advertisement features a series of text elements over an image of a house with trees and mountains in the background. It begins with “Sleek design,” followed by “Your need first.” and “Consult Now!” promoting expertise in various construction types.

(b) The bottom banner contains a logo and three text elements: “Location Information,” “Search “Coin Butcher”” and “010.1234.5678.” The top-left banner includes a single text stating, “www.meat.com.” There is no top-right banner. The video advertisement presents a scene containing text with the content “Fresh cuts, finest quality!”.



(c) There are no banners. The scene depicts an image of gym equipment. Accompanying this image are two pieces of text, one stating “Work out comfortably” and the other mentioning “free weights/cardio zones.”

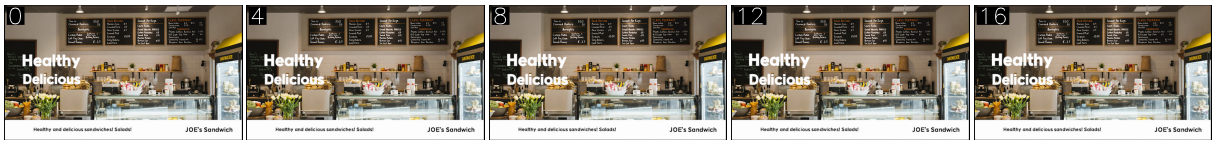
(d) The bottom banner contains a logo and three text elements. The texts include “5 weekly lessons from a 20-year pro!”, “Location Information” and “T.02.123.4567.” There are no banners on the top-left or top-right of the screen. The scene features a single piece of text asking, “Still the same at golf?”.

(e) The bottom banner contains a logo and three text elements: “Location Information” “Search “Coin Butcher” online!” and “010.1234.5678.” The top-left banner includes a single text stating, “www.meat.com” There is no top-right banner. The scene features a text overlay with the message, “A genius way to buy meat at wholesale prices?”. The text appears over an image of raw meat slices.

Figure 1. Videos generated by VAKER. The number on the top-left corner of each frame indicates the frame index.

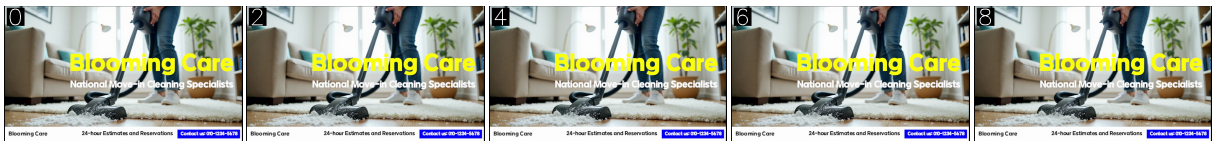
(a) There are no banners. The scene features a piece of text with the content “Isn’t it difficult?”, which is displayed prominently on the screen.

(b) The bottom banner contains a logo and three text elements. The texts include “20y experience expert pro’s 5 lessons a week,” “Location Information,” and “T.02.123.4567.” There are no banners on the top-left or top-right of the screen. The scene features an image of a golf course displayed on a large screen. Superimposed on this background is a text that states “20y Experience,” followed by “Expert Pro’s” signaling training by a certified instructor. Additionally, it highlights “5 Lessons a Week!”.

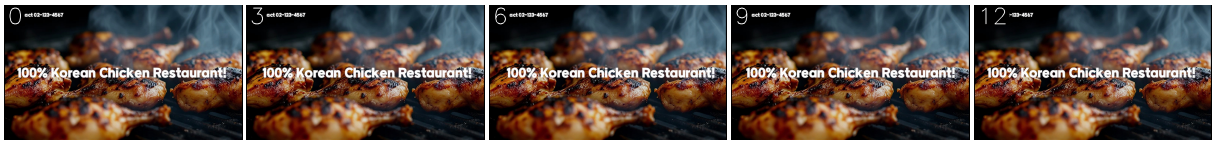


(c) The bottom banner contains two text objects. The first text object reads, “Healthy and delicious sandwiches! Salads!” The second text object says, “JOE’s Sandwich.” There are no banners at the top-left or top-right of the screen. The video advertisement features an interior image of a sandwich shop as its background. It contains text elements with messages like “Healthy” and “Delicious,” presented with distinctive typography.

(d) ”The top-left banner includes a single text stating, “http://www.meat.com.” There is no top-right and bottom banner. The advertisement scene consists of two text objects. The first text displays “Korean Pork Belly.” The second text shows the price of the product as “10,000 KRW.”



(e) The bottom banner contains three text objects: “Blooming Care,” “24-hour Estimates and Reservations,” and “Contact us: 010-1234-5678.” There are no banners at the top-left or top-right of the screen. The advertisement features two text objects. The first text object displays the content “Blooming Care.” The second text object reads “National Move-In Cleaning Specialist.”



(f) The top-left banner contains a text object displaying the message “Contact 02-123-4567.” There are no banners at the bottom or top-right positions. The scene features an image of grilled or cooked chicken in the background. Overlaying the image is text that reads “100% Korean Chicken Restaurant!”. The text presented is clearly visible against the background.

Figure 2. Videos generated by VAKER. The number on the top-left corner of each frame indicates the frame index.

2. Details of VAKER

2.1. Implementation details

Training details. To train each expert, we fine-tune the Llama-3.1 70B-Instruct model [21] using LoRA with a rank of 16 and $\alpha = 32$. The training is performed on 8$\times$ A6000 GPUs, with a learning rate of 2e-4, a batch size of 1, 5 epochs for S-LoRA and T-LoRA, and 8 epochs for B-LoRA. The total training time ranged from 5 to 10 hours, depending on the context length.

2.2. Dataset construction

ST-Representation extraction. The ST-Representation extraction process consists of two main components: spatial and temporal information extraction. Given that object appearances outnumber disappearances in our videos, we adopt reverse-chronological processing starting from the last frame. For spatial extraction, we first analyze this last frame, where all information is encoded through bounding boxes—b_j^{ban} for banners and b_i for objects. To detect and classify visual objects across banner positions $\mathcal{J}$ and object categories $\mathcal{C}$, we implement two fine-tuned YOLOv8 [28] models. These models detect and classify objects into banners, texts, and logos, with banners further classified into positions $j \in \mathcal{J}$ based on their spatial coordinates. We then leverage Vision-Language Models [23] to extract additional attributes: banner colors (y_j^{ban}), background information (y^{bg}, as semantic captions or colors), and object attributes (y_i, including text content and colors).

For temporal information extraction, we employ a pixel-wise tracking model [33] to obtain animation trajectories (a_n^{fg}) of the extracted foreground objects (o_i^{fg}). The pixel-level outputs are transformed into bounding box representations through a post-processing algorithm. Following our reverse-chronological strategy, we track objects backward from the analyzed last frame, which ensures robust tracking by starting from the most complete set of objects.

UT Reasoning and prompt extraction. Using template prompts with LLMs [23] through in-context learning, we generate two types of descriptions from ST-Representations. For UT Reasonings, we convert ST-Representations into template-formatted descriptions. For prompts, we synthesize ST-Representations into natural user-like descriptions of 2-3 sentences. These generated prompts, UT Reasonings, and their corresponding ST-Representations are then used for fine-tuning our model.

YOLO fine-tuning To automate the extraction of bounding boxes from videos, we fine-tune two YOLOv8 [28] detection models—one specialized for detecting banner regions and the other for detecting logos and text. These models are trained on a dataset of 632 video clips annotated by humans, ensuring high-quality ground truth annotations. We employ two distinct models for detection to optimize performance and ensure efficient processing. For the full dataset of 2,224 video clips, the remaining 1,592 clips were annotated using the trained models, leveraging the human-annotated data as a foundation.

Bounding box post-processing We introduce a robust bounding box post-processing algorithm (Algorithm 1) that converts pixel-wise tracking results into rectangular bounding boxes while handling tracking errors. Since pixel-level tracking can contain noise, our algorithm first searches between minimum and maximum coordinates of the mask, employing bidirectional scanning to detect empty rows (all-zero lines). To handle cases where noise causes premature detection of empty rows, we employ early-stop detection with 30-70% thresholds: if an empty row is found too early (above 70% of height in top-down scan or below 30% in bottom-up scan), we trigger additional middle-region verification. The subsequent boundary refinement uses 50% active ratio thresholds, and boxes smaller than 20% of the maximum historical size are excluded for temporal consistency.

3. User study

We conduct a user study to evaluate our results against LayoutPrompter (T2L) [11], which exhibits the lowest failure rate among the baselines. We recruit 30 participants and assign 20 comparison examples, focusing on two criteria: layout quality and advertising effectiveness. Layout quality assesses the aesthetic arrangement of elements, while advertising effectiveness measures the promotional impact. Participants are asked to select the superior result or indicate a draw for each criterion. As shown in Fig. 3, our approach significantly outperforms the state-of-the-art text-to-layout model in both criteria.

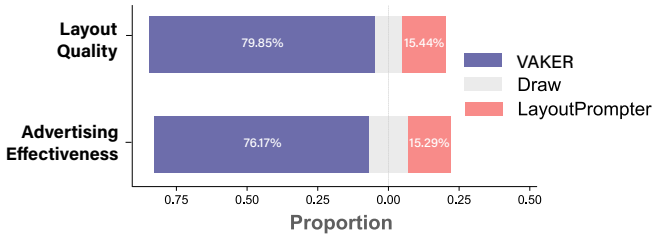


Figure 3. **User study results.** We compare VAKER with LayoutPrompter for Layout Quality and Advertising Effectiveness. VAKER outperforms the baseline in both criteria.

4. Fréchet Motion Distance (FMD)

Fréchet Motion Distance (FMD) quantifies the similarity between distributions of motion trajectories in generated and ground-truth data. This section provides detailed information about our implementation.

4.1. Motion Vector Extraction

Given a sequence of bounding boxes for each object, we extract motion vectors as follows:

For each object, we identify its initial bounding box $b(0) = (x_0, y_0, w_0, h_0)$ at frame $t = 0$. We normalize all coordinates by the frame dimensions (width W and height H):

$$b_{norm}(t) = \left(\frac{x_t}{W}, \frac{y_t}{H}, \frac{w_t}{W}, \frac{h_t}{H} \right) \tag{14}$$

For each subsequent frame $t > 0$, we compute the relative motion vector:

$$\Delta b(t) = b_{norm}(t) - b_{norm}(0) = (\Delta x_t, \Delta y_t, \Delta w_t, \Delta h_t) \tag{15}$$

where $\Delta x_t, \Delta y_t$ capture position changes and $\Delta w_t, \Delta h_t$ capture size changes.

This results in a set of 4-dimensional vectors representing how each object moves relative to its initial position and size.

4.2. Fréchet Distance Computation

Given two sets of motion vectors (from ground truth and generated sequences), we compute the mean vector $\mu \in \mathbb{R}^4$ and covariance matrix $\Sigma \in \mathbb{R}^{4\times4}$ for each distribution. The FMD is then calculated using:

$$\text{FMD} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g}) \tag{16}$$

where μ_r, Σ_r are the mean and covariance of the real data distribution, μ_g, Σ_g are the mean and covariance of the generated data distribution, and $\sqrt{\Sigma_r \Sigma_g}$ is the matrix square root of the product $\Sigma_r \Sigma_g$.

5. ST-Representation for VAKER

We present examples of all ST-Representation components for VAKER: Banner Prompt, Banner UT, Banner ST, Mainground Prompt, Mainground UT, Mainground ST, Animation UT, and Animation ST. For data privacy, all brand names, locations, and numerical information have been modified — numbers are standardized to 012.345.6789 and specific locations are set to 0.

Banner Prompt

The bottom banner contains a logo and three text objects. The texts include "Number One Academy," the contact number "012.345.6789," and the phrase "Songdo High School Exam Specialist." There are no banners in the top-left or top-right positions.

Banner UT

< Bottom >

1. **Color**: The bottom banner has a green background.
2. **Logo**: A logo is located at the bottom-left corner of the banner.
3. **Text**: A text box with the content "Number One Academy" is prominently positioned to the right of the logo. The text is yellow with a transparent background. Following this, a text box containing "012.345.6789" appears, also set against a transparent background with black text. Finally, the text "Songdo High School Exam Specialist" is displayed in white text with a transparent background further along the banner.
4. **Layout**: The bottom banner consists of a logo on the left and three text boxes aligned horizontally. The first text box is adjacent to the logo, followed by the second and third text boxes, offering a clear sequence of the institution's name, contact number, and specialization.

Banner ST


```
{
  "bottom banner": [
    {
      "bbox": [ 0, 979, 1920, 101 ],
      "color": "green",
      "objects": [
        {
          "type": "logo",
          "object_id": 1,
          "bbox": [ 25, 987, 90, 88 ]
        },
        {
          "type": "text",
          "object_id": 2,
          "bbox": [ 123, 991, 354, 77 ],
          "content": "Number One Academy",
          "text_color": "yellow",
          "textbox_color": "transparent"
        },
        {
          "type": "text",
          "object_id": 3,
          "bbox": [ 582, 991, 440, 77 ],
          "content": "012.345.6789",
          "text_color": "black",
          "textbox_color": "transparent"
        },
        {
          "type": "text",
          "object_id": 4,
          "bbox": [ 1052, 993, 848, 79 ],
          "content": "Songdo High School Exam Specialist",
          "text_color": "white",
          "textbox_color": "transparent"
        }
      ]
    }
  ],
  "top-left banner": "None",
  "top-right banner": "None"
}
```


Figure 4. Example 1 for Banner Prompt, Banner UT, and Banner ST

Banner Prompt

The bottom banner features a logo and multiple text objects. The text includes the phrases "Daily PT," which may refer to a company or brand, "0th floor, 00 Sangdae-dong-ro," likely indicating an address, "Sangdae-dong Fitness Center," suggesting a health or fitness related venue, and the phone number "012.345.6789." There are no banners in the top-left or top-right positions.

Banner UT

< Bottom >

1. **Color**: The bottom banner has a black background.
2. **Logo**: A logo is situated near the bottom-left corner of the banner.
3. **Text**: A text box with the content "Daily PT" is centrally positioned nearer to the left side. The text is black with a transparent background. Further towards the center, a text stating "0th floor, 00 Sangdae-dong-ro" is displayed; it is white with a transparent background. To the right, another text box with "Sangdae-dong Fitness Center" is present in white text with a transparent background, followed by the text "012.345.6789" in gold text also with a transparent background.
4. **Layout**: The bottom banner presents a layout with a logo on the far left and four text boxes spread across its width, progressively moving from left to right, each providing distinct information in a clear sequence.

Banner ST


```
{
  "bottom banner": [
    {
      "bbox": [ 0, 984, 1920, 96 ],
      "color": "black",
      "objects": [
        {
          "type": "logo",
          "object_id": 1,
          "bbox": [ 35, 988, 92, 88 ]
        },
        {
          "type": "text",
          "object_id": 2,
          "bbox": [ 163, 1005, 210, 55 ],
          "content": "Daily PT",
          "text_color": "black",
          "textbox_color": "transparent"
        },
        {
          "type": "text",
          "object_id": 3,
          "bbox": [ 929, 1029, 436, 42 ],
          "content": "0th floor, 00 Sangdae-dong-ro",
          "text_color": "white",
          "textbox_color": "transparent"
        },
        {
          "type": "text",
          "object_id": 4,
          "bbox": [ 1189, 993, 177, 35 ],
          "content": "Sangdae-dong Fitness Center",
          "text_color": "white",
          "textbox_color": "transparent"
        },
        {
          "type": "text",
          "object_id": 5,
          "bbox": [ 1466, 999, 411, 65 ],
          "content": "012.345.6789",
          "text_color": "gold",
          "textbox_color": "transparent"
        }
      ]
    }
  ],
  "top-left banner": "None",
  "top-right banner": "None"
}
```


Figure 5. Example 2 for Banner Prompt, Banner UT, and Banner ST

Banner Prompt

The bottom banner features text with the content "$2 discount for takeout orders" and "00-0, Punggeumro 00" The top-left banner contains the text "Contact Us:012-345-6789." The top-right banner displays the text "Pungam Chicken."

Banner UT

< Bottom >

1. **Color**: The bottom banner has a white background.
2. **Logo**: There is no logo present.
3. **Text**: A text box with the content "$2 discount for takeout orders" is displayed prominently in the center with white text on an orange background. Below this, the text "00-0, Punggeumro 00" is presented, featuring black text on a transparent background.
4. **Layout**: The layout consists of two text boxes stacked vertically in the center of the banner, providing promotional and location information.

< Top-left >

1. **Color**: The top-left banner has a black background.
2. **Logo**: There is no logo present.
3. **Text**: A text box with the content "Contact Us: 012-345-6789." is displayed towards the center with white text on an orange background.
4. **Layout**: The banner contains a single text box featuring contact information, centrally aligned.

< Top-right >

1. **Color**: The top-right banner has a black background.
2. **Logo**: There is no logo present.
3. **Text**: A text box with the content " Pungnam Chicken" is displayed centrally with white text on an orange background.
4. **Layout**: The banner features a single text box, centrally aligned, highlighting the business name.

Banner ST


```
{
  "bottom banner": [
    {
      "bbox": [  414, 869, 1067, 211 ],
      "color": "white",
      "objects": [
        {
          "type": "text",
          "object_id": 1,
          "bbox": [ 559, 916, 785, 72 ],
          "content": "$2 discount for takeout orders",
          "text_color": "white",
          "textbox_color": "orange"
        },
        {
          "type": "text",
          "object_id": 2,
          "bbox": [ 692, 1002, 511, 59 ],
          "content": "00-0, Punggeumro 00",
          "text_color": "black",
          "textbox_color": "transparent"
        }
      ]
    }
  ],
  "top-left banner": [
    {
      "bbox": [ 0, 0, 555, 126 ],
      "color": "black",
      "objects": [
        {
          "type": "text",
          "object_id": 1,
          "bbox": [ 15, 31, 504, 65 ],
          "content": "Contact Us: 012-345-6789.",
          "text_color": "white",
          "textbox_color": "orange"
        }
      ]
    }
  ],
  "top-right banner": [
    {
      "bbox": [ 1439, 0, 481, 137 ],
      "color": "black",
      "objects": [
        {
          "type": "text",
          "object_id": 1,
          "bbox": [ 1497, 33, 373, 65 ],
          "content": "Pungnam Chicken",
          "text_color": "white",
          "textbox_color": "orange"
        }
      ]
    }
  ]
}
```


Figure 6. Example 3 for Banner Prompt, Banner UT, and Banner ST

Mainground Prompt

The video advertisement features text stating, "Class is not the end of learning." The text suggests that a lesson is not the end of learning, implying continued education beyond traditional classroom settings.

Mainground UT

1. **Background**: The scene has a solid medium blue background.
2. **Logo**: There is no logo present in the scene.
3. **Text**: A text box with the content "Class is not the end of learning." is displayed. The text is white with a blue background.
4. **Layout**: The overall layout is simple with a medium blue solid background dominating the scene. There is a single text box prominently displayed with its content in white text on a blue background, creating a consistent and cohesive visual aesthetic.

Mainground ST


```
{
  "background": {
    "type": "solid color",
    "content": "mediumblue"
  },
  "objects": [
    {
      "type": "text",
      "object_id": 1,
      "bbox": [ 351, 465, 1225, 130 ],
      "content": "Class is not the end of learning.",
      "text_color": "white",
      "textbox_color": "blue"
    }
  ]
}
```


Figure 7. Example 1 for Mainground Prompt, Mainground UT, and Mainground ST

Mainground Prompt

In the scene, there is a text overlay stating "Clean, modern facilities". Additionally, another text is displayed saying "Wide variety of fitness equipment available!". These texts emphasize the modern and fully equipped nature of the gym depicted in the background image.

Mainground UT

1. **Background**: The scene features an image of a gym with various exercise equipment. This sets an active and energetic atmosphere, typical of a fitness environment.
2. **Logo**: There is no logo present in this scene.
3. **Text**:
   a. A text box with the content "Clean, modern facilities" is displayed. The text is yellow with a transparent background, making it stand out against the gym image.
   b. Another text box with the content "Wide variety of fitness equipment available!" is shown. This text is white with a transparent background, ensuring it is clearly readable over the gym image.
4. **Layout**: The gym's image serves as a dynamic backdrop, with the text elements strategically placed to attract attention. The phrase "Clean, modern facilities" appears above the phrase "Wide variety of fitness equipment available!", both floating against the image's background. The use of contrasting text colors (yellow and white) helps in maintaining readability and emphasis on the promotional messages, integrating well within the gym environment scene.

Mainground ST


```
{
  "background": {
    "type": "image",
    "content": "An image of a gym with exercise equipment"
  },
  "objects": [
    {
      "type": "text",
      "object_id": 1,
      "bbox": [
        983,
        674,
        851,
        128
      ],
      "content": "Clean, modern facilities",
      "text_color": "yellow",
      "textbox_color": "transparent"
    },
    {
      "type": "text",
      "object_id": 2,
      "bbox": [
        734,
        806,
        1100,
        128
      ],
      "content": "Wide variety of fitness equipment
available!",
      "text_color": "white",
      "textbox_color": "transparent"
    }
  ]
}
```


Figure 8. Example 2 for Mainground Prompt, Mainground UT, and Mainground ST

Mainground Prompt

In the scene, there is an image featuring a dish adorned with an abundance of green garnishes. Superimposed on the image is a piece of text that reads "Popular local restaurant," which appears without any background color obstructing it. The text is presented in a white color, adding contrast to the image.

Mainground UT

1. **Background**: The scene has a background image featuring a dish adorned with plenty of green garnishes.
2. **Logo**: No logo is present in this scene.
3. **Text**:
   a. A text box with the content "Popular local restaurant" is displayed toward the center.
   b. The text is white, and the textbox has a transparent background.
4. **Layout**: The overall layout includes a central focus on the background image of the dish with garnishes. The white-text "Popular local restaurant" is prominently placed over the image, ensuring the scene remains uncluttered and visually appealing.

Mainground ST


```
{
  "background": {
    "type": "image",
    "content": "An image of a dish with plenty of green garnishes"
  },
  "objects": [
    {
      "type": "text",
      "object_id": 1,
      "bbox": [
        660,
        469,
        608,
        153
      ],
      "content": "Popular local restaurant",
      "text_color": "white",
      "textbox_color": "transparent"
    }
  ]
}
```


Figure 9. Example 3 for Mainground Prompt, Mainground UT, and Mainground ST

Animation UT

1. **Object ID: 1**
   - **Movement Description**: The object, which is white text "Class is not the end of learning." on a blue textbox background, appears first at frame 0. Initially, it has a bounding box of [351, 465, 504, 130]. As time progresses from frame 0 to frame 8, the bounding box expands horizontally to [351, 465, 1225, 130] and maintains this size and position through frames 16, 24, 32, 40, and 48. There is no vertical or horizontal movement of the entire box's position; only the expansion in width is noted.

Animation ST


```
{
  "scene_length": 48,
  "objects": [
    {
      "object_id": 1,
      "keyframes": [
        {
          "frame": 48,
          "bbox": [ 351, 465, 1225, 130 ]
        },
        {
          "frame": 40,
          "bbox": [ 351, 465, 1224, 130 ]
        },
        {
          "frame": 32,
          "bbox": [ 351, 465, 1225, 130 ]
        },
        {
          "frame": 24,
          "bbox": [ 351, 465, 1225, 130 ]
        },
        {
          "frame": 16,
          "bbox": [ 351, 465, 1225, 130 ]
        },
        {
          "frame": 8,
          "bbox": [ 351, 465, 1225, 130 ]
        },
        {
          "frame": 0,
          "bbox": [ 351, 465, 504, 130 ]
        }
      ]
    }
  ]
}
```


Figure 10. Example 1 for Animation UT and Animation ST

Animation UT

1. **Object ID: 1**
   - **Movement Description**: The object is a yellow text that reads "Clean, modern facilities". It appears from frame 0 and remains in a fixed vertical position (y_min: 674). Initially, starting with a width of 820 in frame 0, it expands slightly to a width of 851 by frame 40. The object maintains the same vertical position throughout.

2. **Object ID: 2**
   - **Movement Description**: The object is a white text that reads "Wide variety of fitness equipment available!". It first appears in frame 0. Initially, the text moves slightly rightward from x_min 952 in frame 0 to x_min 734 by frame 8, indicating a significant horizontal movement to the left as time progresses although it returns to the original x position. The width of the text decreases from 874 in frame 0 to 1072 in frame 8 and then expands slightly up to 1100 by frame 40. The vertical position remains constant at y_min: 806 throughout the frames.

Animation ST


```
{
  "scene_length": 40,
  "objects": [
    {
      "object_id": 1,
      "keyframes": [
        {
          "frame": 40,
          "bbox": [ 983, 674, 851, 128 ]
        },
        {
          "frame": 32,
          "bbox": [ 983, 674, 824, 128 ]
        },
        {
          "frame": 24,
          "bbox": [ 983, 674, 822, 128 ]
        },
        {
          "frame": 16,
          "bbox": [ 983, 674, 851, 128 ]
        },
        {
          "frame": 8,
          "bbox": [ 983, 674, 824, 128 ]
        },
        {
          "frame": 0,
          "bbox": [ 983, 674, 820, 128 ]
        }
      ]
    },
    {
      "object_id": 2,
      "keyframes": [
        {
          "frame": 40,
          "bbox": [ 734, 806, 1100, 128 ]
        },
        {
          "frame": 32,
          "bbox": [ 734, 806, 1100, 128 ]
        },
        {
          "frame": 24,
          "bbox": [ 734, 806, 1075, 128 ]
        },
        {
          "frame": 16,
          "bbox": [ 734, 806, 1075, 128 ]
        },
        {
          "frame": 8,
          "bbox": [ 734, 806, 1072, 128 ]
        },
        {
          "frame": 0,
          "bbox": [ 952, 806, 874, 128 ]
        }
      ]
    }
  ]
}
```


Figure 11. Example 2 for Animation UT and Animation ST

Animation UT

1. **Object ID: 1**
   - **Movement Description**: The object is a text element with the content "Popular local restaurant", appearing from the start at frame 0. Initially, the text is positioned further to the left with a bounding box of [539, 469, 868, 153]. As time progresses to frame 8 and onwards to frame 16, the object moves rightward, settling into a position with a bounding box of [660, 469, 608, 153].

Animation ST


```
{
  "scene_length": 16,
  "objects": [
    {
      "object_id": 1,
      "keyframes": [
        {
          "frame": 16,
          "bbox": [ 660, 469, 608, 153 ]
        },
        {
          "frame": 8,
          "bbox": [ 660, 469, 608, 153 ]
        },
        {
          "frame": 0,
          "bbox": [ 539, 469, 868, 153 ]
        }
      ]
    }
  ]
}
```


Figure 12. Example 3 for Animation UT and Animation ST

Algorithm 1 Bounding Box Post-processing Algorithm

Require: $pred\_mask$, max_values, $is\_firstframe$
Output: $boxes$, max_values

$objects \leftarrow$ unique$(pred_mask) \setminus \{0\}$ ▷ Remove background
for all $obj\_id \in objects$ **do**
$max\_value \leftarrow is\_firstframe \; ? \; 0 : max\_values[obj\_id - 1]$
$mask \leftarrow pred\_mask = obj\_id$
$y\_indices, x\_indices \leftarrow$ where$(mask)$

/* Get vertical bounds via bidirectional scanning */
$y\_max, y\_min \leftarrow \max(y\_indices), \min(y\_indices)$
$y\_mid \leftarrow (y\_max + y\_min)/2$
$(y1\_1, y2\_1) \leftarrow$ ScanTopDown$(mask, y_min, y_max)$ ▷ Check early-stop at 70%
$(y1\_2, y2\_2) \leftarrow$ ScanBottomUp$(mask, y_min, y_max)$ ▷ Check early-stop at 30%
if BothEarlyStop() **and** HasContent$(mask[y_mid, :])$ **then**
$y1, y2 \leftarrow$ ScanFromMiddle$(mask, y_mid)$
else
$y1, y2 \leftarrow$ SelectBestBounds$(y1_1, y2_1, y1_2, y2_2)$
end if

/* Get horizontal bounds */
$x1, x2 \leftarrow$ GetHorizontalBounds$(mask[y1 : y2 + 1, :])$

/* Refine boundaries */
$box\_xy \leftarrow$ ScanXthenY$(mask, x1, y1, x2, y2)$ ▷ 50% ratio
$box\_yx \leftarrow$ ScanYthenX$(mask, x1, y1, x2, y2)$ ▷ 50% ratio
$final\_box \leftarrow$ SelectLargerBox(box_xy, box_yx)

/* Size filtering */
$size \leftarrow (final\_box.x2 - final\_box.x1) \times (final\_box.y2 - final\_box.y1)$
if $is\_firstframe$ **or** $size \geq 0.2 \times max\_value$ **then**
$boxes[obj\_id] \leftarrow final\_box$
if $is\_firstframe$ **then**
max_values.append$(size)$
end if
end if
end for
return $boxes, max\_values$

6. Prompt template

We provide the prompt templates for training B-LoRA (Fig. 13), S-LoRA (Fig. 14), and T-LoRA (Fig. 15).

B-LoRA prompt template

You are an expert advertisement director tasked with designing banners of a 1920x1080 scene including logo and text objects.

Instruction:

1. **Banner Details**: Based on the provided information about banner, describe the bottom, top-left, and top-right banners respectively. Each detail should include the color of the banner, the bbox positions [x_min, y_min, width, height] of the objects (Logo and Text) within the banner, and Layout of the objects.
2. **Banner JSON**: Based on the details, create a JSON object that defines the positions of the banners. Banners may be placed in the top-left, top-right, or bottom of the scene. Each banner contains objects, or is set to None if no banner exists in that position. Ensure that objects are placed within their respective banners. Ensure there is NO overlap between the objects.

Double Check the JSONs: Ensure that NO objects overlap in the scene. Ensure that all bboxes [x_min, y_min, width, height] of the objects are placed within the 1920x1080 image and well-aligned each other. Ensure that the Banner JSON strictly adheres to the valid JSON format.

Banner Info:

{Banner Prompt}

Banner Details:

{Banner UT}

Banner JSON:

```json
{Banner ST}
```

Figure 13. B-LoRA prompt template.

S-LoRA prompt template

You are an expert advertisement director tasked with designing a 1920x1080 scene that includes text and logo objects, given the banner information as JSON.

Instruction:

1. **Image Details**: Using the provided information and the given Banner JSON, describe the rest of the scene with key details about the Background, Logo, Text, and Layout. This description should outline the placement, color and contents of each object.
2. **Image JSON**: Create a JSON object that defines the background and positions of all objects (logo and text) within the 1920x1080 image. Include bounding box position [x_min, y_min, width, height], content, text color, textbox color for each object. Ensure that the logo and text objects do not overlap. Ensure NO objects in the Image JSON are placed in the banner area defined in the Banner JSON.

Double Check the JSONs: Ensure that NO objects overlap in the scene. Ensure that all bboxes [x_min, y_min, width, height] of the objects are placed within the 1920x1080 image. Ensure that Image JSON objects strictly adheres to the valid JSON format.

Info:

{Mainground Prompt}

Banner JSON:

```json
**{Banner ST}**
```

Image Details:

{Mainground UT}

Image JSON:

```json
**{Mainground ST}**
```

Figure 14. S-LoRA prompt template.

T-LoRA prompt template

You are an expert video ad director tasked with animating text and logo objects in a 1920x1080, 30fps, and a few seconds video ad.

Instruction:

You are provided Image JSON object that outlines the object positions all objects (logo and text) accordingly.

1. **Animation Description**: Using the provided Image JSON, describe how each object (logo and text) should move. For each Object ID, provide a clear Animation Description to make the object move smoothly and realistically.
2. **Video JSON**: Create a JSON object detailing the scene length and track information for each object. Include the bounding box position [x_min, y_min, width, height] for every 8 keyframes, in a reverse order.

Ensure that the Video JSON strictly adheres to the valid JSON format. Ensure that the objects are placed within the 1920x1080 video frame.

Image JSON:

```json
**{Mainground ST}**
```

Animation Description:

{Animation UT}

Video JSON:

```json
**{Animation ST}**
```

Figure 15. T-LoRA prompt template.