Visual Executability: Enabling Language Models to Produce Executable Cinematic Plan

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated strong capabilities in narrative understanding and generation, enabling applications such as script drafting and story planning. With the rapid rise of AI-generated web dramas and short-form narrative videos, there is increasing demand for systems that can translate narrative text into precise and controllable visual instructions at scale. However, LLM outputs are typically free-form and unstructured, making them difficult to integrate into downstream visual generation pipelines. While recent multimodal models attempt to directly map text to images or videos, such end-to-end approaches often lack interpretability, controllability, and compatibility with practical cinematic workflows.

In this work, we introduce *visual executability*, a property of language model outputs that enables deterministic execution by downstream visual systems. We propose a schema-guided framework that converts narrative text into shot-level executable cinematic plans, explicitly encoding scene context, character presence, emotional states, and cinematographic parameters such as shot type, camera movement, composition, and perspective. By constraining language model generation with a strict JSON schema, our approach guarantees structural validity, atomicity, and completeness of each visual unit, without requiring access to visual data or retraining multimodal models.

Through quantitative and qualitative evaluations on narrative texts, we demonstrate that our method produces fine-grained, coherent, and structurally valid cinematic plans that are directly usable for storyboarding and automated image or video generation pipelines. Our results suggest that enforcing visual executability at the language level provides a principled and modular alternative to end-to-end multimodal generation for controllable visual content creation.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in narrative understanding and text generation, enabling applications such as creative writing, story summarization, and script drafting. Recent studies further show that LLMs can perform high-level reasoning and planning over complex semantic structures (Valmeekam et al., 2023; Guan et al., 2023; Pallagani et al., 2024). In parallel, AI-generated web dramas and short-form narrative videos have rapidly gained popularity, leading to the emergence of large-scale, fast-iteration content production pipelines. In these settings, narrative scripts must be translated into visual content efficiently and consistently, placing increasing pressure on automated systems to produce precise and controllable visual plans. Despite these advances, LLM outputs remain fundamentally non-executable with respect to visual generation systems. Bridging the gap between narrative language and controllable visual realization therefore remains a central challenge, particularly for applications such as storyboarding, film previsualization, and automated video generation.

Most existing approaches address this gap in one of two ways. Prompt-based methods rely on carefully engineered natural language prompts to guide text-to-image or text-to-video models. While flexible, these approaches produce free-form descriptions that lack explicit structure, making them difficult to verify, reuse, or systematically control (Zhou et al., 2023b). Alternatively, end-to-end multimodal models attempt to directly map narrative text to visual outputs through joint training on large-scale datasets. Although effective in certain scenarios, such models operate largely as black boxes and provide limited interpretability or explicit control over cinematographic elements such as shot boundaries, camera motion, and composition (He et al., 2023; Zhou et al., 2024; Yang et al.,

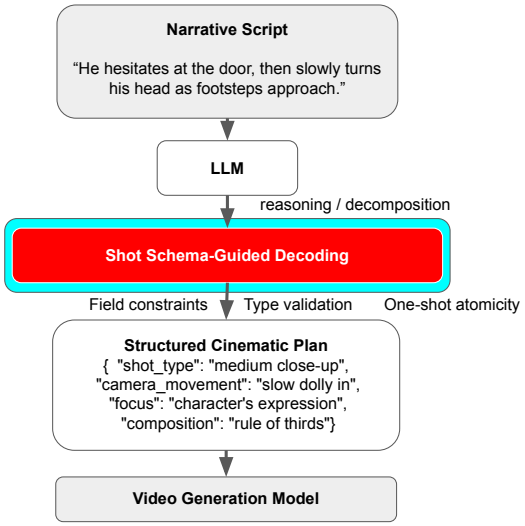

Figure 1: Overview of visual executability. Given a narrative script, a language model generates a structured cinematic plan through schema-guided decoding, enforcing field constraints, type validation, and one-shot atomicity. The resulting plan is directly executable by downstream video generation models.

2024).

In contrast, professional cinematic workflows do not generate visual content holistically. Instead, narratives are planned and executed at the level of shots—atomic visual units that specify what is shown, how it is framed, and how the camera moves. This shot-based planning paradigm underlies storyboards, shot lists, and pre-visualization pipelines, and relies on a shared visual language used by directors, cinematographers, and editors. Although recent work has explored language-based planning and action generation (Wang et al., 2023c; Zhou et al., 2023a), current language models do not natively produce outputs aligned with this shot-level visual representation.

Figure 1 summarizes the overall framework of our approach. Rather than generating free-form prompts or directly synthesizing visual outputs, we introduce an explicit intermediate representation in the form of a structured cinematic plan. Given a narrative script, a language model performs semantic reasoning and decomposition, while its generation process is constrained through schema-guided decoding. The schema enforces field completeness, type validity, and one-shot atomicity, resulting in a sequence of shot-level specifications that can be deterministically consumed by downstream image or video generation models.

Based on this formulation, we argue that the key limitation of existing methods lies not in language understanding itself, but in the absence of an explicit intermediate representation that bridges narrative semantics and executable visual control. To address this limitation, we introduce the concept of *visual executability*, which characterizes whether a language model's output can be deterministically mapped to a sequence of visual operations without additional interpretation or heuristic post-processing. Our notion of visual executability is closely related to recent advances in structured generation and constrained decoding (Wang et al., 2023a; Geng et al.; Zheng et al., 2024), but is specifically grounded in cinematic planning and visual realization.

We propose a schema-guided framework that transforms narrative text into executable cinematic plans composed of shot-level representations. Each shot is modeled as an atomic visual unit with explicitly defined scene context, character information, emotional tone, and cinematographic parameters such as shot type, composition, perspective, and camera movement. By enforcing a formal schema during generation, we guarantee structural validity by construction, eliminating the ambiguity inherent in free-form language.

Importantly, our approach does not require visual supervision or retraining of multimodal generation models. Instead, it leverages the implicit cinematic knowledge encoded in large-scale narrative corpora and makes this knowledge operational through structured generation. In this way, we shift the role of language models from passive narrators to active visual planners, capable of producing outputs that are both semantically grounded and operationally executable.

We evaluate our framework on narrative-to-cinematic planning tasks and demonstrate that it consistently produces fine-grained, coherent, and structurally valid cinematic plans. Human evaluations conducted with professional AI short-drama creators and film-directing students further indicate that the generated plans are directly usable for storyboarding and integration into automated image and video generation pipelines. Overall, our results suggest that visual executability provides a principled and modular alternative to end-to-end multimodal generation, enabling controllable and interpretable visual content creation driven by language models.

2 Related work

2.1 Language Models for Planning and Executable Representations

Recent studies have investigated the use of large language models (LLMs) for planning and decision-making, focusing on their ability to generate intermediate representations that support executable actions. A number of works analyze the planning capabilities of LLMs, revealing both their potential and inherent limitations in symbolic reasoning and long-horizon planning (Valmeekam et al., 2023; Pallagani et al., 2024). Other approaches leverage LLMs to construct explicit world models or planning domains that can be integrated into classical or model-based planning frameworks (Guan et al., 2023; Oswald et al., 2024).

Beyond symbolic planning, LLMs have been embedded into interactive or agent-based systems that unify reasoning, planning, and action selection in open-world environments (Wang et al., 2023c; Zhou et al., 2023a). Closely related work in embodied and language-to-action planning grounds language outputs in executable physical actions (Raman et al., 2022; Wu et al., 2023; Gramopadhye and Szafir, 2023). These methods define executability with respect to environment dynamics and task completion.

In contrast, our work focuses on visual executability rather than physical or symbolic action execution. We treat cinematic constructs—such as shots, camera movements, and framing—as executable units, and frame narrative understanding as a planning problem whose output is an intermediate representation designed specifically for visual realization.

2.2 Structured and Constrained Language Generation

Structured generation has emerged as a key technique for improving the reliability and controllability of language model outputs. Recent work evaluates schema- and grammar-constrained decoding, showing that formal constraints such as JSON schemas can significantly improve structural validity while maintaining output quality (Geng et al.; Wang et al., 2023a). Related approaches apply schema-guided generation to information extraction tasks, including scientific text and non-standardized tables, demonstrating that strict structural constraints enable robust downstream processing (Dagdelen et al., 2024; Hu et al., 2025).

More broadly, constrained and controlled text generation methods aim to steer LLM outputs using explicit instructions or structural requirements (Zhou et al., 2023b; Meng et al., 2022). Systems such as SGLang further emphasize the execution of structured language programs, highlighting the benefits of treating LLM outputs as executable artifacts rather than free-form text (Zheng et al., 2024). Feature Execution Graphs similarly formalize LLM outputs as executable graphs to support human–AI co-programming (Batatia and Svetlichnyi, 2025).

Our work builds on this line of research but departs in its objective. Rather than enforcing structure for semantic correctness or program execution, we impose schema constraints to guarantee visual executability, ensuring that each generated unit corresponds to an atomic and complete cinematic operation.

2.3 Visual Planning and Multimodal Generation from Language

Language-conditioned visual generation has been widely studied in both image and video domains. End-to-end story-to-image and story-to-video systems aim to directly synthesize visual content from narratives, often using diffusion-based or retrieval-augmented frameworks (He et al., 2023; Zhou et al., 2024; Yang et al., 2024). While these methods demonstrate impressive visual fidelity, they typically lack explicit intermediate representations for shot structure or cinematographic intent, limiting controllability and editability.

More recent work explores compositional and modular approaches to visual generation. LayoutGPT shows that LLMs can generate structured visual plans, such as layouts, that guide downstream image synthesis (Feng et al., 2023). Other studies emphasize the role of intermediate representations in multimodal systems, suggesting that structured control signals can simplify alignment and improve generation quality (Chen et al., 2025; Wang et al., 2023b).

Our approach aligns with these efforts but targets a different level of abstraction. Rather than generating spatial layouts or visual outputs directly, we propose a shot-level cinematic plan as an intermediate representation that captures temporal structure, visual focus, and cinematographic operations. This positions our work as a bridge between narrative language understanding and controllable multimodal generation.

Prior work has explored planning with language

models, structured generation, and multimodal synthesis from complementary perspectives. Our work differs by introducing visual executability as a unifying principle and by framing cinematic planning as a schema-constrained language generation problem. This enables language models to produce intermediate representations that are both semantically grounded and directly executable by visual generation pipelines.

3 Method

3.1 Problem Formulation

We study the problem of converting narrative text into a representation that can be deterministically executed by downstream visual generation systems. Let $x \in \mathcal{X}$ denote an input narrative text. Our goal is to learn a mapping

$$f : \mathcal{X} \to \mathcal{P}, \tag{1}$$

where

$$\mathcal{P} = (s_1, s_2, \ldots, s_N) \tag{2}$$

is an ordered sequence of cinematic shots. Each shot s_i represents an atomic visual unit that can be independently executed by a visual generation pipeline.

3.2 Visual Executability

We introduce *visual executability* as a property of language model outputs.

Definition (Visual Executability). A representation $\mathcal{P}$ is said to be *visually executable* if there exists a deterministic mapping

$$g : \mathcal{P} \to \mathcal{A}, \tag{3}$$

where $\mathcal{A}$ denotes a sequence of executable visual operations, such that no additional heuristic interpretation is required.

In our setting, visual executability is operationalized via the following criteria:

- **Atomicity:** each s_i corresponds to a single visual focus (e.g., one action, one reaction, or one observation).
- **Completeness:** each s_i explicitly specifies all attributes required for execution.
- **Discreteness:** visual attributes are drawn from predefined finite sets of primitives.
- **Structural Validity:** $\mathcal{P}$ conforms to a formally defined schema that can be automatically validated.

3.3 Shot-Level Cinematic Representation

We model a cinematic plan as a sequence of shots, where each shot is treated as an *atomic visual action*. Specifically, each shot s_i is represented as a structured tuple:

$$s_i = \langle C_i, V_i, M_i, D_i \rangle, \tag{4}$$

where

$$\begin{array}{ll} C_i \in \mathcal{C} & \text{(scene context)}, \\ V_i \in \mathcal{V} & \text{(visual focus and affect)}, \\ M_i \in \mathcal{M} & \text{(cinematographic parameters)}, \\ D_i \in \mathcal{D} & \text{(dialogue)}. \end{array} \tag{5}$$

Intuitively, C_i captures contextual factors such as time and environment; V_i captures the visual focus (e.g., characters present) and the intended affect; M_i encodes cinematographic primitives such as shot type, camera perspective, composition, and camera movement; and D_i stores dialogue when present. This shot-level abstraction matches practical cinematic workflows and supports fine-grained decomposition of narratives into executable units.

We define a shot-level schema that specifies the required structure and attributes of each cinematic shot. Formally, each shot s_i must include the following fields:

- **Scene**: a textual description of the scene context (e.g., time, location, environment).
- **Characters**: a list of entities appearing in the shot.
- **Mood**: a categorical variable describing the emotional tone of the shot.
- **Shot Type**: a discrete cinematographic primitive (e.g., close-up, medium shot, long shot).
- **Composition**: a description of spatial composition (e.g., rule of thirds, centered).
- **Perspective**: the camera viewpoint (e.g., eye-level, low-angle, high-angle).
- **Camera Movement**: a discrete camera operation (e.g., static, pan, tilt, dolly).
- **Focus**: the primary visual focus of the shot.
- **Dialogue**: spoken content, if present.

All fields except *Dialogue* are required. Each field is associated with a predefined type and, where applicable, a finite set of admissible values.

3.4 Schema-Guided Generation

To guarantee visual executability, we constrain generation to a formally defined schema. Let $\mathcal{S}$ denote a predefined schema specifying required fields, types, and admissible values for each shot and for the overall plan. We generate shots under schema constraints:

$$s_i \sim p_\theta(s \mid x, \mathcal{S}), \tag{6}$$

where p_θ is restricted to outputs that satisfy $\mathcal{S}$. Equivalently, we enforce the hard constraint

$$\forall s_i \in \mathcal{P}, \quad s_i \in \mathcal{S}, \tag{7}$$

which ensures structural validity by construction and eliminates structurally invalid generations.

Compared to unconstrained free-form generation, schema-guided decoding shifts the model from descriptive narration to structured cinematic planning, producing outputs that can be deterministically mapped to downstream visual operations.

3.5 Implementation Note

In practice, the schema can be instantiated as a strict JSON schema with enumerated cinematographic primitives and required fields. This enables automatic validation of $\mathcal{P}$ and supports modular integration with downstream storyboard tools and automated image/video generation pipelines.

4 Experiments

4.1 Experimental Setup

Tasks and Data. We evaluate cinematic planning on story-to-video planning tasks, where the input is a short narrative description and the output is a cinematic plan. We construct a prompt set covering diverse genres and scene types (e.g., dialogue-heavy, action-centric, interior/exterior, day/night). All methods are evaluated on the same prompt set to ensure comparability.

Language Model Backbones. All planning-based methods are instantiated with open-source instruction-tuned language models. We report results under three representative 7B-scale backbones: Qwen-7B-Instruct, LLaMA-2-7B-Chat, and Mistral-7B-Instruct. For each backbone, all compared methods and ablation variants use the same model to control for capacity.

Compared Methods and Output Alignment. We compare three representative paradigms: (i) prompt-based text-to-video planning, (ii) workflow-based planning with node-level structure, and (iii) free-form LLM planning without explicit schema constraints. Since our focus is planning quality rather than rendered videos, we evaluate the generated *plans*. For baselines that produce free-form prompts or workflow descriptions, we apply a deterministic field-mapping to extract comparable shot-level attributes when possible; otherwise, missing attributes are counted as missing fields by definition.

4.2 Automatic Evaluation: Structural Validity and Executability

We first evaluate whether generated outputs satisfy the requirements of visual executability. This evaluation is fully automatic and independent of human judgment.

Metrics. We report the following metrics:

- **Schema Validity Rate (%):** The percentage of outputs that fully conform to the predefined schema, including required fields, data types, and admissible values.
- **Missing Field Rate (%):** The proportion of shots that omit at least one required attribute.
- **Atomicity Violation Rate (%):** The proportion of shots that contain multiple visual foci, actions, or camera operations within a single shot.

Operational Definition and Examples. An output is considered schema-valid if and only if it fully conforms to the predefined cinematic schema. For example, a shot specification that includes all required fields such as shot type, composition, camera movement, and focus, with correct types and values, is counted as valid. In contrast, free-form textual descriptions or partially structured outputs that omit required attributes or mix multiple actions in a single shot are counted as invalid.

Missing fields are detected by checking the absence of any required attribute in the schema. Atomicity violations occur when a single shot description combines multiple actions or camera movements that would correspond to multiple cinematic shots in professional practice (e.g., "the character enters the room, the camera pans, and another character reacts").

Results. Table 1 summarizes the structural validity results across different planning paradigms.

We note that schema validity reaches 100% for our method across all backbones because invalid outputs are prevented *by construction* via schema-guided decoding rather than filtered post hoc. Therefore, differences across backbones mainly manifest in higher-level content choices (e.g., phrasing or stylistic preferences), while structural validity remains guaranteed.

4.3 Human Evaluation: Cinematic Plan Quality

To assess the practical usefulness of generated cinematic plans, we conduct a human evaluation study focusing on real-world creative workflows.

Participants. We recruit two groups of evaluators:

- Professionals from an AI short-drama production company.
- Graduate students majoring in film directing and cinematography.

All participants have prior experience with cinematic planning or storyboard creation.

Protocol and Statistics. We conduct a randomized, anonymized evaluation in which each participant rates outputs from all methods for the same set of inputs. To reduce order effects, we randomize the presentation order per input. We report mean Likert scores aggregated across participants and prompts.

Evaluation Criteria. Evaluators rate each method on a 5-point Likert scale (1: very poor, 5: excellent) along three dimensions:

- **Visual Clarity:** How clearly the plan specifies visual content and camera intent.
- **Shot Coherence:** How well shots are temporally and semantically organized.
- **Direct Usability:** How easily the plan can be directly used in storyboarding or video generation pipelines with minimal modification.

Results. Human evaluation results are reported in Table 2.

Discussion. Table 2 shows consistent improvements from our method across all three backbones. A key observation is that prompt-based T2V outputs are often visually evocative but operationally underspecified: evaluators reported that important cinematographic decisions (e.g., shot boundary, camera movement, composition intent) are either implicit or entangled in long descriptive sentences, making direct translation to storyboards labor-intensive. Free-form LLM planning improves narrative coherence but still tends to conflate multiple actions or camera intents in a single "shot-like" paragraph, which reduces shot coherence and direct usability.

Workflow-based planning partially mitigates these issues by introducing intermediate structure, leading to better coherence than free-form baselines. However, evaluators noted that workflow outputs frequently require manual completion of missing fields and normalization of terminology before they can be executed reliably. In contrast, our schema-guided shot-level representation produces plans that are both complete and granular, which directly supports downstream use: professionals reported that the output can be pasted into their storyboard templates with minimal edits, while film students found the shot boundaries and camera intent easier to interpret and critique.

Finally, the cross-backbone results suggest that backbone choice mainly affects stylistic preference (e.g., brevity vs. elaboration), but the relative ranking remains stable. This aligns with our hypothesis that visual executability is primarily a representation and constraint design problem rather than a model-size effect.

4.4 Ablation Study

We conduct ablation studies to analyze the contribution of shot-level decomposition and schema constraints.

Variants. We compare the following variants:

- **Scene-Level Only**: Coarse scene-level planning without shot decomposition.
- **Shot-Level, No Schema**: Shot-level planning without schema constraints.
- **Shot-Level + Schema (Full)**: The full proposed model.

Results. Table 3 presents the ablation results.

Method	Qwen-7B-Instruct			LLaMA-2-7B-Chat			Mistral-7B-Instruct		
	SV	MF	AV	SV	MF	AV	SV	MF	AV
Prompt-based T2V	2.1	48.6	61.3	1.7	51.2	64.8	2.4	46.9	59.7
Workflow-based T2V	6.4	27.9	34.5	5.8	29.4	36.1	6.9	26.7	33.2
LLM Planning (Free-form)	4.8	36.2	42.7	4.1	38.9	45.3	5.2	34.6	40.8
Ours (Visual Executability)	**100.0**	**0.0**	**3.1**	**100.0**	**0.0**	**4.2**	**100.0**	**0.0**	**3.8**

Table 1: Automatic evaluation across different language model backbones. SV: Schema Validity (%), MF: Missing Fields (%), AV: Atomicity Violations (%).

Method	Qwen-7B-Instruct			LLaMA-2-7B-Chat			Mistral-7B-Instruct		
	VC	SC	DU	VC	SC	DU	VC	SC	DU
Prompt-based T2V	2.7	2.4	1.9	2.5	2.2	1.8	2.8	2.5	2.0
Workflow-based Planning	3.4	3.6	3.2	3.2	3.4	3.0	3.5	3.7	3.3
LLM Planning (No Schema)	3.1	2.9	2.5	2.9	2.7	2.3	3.2	3.0	2.6
Ours (Visual Executability)	**4.3**	**4.5**	**4.6**	**4.1**	**4.3**	**4.4**	**4.2**	**4.4**	**4.5**

Table 2: Human evaluation results across different language model backbones (5-point Likert scale). VC: Visual Clarity, SC: Shot Coherence, DU: Direct Usability.

Discussion. The ablation results in Table 3 reveal two complementary factors behind visual executability. First, removing shot-level decomposition (Scene-Level Only) substantially reduces shot consistency and usability, indicating that coarse scene descriptions do not provide sufficient granularity for execution: evaluators reported ambiguous shot boundaries and unclear camera intent, which prevents direct storyboard translation. Second, keeping shot-level structure but removing schema constraints (Shot-Level, No Schema) improves consistency compared to scene-level outputs, yet still yields structurally unstable plans with missing or inconsistent attributes, resulting in lower usability.

The full model combines shot-level granularity with hard schema constraints and achieves the best performance under all backbones. This confirms that visual executability is not obtained by adding format alone; rather, it emerges from jointly enforcing (i) an execution-oriented unit of planning (shots) and (ii) a verifiable interface (schema) that eliminates underspecification.

5 Qualitative Analysis

We present a qualitative example to illustrate how visual executability enables the generation of concrete and controllable cinematic plans, and how these plans translate into executable video outputs.

5.1 Case Study: From Story to Executable Video

Input Story. *"A man waits alone at a bus stop late at night. Rain begins to fall. He checks his phone repeatedly, growing more anxious as time passes."*

Generated Cinematic Plan. Using our method, the language model produces the following shot-level plan (abridged for brevity):

```
Shot 1:
  Shot Type: Wide shot
  Composition: Rule of thirds
  Camera Movement: Static
  Focus: Empty bus stop under streetlight,
  light rain

Shot 2:
  Shot Type: Medium shot
  Composition: Centered
  Camera Movement: Slow dolly in
  Focus: Man standing alone, checking his phone

Shot 3:
  Shot Type: Close-up
  Composition: Tight framing
  Camera Movement: Static
  Focus: Man's face, rain droplets visible,
  anxious expression
```

The plan is fully schema-valid and specifies shot boundaries, camera intent, and visual focus for each shot, enabling direct execution without additional interpretation.

Executable Image Output. To qualitatively demonstrate visual executability, we instantiate the generated cinematic plan using representative image outputs rather than full video generation. Specifically, we render one image per selected shot using an open-source text-to-image diffusion model conditioned on the corresponding shot specification.

Figure 2 visualizes three shots generated from a single story input. Each image corresponds to one shot in the plan and reflects the intended shot type, composition, and visual focus. This example illustrates that the proposed shot-level schema produces

Variant	Qwen-7B-Instruct			LLaMA-2-7B-Chat			Mistral-7B-Instruct		
	SV	HU	SC	SV	HU	SC	SV	HU	SC
Scene-Level Only	1.3	2.4	1.9	0.9	2.2	1.7	1.5	2.5	2.0
Shot-Level, No Schema	5.1	3.2	3.4	4.6	3.0	3.2	5.4	3.3	3.5
Shot-Level + Schema (Full)	**100.0**	**4.6**	**4.5**	**100.0**	**4.4**	**4.3**	**100.0**	**4.5**	**4.4**

Table 3: Ablation study across different language model backbones. SV: Schema Validity (%), HU: Human Usability, SC: Shot Consistency.

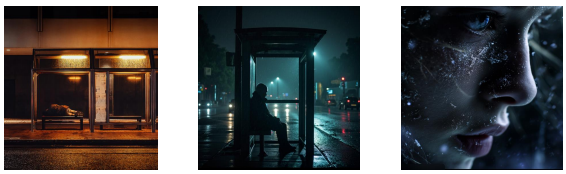

Figure 2: Executable image outputs corresponding to three shots in a generated cinematic plan. From left to right: (Shot 1) establishing wide shot of the environment, (Shot 2) medium shot focusing on the character in context, and (Shot 3) close-up emphasizing emotional detail. Each image is rendered independently from its shot specification, demonstrating the executability of the plan at the shot level.

plans that can be directly translated into concrete visual realizations without additional interpretation or manual refinement.

5.2 Summary of Experimental Findings

Across automatic evaluation, human studies, ablation experiments, and backbone robustness analysis, results consistently demonstrate the effectiveness of enforcing visual executability. Shot-level schema constraints lead to structurally valid, coherent, and practically usable cinematic plans. These findings validate visual executability as a core design principle for language-driven cinematic planning systems.

6 Conclusion

We introduced visual executability as a design principle for language-driven cinematic planning. By representing plans at the shot level and enforcing schema-guided constraints during generation, our approach produces cinematic plans that are structurally valid, semantically coherent, and directly executable in real-world creative workflows.

Through automatic evaluation, human studies, ablation experiments, and qualitative analysis, we show that visual executability substantially improves both the reliability and usability of generated plans compared to prompt-based, workflow-based, and free-form planning baselines. Importantly, these improvements are consistent across multiple open-source language model backbones, indicating that executability arises from representation and constraint design rather than model-specific behavior.

We believe that visual executability provides a principled interface between language models and visual generation systems. Beyond cinematic planning, this paradigm opens up new opportunities for controllable multimodal generation, structured creative assistance, and human-AI collaboration in visual storytelling and content creation.

Limitations

While our results demonstrate the effectiveness of visual executability for cinematic planning, several limitations remain.

First, our framework relies on a predefined cinematic schema that encodes common shot-level attributes such as shot type, composition, and camera movement. Although this schema covers a wide range of standard cinematic practices, it may not fully capture highly unconventional or experimental filmmaking styles. Extending the schema to support richer stylistic variation or alternative cinematic grammars is an important direction for future work.

Second, while schema-guided decoding guarantees structural validity, it does not by itself ensure aesthetic quality or creative originality. As illustrated by our qualitative examples, generated plans are executable and well-structured, but artistic appeal still depends on downstream rendering models and subjective human preferences. Balancing strict executability constraints with creative flexibility remains an open challenge.

Third, our human evaluation focuses on usability and clarity in professional workflows, rather than long-term narrative quality or audience reception. Evaluating how executable cinematic plans influence end-user engagement or storytelling effectiveness over longer video sequences is beyond the scope of this work.

Finally, although we evaluate our method across multiple open-source language model backbones, all experiments are conducted at a similar model scale. Investigating how visual executability interacts with larger models or multimodal foundation models is left for future research.

References

Hadj Batatia and Ilia Svetlichnyi. 2025. Feature execution graphs: A human-ai co-programming paradigm for graph-driven llm code synthesis. In *Proceedings of the AAAI Symposium Series*, volume 6, pages 184–191.

Liang Chen, Shuai Bai, Wenhao Chai, Weichu Xie, Haozhe Zhao, Leon Vinci, Junyang Lin, and Baobao Chang. 2025. Multimodal representation alignment for image generation: Text-image interleaved control is easier than you think. *arXiv preprint arXiv:2502.20172*.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature communications*, 15(1):1418.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250.

Saibo Geng, Hudson Cooper, Michał Moskal, Samuel Jenkins, Julian Berman, Nathan Ranchin, Robert West, Eric Horvitz, and Harsha Nori. Jsonschemabench: Evaluating constrained decoding with llms on efficiency, coverage and quality.

Maitrey Gramopadhye and Daniel Szafir. 2023. Generating executable action plans with environmentally-aware language models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3568–3575. IEEE.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36:79081–79094.

Yingqing He, Menghan Xia, Haoxin Chen, Xiaodong Cun, Yuan Gong, Jinbo Xing, Yong Zhang, Xintao Wang, Chao Weng, Ying Shan, and 1 others. 2023. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint arXiv:2307.06940*.

Rong Hu, Ye Yang, Sen Liu, Zuchen Li, Jingyi Liu, Xingchen Ding, Hanchi Sun, and Lingli Ren. 2025. Large language model driven transferable key information extraction mechanism for nonstandardized tables. *Scientific Reports*, 15(1):29802.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

James Oswald, Kavitha Srinivas, Harsha Kokel, Junkyu Lee, Michael Katz, and Shirin Sohrabi. 2024. Large language models as planning domain generators. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 423–431.

Vishal Pallagani, Bharath Chandra Muppasani, Kaushik Roy, Francesco Fabiano, Andrea Loreggia, Keerthiram Murugesan, Biplav Srivastava, Francesca Rossi, Lior Horesh, and Amit Sheth. 2024. On the prospects of incorporating large language models (llms) in automated planning and scheduling (aps). In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 432–444.

Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. 2022. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005.

Bailin Wang, Zi Wang, Xuezhi Wang, Yuan Cao, Rif A Saurous, and Yoon Kim. 2023a. Grammar prompting for domain-specific language generation with large language models. *Advances in Neural Information Processing Systems*, 36:65030–65055.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, and 1 others. 2023b. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Shawn Ma, and Yitao Liang. 2023c. Describe, explain, plan and select: interactive planning with llms enables open-world multi-task agents. *Advances in Neural Information Processing Systems*, 36:34153–34189.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. 2023. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*.

Dingyi Yang, Chunru Zhan, Ziheng Wang, Biao Wang, Tiezheng Ge, Bo Zheng, and Qin Jin. 2024. Synchronized video storytelling: Generating video narrations with structured storyline. *arXiv preprint arXiv:2405.14040*.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, and 1 others. 2024. Sglang: Efficient execution of structured language model programs. *Advances in neural information processing systems*, 37:62557–62583.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023a. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. Controlled text generation with natural language instructions. In *International Conference on Machine Learning*, pages 42602–42613. PMLR.

Yupeng Zhou, Daquan Zhou, Ming-Ming Cheng, Jiashi Feng, and Qibin Hou. 2024. Storydiffusion: Consistent self-attention for long-range image and video generation. *Advances in Neural Information Processing Systems*, 37:110315–110340.