Agentic Generative Systems: A Survey on Autonomous Multimodal Content Creation

Xinli Xu [†] , Litao Guo [†] , Yehang Zhang , Haotian Bai , Jiantao Lin , Jiawei Feng , Luozhou Wang , Zhen Yang , Man Chen , Zixin Zhang , Harold Haodong Chen , Chen Min , Ying-Cong Chen [*]

Posted Date: 28 July 2026

doi: 10.20944/preprints202607.1997.v1

Keywords: agentic generative systems; multimodal generation; generative agents; tool-orchestrated generation; survey

 doi:10.20944/preprints202607.1997.v1

Article

Agentic Generative Systems: A Survey on Autonomous Multimodal Content Creation

Xinli Xu [1,†], Litao Guo [1,†], Yehang Zhang [1], Haotian Bai [1], Jiantao Lin [1], Jiawei Feng [1], Luozhou Wang [1], Zhen Yang [1], Man Chen [1], Zixin Zhang [1], Harold Haodong Chen [1,2], Chen Min [3] and Ying-Cong Chen [1,2,*]

[1] The Hong Kong University of Science and Technology (Guangzhou)
[2] The Hong Kong University of Science and Technology
[3] Institute of Computing Technology, Chinese Academy of Sciences
[*] Correspondence: yingcongchen@hkust-gz.edu.cn
[†] Equal contribution.

Abstract

Recent advances in multimodal generative models have enabled high-quality synthesis across images, videos, audio, 3D content, and document-centric media. Yet most systems still treat generation as a direct prompt-to-artifact mapping or a fixed pipeline, limiting their ability to handle long-horizon objectives, compositional constraints, tool coordination, and iterative revision. At the same time, agent-based systems show that complex tasks can be addressed through explicit goals, planning, tool use, memory, and feedback-conditioned decisions. This convergence suggests that the central challenge is no longer only how to improve individual generators, but also how to control the generation trajectory that connects intent, intermediate artifacts, and final outputs. In this survey, we formalize this perspective as *Agentic Generative Systems (AGS)*: generation-centered systems that represent content creation goals, decompose them into intermediate steps or structured representations, and orchestrate generative models, tools, or executable operations along a multi-step trajectory. Depending on capability level, an AGS may execute an open-loop plan, revise outputs through local feedback, or strategically replan using persistent state and memory. We introduce a three-level capability hierarchy and a mechanism-centered taxonomy covering agent architectures, planning and control, tool orchestration, memory, and feedback. Under this framework, we review representative systems across image, video, audio, 3D, and document-centric generation, showing that the same agentic progression recurs across modalities while the object of control changes from spatial composition to temporal coherence, event timing, world state, or information structure. We further consolidate datasets, benchmarks, and evaluation protocols for AGS, emphasizing both final artifact quality and trajectory-level behavior. Finally, we discuss open challenges in long-horizon consistency, process-level evaluation, reliability, safety, cost-aware control, and generalization. This survey provides a unified conceptual foundation for studying autonomous multimodal content creation as a system-level decision process. We maintain a curated repository at https://github.com/xxlbigbrother/Awesome-Agentic-Generative-Systems.

Keywords: agentic generative systems; multimodal generation; generative agents; tool-orchestrated generation; survey

1. Introduction

Recent years have witnessed rapid progress in multimodal generative models, enabling impressive advances in image [1–4], video [5–7], audio [8,9], and 3D content creation [10,11]. By leveraging large-scale pre-training and powerful generative architectures, these models can synthesize visually and semantically rich content from diverse inputs such as text, images, and videos. Despite these achievements, most existing multimodal generation systems still follow a *single-step* or *open-loop*

 doi:10.20944/preprints202607.1997.v1

paradigm, where a user provides a prompt and the system produces an output in one pass. As a result, generation is predominantly treated as a static prediction problem rather than a dynamic decision-making process.

While effective for simple or unconstrained tasks, single-step generation struggles to support realistic content creation scenarios that involve complex goals, long-horizon dependencies, and fine-grained control. In practice, creative processes rarely terminate after one generation attempt. Instead, humans typically decompose high-level intentions into subtasks, leverage heterogeneous tools with distinct capabilities, inspect intermediate artifacts, and iteratively refine their outputs based on feedback. The absence of such process-level reasoning, tool coordination, and iterative refinement fundamentally limits the controllability, robustness, and scalability of conventional generative pipelines.

To make this concrete, consider an increasingly common document-to-media authoring task: turning a twelve-page scientific paper into a three-minute conference presentation video, complete with slides, a narrated voice-over, and animated figure explanations. Although such requests are becoming routine in scientific communication and education, they are not simple prompt-to-output problems. No single text-to-video model, however large, can solve this in one forward pass. The system must read the paper, decide which contributions to highlight, segment the narrative into slides, extract or regenerate figures, write a script, render the slides, synthesize narration, align audio with video, and verify that the result actually conveys the original ideas. Each of these subtasks already has a specialized tool—document understanders, slide generators, text-to-speech engines, video assemblers, alignment checkers—but turning them into a coherent presentation requires deciding which tools to invoke, in what order, and how to react when an intermediate result does not match the plan. Recent systems such as Preacher [12] and Paper2Video [13] attack exactly this task as a multi-step planning problem rather than a single generative call, and the same structural shape recurs in long-form video storytelling [14,15], 3D world building [16,17], and creative image editing [18,19]. The task is concrete, but the structural lesson is general: realistic generation requires planning, tool coordination, intermediate verification, and iterative revision.

Motivated by these limitations, recent research has begun to explore a shift toward an *agentic* generation paradigm [14,16,18–24]. As illustrated in Figure 1, agentic generative systems move beyond one-shot prediction and instead formulate generation as a multi-step decision process, ranging from planned tool orchestration to feedback-conditioned refinement. Given a high-level user goal, an agent performs multimodal understanding, decomposes objectives into subtasks, and selects or invokes appropriate generative tools; higher-level systems additionally evaluate intermediate outputs and revise their actions based on feedback. This paradigm enables systems to decide, to varying degrees, *what* to generate, *how* to generate it, and *when* to refine the results. More fundamentally, it reframes generation from model-centric inference to system-level orchestration: the system is no longer judged only by how well it maps prompts to outputs, but by how well it controls the entire generation trajectory it produces.

Importantly, the agentic paradigm is not restricted to a single modality or task. Figure 2 summarizes this cross-modal regularity across image synthesis and editing [18–21], video generation [14,15,25], long-form audio synthesis [23,26,27], 3D scene creation [16,17,28], and document-centric authoring [12,13,29]. Across domains, AGS operate on different controlled representations: spatial–semantic image states for images, temporally extended production states for videos, acoustic event timelines for audio, information-design states for documents, and editable world states for 3D scenes. Yet the same capability progression recurs across these representations. L1 systems externalize intent into plans, layouts, scripts, or workflows; L2 systems condition subsequent actions on intermediate outputs or diagnostic feedback; and L3 systems maintain state, coordinate specialized roles or tools, and replan over longer horizons. Thus, the common object of study is not a particular modality, but the controlled generation trajectory over domain-specific intermediate representations. Here, the L1–L3 terminology is used only as a preview of the capability progression; Sec. 3 later formalizes these levels, and Sec. 4 expands them into mechanism-level design patterns.

 doi:10.20944/preprints202607.1997.v1

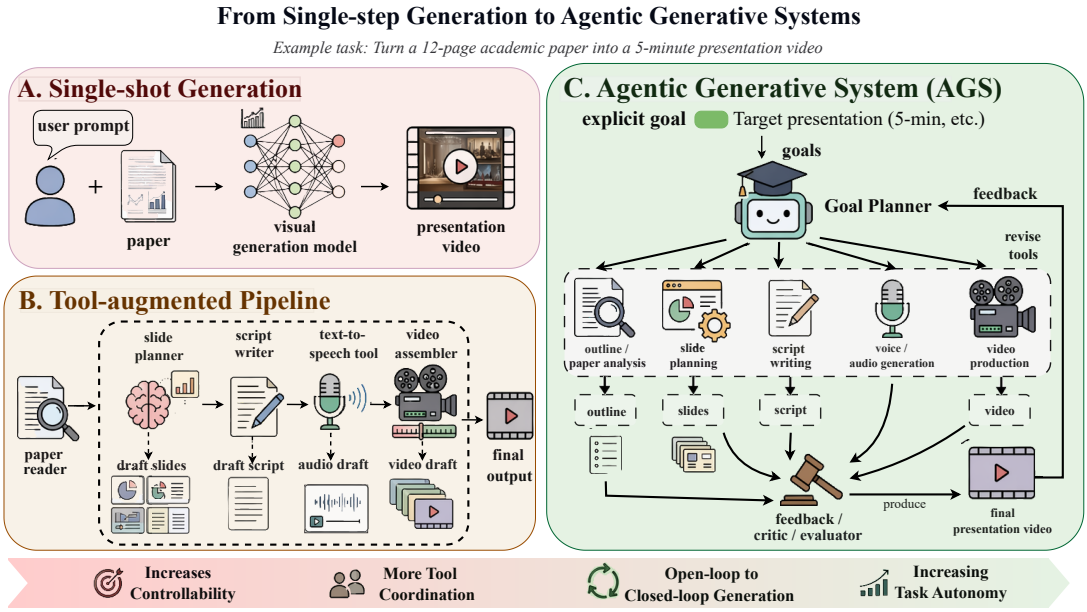

Figure 1. Comparison of three paradigms for multimodal content creation. Single-shot generation maps a prompt directly to an artifact. Tool-augmented pipelines decompose generation into modular steps but typically execute them open-loop. Agentic Generative Systems (AGS) externalize generation into a goal-conditioned decision process: an agent maintains an explicit goal, plans a trajectory over generative tools, and, in stronger closed-loop variants, observes intermediate artifacts and revises its actions until the goal is met.

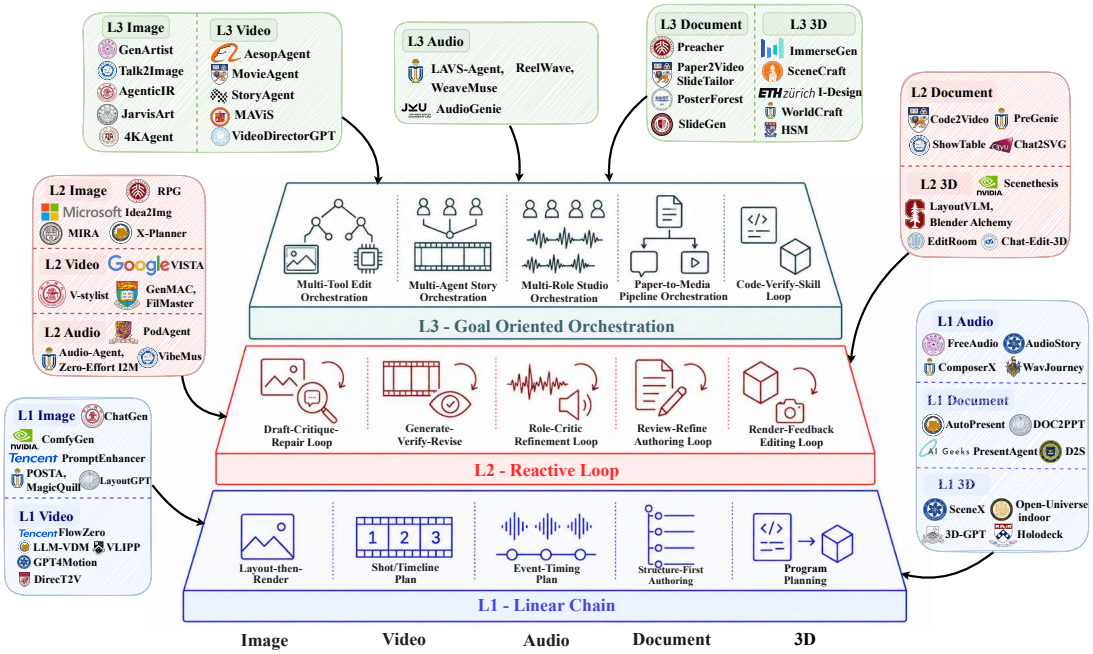

Figure 2. The L1–L3 capability progression of AGS reappears across image, video, audio, 3D, and document-centric generation. What changes across modalities is the *object of control*—spatial composition, temporal coherence, event timing, world state, or information structure—not the underlying agentic principle.

Despite rapid progress, the emerging literature on agentic generative systems remains conceptually fragmented at the mechanism level. Existing works differ substantially in their definitions of agents, system architectures, planning and control strategies, tool interfaces, memory mechanisms, feedback loops, and evaluation protocols. Moreover, agentic generation methods are often developed independently within specific modalities, making it difficult to identify shared design principles or conduct fair comparisons. As a result, the field currently lacks a unified understanding of what

 doi:10.20944/preprints202607.1997.v1

constitutes an agentic generative system, how different design choices relate to one another, and what capability boundaries separate simple orchestration from truly autonomous generation.

Existing surveys cover important parts of this landscape but leave the intersection of multimodal generation and agentic control under-specified. Model-centric generation surveys [30–34] rarely treat the generation process as a controllable trajectory, while surveys on LLM-based agents and agentic multimodal models [35–41] usually organize around general task completion rather than artifact production and revision. The closest survey, Lei et al. [42], studies multi-agent systems for audio-visual generation and understanding, but is organized primarily by modality and coordination patterns. Our survey instead isolates agentic generation as a generation-centered paradigm and studies it through a capability-graded, mechanism-centered framework that applies across image, video, audio, 3D, and document-centric authoring. In formalizing this paradigm, we also distinguish AGS from two adjacent concepts that are sometimes conflated with it: general-purpose LLM agents, which are organized around task-solving rather than artifact production; and native agentic generators [43,44], in which planning and reflection are partly internalized into a single model, changing the substrate of AGS but not eliminating the need for system-level orchestration across heterogeneous tools and modalities. To the best of our knowledge, this is the first survey to formalize agentic generation as a distinct paradigm under a mechanism-centered, capability-graded framework that applies consistently across image, video, audio, 3D, and document-centric content creation.

In this survey, we introduce *Agentic Generative Systems (AGS)* as a first-class paradigm for autonomous multimodal content creation. Our goal is to provide a representative, mechanism-centered survey rather than an exhaustive catalog of all generative models. We primarily cover systems published between 2023 and 2026 that apply agentic control to image, video, audio, 3D, or document-centric content creation. We include a system when the generation trajectory itself becomes an object of control: the system explicitly decomposes goals, constructs intermediate representations, orchestrates tools or executable operations, uses feedback to revise artifacts, maintains state or memory, or coordinates multiple specialized roles during artifact production. Conversely, we do not treat ordinary single-shot generators, isolated prompt-engineering methods, or fixed pipelines without explicit system-level control as core AGS instances, even when they use strong generative models. This scope motivates our organization of the literature by capability level and mechanism rather than by modality alone. Within this scope, we review a broad and representative set of systems and organize them under a unified conceptual framework. Concretely, this survey makes four contributions.

- We *formalize* Agentic Generative Systems by providing a definition that identifies core components (goal representation, planning, tool-oriented generation, feedback, and memory) and key properties (goal-driven, multi-step, tool-centric, modality-agnostic, and progressively closed-loop/stateful), while clarifying that feedback and persistent memory characterize stronger systems rather than the minimum AGS threshold. We also introduce a three-level capability hierarchy: **L1** (linear-chain), **L2** (reactive-loop), and **L3** (goal-oriented orchestrated) (Sec. 3).
- We develop a *mechanism-centered taxonomy* that characterizes AGS along architecture, planning and control, tool orchestration, memory, and feedback dimensions, and argue that these dimensions—not modality—are the right axes for comparing agentic generative systems (Sec. 4).
- We *review representative AGS* across image, video, audio, 3D, and document-centric generation under a unified L1–L3 lens, and show that the same capability progression reappears across modalities with different objects of control (Sec. 5).
- We consolidate benchmarks and evaluation protocols for AGS, emphasizing artifact quality, trajectory quality, interaction utility, long-horizon robustness, and efficiency; we further synthesize open challenges including long-horizon consistency, cost-aware control, process-level evaluation, and the relationship between external orchestration and native agentic models (Sec. 6 and Sec. 7).

Figure 3 summarizes the organization of this survey.

 doi:10.20944/preprints202607.1997.v1

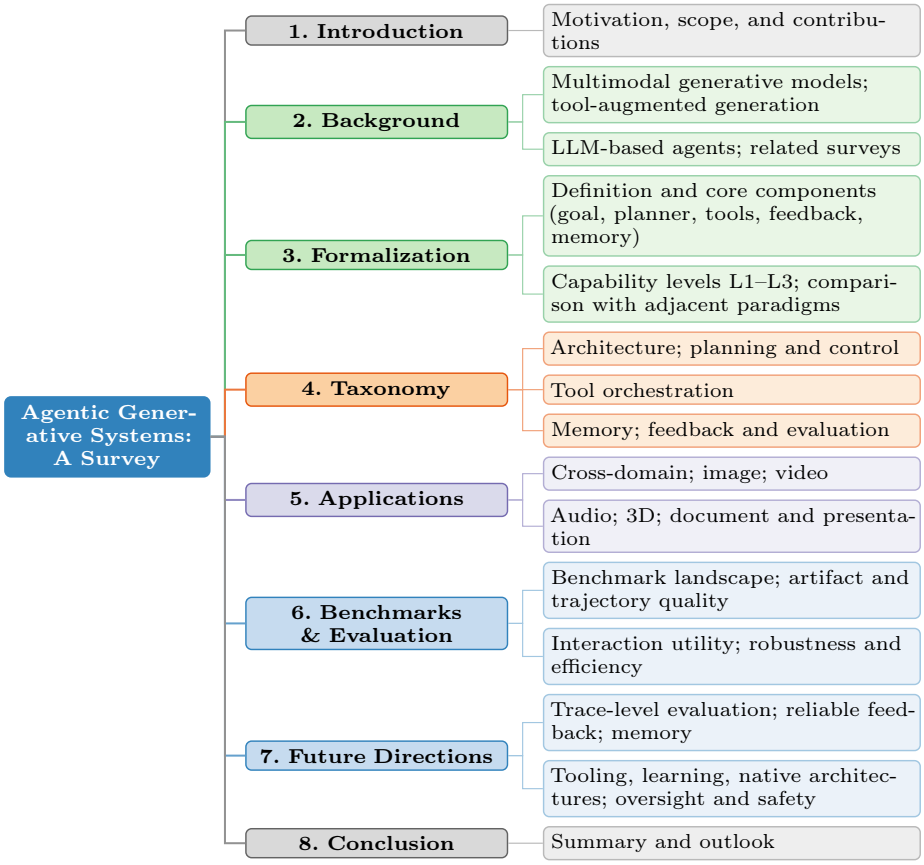

Figure 3. Overview of the paper structure, color-grouped into foundations (Sec. II–III), mechanisms (Sec. IV), applications (Sec. V), and evaluation and outlook (Sec. VI–VIII).

2. Background

Recent advances in foundation models have substantially improved multimodal content generation, enabling high-quality synthesis of images, videos, audio, 3D assets, and mixed-media artifacts from natural language and other conditioning signals [1,5,8,10]. At the same time, large language models (LLMs) and multimodal large language models (MLLMs) have introduced new forms of reasoning, planning, tool use, memory, and feedback-conditioned interaction that extend beyond one-shot prediction [35,38]. These trends are increasingly converging: generative systems are no longer only model-centric predictors, but are gradually becoming structured systems that can externalize goals, invoke heterogeneous tools, and, in stronger variants, revise outputs using feedback and persistent state.

To situate Agentic Generative Systems (AGS) within this broader landscape, this section reviews four closely related bodies of work: multimodal generative models, tool-augmented and compositional generation, agentic systems and LLM-based agents, and existing survey literature. We then summarize how these developments jointly motivate the transition from conventional prompt-to-artifact pipelines to agentic generation.

2.1. Multimodal Generative Models

Multimodal generative models form the algorithmic foundation of modern content creation systems. Their central objective is to synthesize target content in one or more modalities conditioned on textual, visual, audio, or structured inputs. Over the past few years, this area has witnessed rapid progress across images, videos, audio, 3D, and document-centric or mixed-media authoring, driven by large-scale pre-training, improved architectures, and richer conditioning mechanisms.

Text-to-image generation has become one of the most mature branches of multimodal generation. Early progress was driven by autoregressive [45] and GAN-based approaches [46,47], while diffusion

 doi:10.20944/preprints202607.1997.v1

models later emerged as the dominant paradigm due to their strong fidelity, diversity, and controllability [1–4,48]. Modern image generators can synthesize high-quality content from prompts, image references, spatial layouts, masks, and style constraints, with controllability further improved by conditioning mechanisms such as ControlNet [49] and IP-Adapter [50]. Beyond generation from scratch, image editing methods support operations such as inpainting, style transfer, subject replacement, compositional modification, and localized refinement [51–53]. These advances have greatly improved visual quality and controllability, but many systems still treat generation as a single forward process, with limited support for explicit multi-step planning, tool selection, or structured self-correction.

Video generation extends image synthesis into the temporal domain and introduces additional challenges such as motion consistency, temporal coherence, camera dynamics, scene continuity, and long-range narrative structure. Recent text-to-video and image-to-video models have demonstrated impressive capabilities in generating short clips with plausible motion and semantics [5–7,54–56]. However, compared with image generation, video creation is more sensitive to planning deficiencies: producing a coherent video often requires decomposition into scenes, shots, actions, and temporal constraints, which are difficult to capture in a single prompt-to-video step. This makes video generation a natural domain in which explicit planning and iterative refinement become especially valuable.

Audio generation includes speech synthesis, sound effect generation, music composition, long-form audio production, and multimodal audio generation aligned with visual or textual inputs. Recent systems have shown strong capabilities in generating realistic speech [57], expressive music [9,58], and semantically aligned soundscapes [8,59]. Yet audio is inherently temporal and layered: useful generation often depends on event timing, emotional progression, source coordination, and synchronization with other media. As a result, practical audio generation often benefits from explicit decomposition into narrative units, event structures, or production stages.

3D generation moves beyond perceptual synthesis toward structured world construction. Tasks in this area include 3D asset generation, scene layout, object placement, procedural world building, texture generation, and interactive editing. Recent methods based on score distillation [10,60], feed-forward prediction [11,61], and 3D Gaussian representations [62,63] have substantially improved the quality and efficiency of 3D content creation. Compared with 2D generation, 3D systems often require stronger structural priors, geometry-aware reasoning, and compatibility with downstream tools such as rendering engines, simulation environments, and editing software. This makes 3D generation particularly dependent on intermediate representations such as layouts, programs, code, or scene graphs.

Document-centric and presentation-oriented generation further broadens the scope of multimodal content creation. Here the target is not merely a perceptual artifact, but a communicative object such as a poster, slide deck, narrated presentation, or paper-derived video. Such artifacts require source fidelity, rhetorical organization, layout hierarchy, editable visual assets, narration, and often audiovisual synchronization. Recent systems for scientific video, slide-deck, and poster generation therefore expose an important limitation of model-centric generation: high-quality authoring requires coordinating understanding, planning, rendering, and revision across multiple representational levels [12,13,29,64].

Despite remarkable progress, most multimodal generative models are still fundamentally designed as *single-shot predictors*: a user provides an input condition, and the model maps it directly to an output artifact. This paradigm is effective when tasks are simple, underconstrained, or primarily perceptual. However, realistic content creation often involves more than producing a plausible artifact in one pass. Common failure modes include object-count errors in multi-instance image generation [65], temporal drift and identity loss in long-form video [14], inconsistent layering in multi-source audio production [26], geometric implausibility in 3D scene construction [16], and information loss in document-to-media authoring [13,29]. These failures are not primarily failures of local generation quality; they are failures of *process control*. The system lacks a reliable mechanism for decomposing high-level goals, maintaining consistency across stages, coordinating different tools, checking interme-

 doi:10.20944/preprints202607.1997.v1

diate outputs, and deciding when a plan should be revised. This gap motivates the emergence of more structured and agentic forms of generative systems.

2.2. Tool-Augmented and Compositional Generation

An important intermediate step between conventional generation and AGS is the rise of tool-augmented and compositional generation. Instead of relying on a single end-to-end model to solve the entire task, these systems combine multiple specialized components, each responsible for part of the generation process.

Many practical generation tasks naturally decompose into subtasks such as prompt enhancement, layout prediction, asset retrieval, image synthesis, video assembly, audio mixing, rendering, and post-editing. Tool-augmented systems exploit this structure by chaining multiple modules together. Early demonstrations such as Visual ChatGPT [66], HuggingGPT [67], and InternGPT [68] showed that LLMs can serve as coordinators over visual foundation models, enabling multi-step generation and editing through sequential tool selection. In 3D or document-generation settings, systems may further include code generators, retrieval modules, rendering engines, layout optimizers, or software APIs. Such modularization improves controllability and enables systems to leverage the complementary strengths of heterogeneous models.

A related trend is compositional generation, in which generation is explicitly organized around structured intermediate representations. These representations may include scene graphs, temporal plans, shot lists, workflow graphs, scripts, design layouts, motion trajectories, or editable programmatic descriptions. For example, LayoutGPT [69] and LLM-grounded Diffusion [70] convert language into explicit spatial layouts before image synthesis, while GLIGEN [71] conditions generation on grounding inputs such as bounding boxes and keypoints. Compared with raw prompts, such intermediate forms expose latent structure that downstream generative models can follow more reliably. They also make the generation process more interpretable and, in many cases, more editable.

As the number of available generation tools grows, an important challenge becomes deciding *which* tools to invoke, *in what order*, and *with what parameters*. Early solutions often rely on manually designed pipelines or prompt-adaptive workflows. More advanced systems such as ComfyGen [72] and ComfyGPT [73] generate workflow graphs, code, or executable plans dynamically within platforms like ComfyUI. This shift is significant: orchestration itself becomes a problem of reasoning and control rather than fixed software engineering. In other words, the system must not only generate content, but also generate the *process* by which content is produced.

However, tool augmentation alone does not automatically imply AGS. Many modular systems still follow a fixed developer-designed path: they may chain models or rewrite prompts, but they do not externalize a generation goal into an agent-controlled trajectory. For this survey, we place the boundary at systems that explicitly plan over tools, models, or structured representations to realize a content creation goal. Such systems may still be open-loop at the L1 level, but they differ from ordinary pipelines because orchestration is an explicit object of control rather than a hidden implementation detail. Feedback integration, persistent memory, and strategic replanning then mark stronger AGS rather than the minimum threshold.

2.3. Agentic Systems and LLM-Based Agents

In parallel with advances in generative models, another major research trajectory has focused on agentic systems built around LLMs and MLLMs. These systems treat foundation models not merely as predictors, but as decision-making modules capable of reasoning, planning, tool use, and interaction [35,36].

LLM-based agents gained prominence with the observation that large language models can act as general-purpose controllers over external tools, APIs, environments, and memory [74,75]. Given a user instruction, such systems can decompose tasks into subgoals, call tools, inspect outputs, and decide subsequent actions. This paradigm has been explored in areas such as embodied control [76], software engineering [77], web interaction, office automation, scientific assistance, and multimodal

 doi:10.20944/preprints202607.1997.v1

task solving. The key conceptual shift is that the model is no longer used only for direct prediction; instead, it can participate in a control process over actions and observations.

A core promise of agentic systems lies in their ability to externalize deliberation. Instead of compressing all reasoning into a single forward inference, these systems often generate explicit intermediate plans, maintain task state, and adapt their behavior based on observed outcomes. They may employ chain-of-thought reasoning [78], hierarchical decomposition, planner–executor structures, critique-and-revise loops, tree-of-thought search [79], or search over candidate trajectories. Reflection and self-correction further allow agents to detect failures, explain what went wrong, and propose improved actions [80,81]. These capabilities are especially relevant for generation tasks in which intermediate artifacts can be inspected, revised, or reassembled.

Agentic systems frequently operate through tool invocation [75,82]. In this setting, actions may correspond to calling a model, executing code, querying a retriever, manipulating an editable canvas, controlling software, or interacting with a simulator. Such tool use provides a bridge between abstract reasoning and concrete execution. For generative tasks, this means an agent can coordinate image generators, video editors, audio synthesizers, layout engines, 3D software, and evaluation modules within a unified decision process.

Another important property of agentic systems is statefulness. Unlike purely stateless prompt-response systems, many agents maintain short-term working memory, interaction history, world state, or reusable experience [83,84]. This enables them to preserve consistency across turns, accumulate constraints over time, and support long-horizon interaction. In generative settings, memory can maintain character identity in storytelling, track previously executed edits, store assets and scene structure, or preserve user preferences in co-creative workflows.

Recent native agentic generators point to a complementary direction in which planning, reflection, or evaluation is partly internalized within a generative model itself [43,44]. Such models blur the boundary between monolithic generators and externally orchestrated systems, but they do not remove the need to analyze goals, intermediate artifacts, feedback, and control over generation trajectories. In our framework, native agentic generators can instantiate parts of the planner or feedback mechanism, while system-level AGS remain necessary when content creation requires heterogeneous tools, persistent artifact state, or cross-modal production workflows.

Although AGS build upon the broader literature on LLM-based agents, the two concepts are not identical. General-purpose agents are organized around task completion, whereas AGS are organized around the production and revision of media artifacts. The central object in AGS is therefore not only a task state, but also an evolving artifact and its generation trajectory. This artifact-centric emphasis introduces concerns that are secondary in many general agents: fidelity to source material, aesthetics, identity consistency, temporal coherence, editability, perceptual feedback, and multimodal alignment. Thus, AGS inherit reasoning, tool use, and memory from agentic systems, but specialize them for creating, editing, and refining multimodal content.

2.4. Related Surveys

Several existing surveys are relevant to AGS, but they typically address only one side of the emerging landscape. We review the most pertinent works below and clarify how the present survey complements them.

Extensive surveys have reviewed specific generative modalities, including diffusion models [30], text-to-image generation [31], video diffusion, controllable video generation, and human-motion video generation [32,85,86], 3D generation [33,87,88], vision-to-music generation [89], and broader AIGC landscapes [34]. These works provide valuable coverage of architectures, training strategies, controllability mechanisms, and evaluation metrics. However, their primary focus is the generative *model* itself: generation is analyzed as a direct mapping problem, and system-level concerns such as multi-step planning, tool orchestration, and feedback-driven refinement receive limited attention.

Another active line of work reviews LLM-based agents and autonomous systems. Xi et al. [35] organize LLM agents around a brain–perception–action framework; Wang et al. [36] propose a profile–

 doi:10.20944/preprints202607.1997.v1

memory–planning–action taxonomy; Guo et al. [37] survey multi-agent LLM collaboration; and Lin et al. [90] focus specifically on creativity in LLM-based multi-agent systems. These surveys comprehensively cover reasoning, planning, memory, tool use, environment interaction, and creative collaboration. While highly relevant to AGS, they are usually not centered on multimodal content creation as a controllable production process: generation may appear as one tool among many, rather than as the central task around which the system is organized.

More recent surveys bridge the gap between agents and multimodal systems [91]. Xie et al. [40] provide an early taxonomy of large multimodal agents organized by memory and training type (Type I–IV). Durante et al. [41] survey "Agent AI" across embodied, multimodal, and generative settings. Yao et al. [38] focus on agentic multimodal LLMs as *models*, examining how reasoning, reflection, and memory capabilities can be trained via reinforcement learning; their applications span DeepResearch, healthcare, GUI agents, and autonomous driving, but not content-creation orchestration. Schneider [39] articulates the conceptual pivot from generative to agentic AI at a domain-agnostic level, proposing a five-level autonomy hierarchy. These works move closer to the territory of AGS, but they remain broad in scope—spanning understanding, reasoning, embodiment, and interaction—without offering a unified treatment of autonomous content creation as a distinct paradigm.

Lei et al. [42] recently surveyed multi-agent systems for audio-visual generation *and* understanding, organizing generation systems by dominant modality (visual, audio, cross-modal) and identifying four coordination patterns: pipeline, hierarchical, reflection-and-refinement, and dynamic collaboration. This work is the closest in scope to the present survey. However, our survey differs in four key respects. First, we focus exclusively on *generation*, enabling deeper engagement with generation-specific concerns such as long-horizon consistency, editability of intermediate artifacts, and non-scalar perceptual feedback. Second, we propose a unified *capability hierarchy* (L1–L3) that captures an ordered progression from linear-chain execution to goal-oriented orchestration, rather than treating coordination patterns as parallel design alternatives. Third, we organize the literature by *mechanism* (architecture, planning, tool orchestration, memory, feedback) rather than by modality, revealing design principles that recur across domains. Fourth, we include *document-centric authoring* (slides, posters, scientific videos) as a first-class modality, which is scattered or absent in prior work. For a detailed taxonomy of multi-agent communication protocols and internal reasoning strategies, we refer readers to [42]; in this survey, we focus on how these mechanisms compose into system-level control flows specific to generation.

2.5. From Generative Pipelines to Agentic Generative Systems

The developments reviewed above jointly point to a broader transition in the design of generative systems. Traditional multimodal generation is model-centric: a model receives a condition and directly predicts an output. Tool-augmented generation introduces modularity and decomposition, but often remains pipeline-driven. LLM-based agents add explicit reasoning, planning, tool use, memory, and feedback loops, yet are usually not specialized for content creation. As illustrated in Figure 4, AGS emerge at the intersection of these three lines.

Conceptually, AGS can be viewed as systems that elevate generation from a single prediction problem to a multi-step decision-making process. In this process, the system must determine what to generate, how to decompose the goal, which tools or models to invoke, how to represent intermediate artifacts, and when to terminate. In stronger systems, it must also interpret feedback, update memory, and revise the plan. The output is no longer the only object of intelligence; the *generation trajectory* itself becomes a first-class object of design and evaluation.

This transition is particularly important because realistic content creation is rarely achieved in one pass. Human creators typically sketch, critique, revise, combine tools, adjust local details, and preserve global goals over time. AGS aim to bring analogous process-level intelligence into generative systems. Rather than replacing powerful multimodal generators, they organize such generators within a more explicit and controllable system structure.

 doi:10.20944/preprints202607.1997.v1

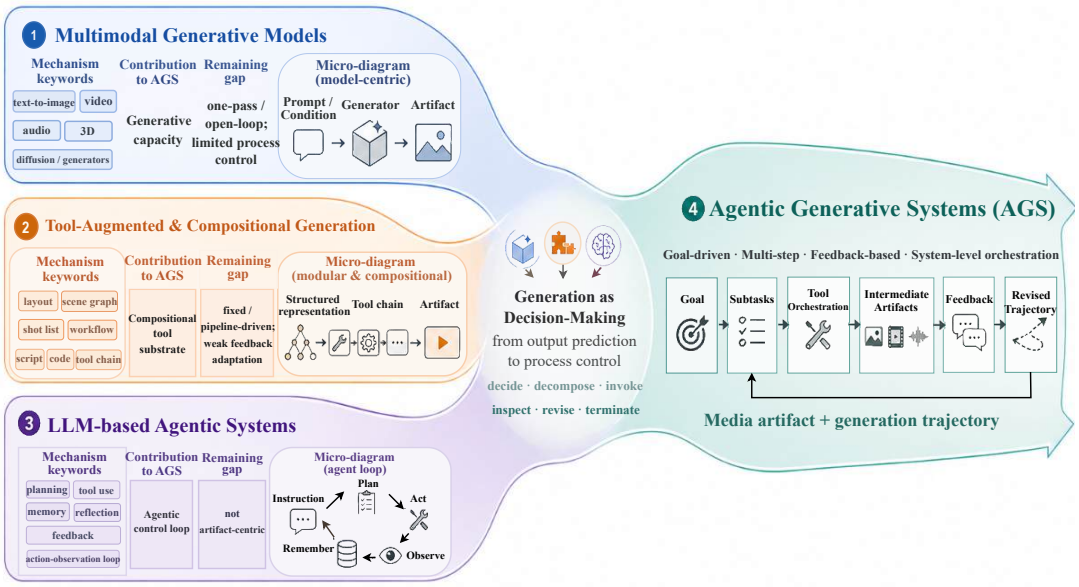

Figure 4. From multimodal generative models to Agentic Generative Systems (AGS). Three research threads—multimodal generation (Sec. 2.1), tool-augmented and compositional generation (Sec. 2.2), and LLM-based agentic systems (Sec. 2.3)—are converging into a unified paradigm that shifts content creation from direct model-centric prediction toward goal-driven, tool-oriented system-level orchestration, with feedback and memory enabling stronger forms of agentic control.

Motivated by this perspective, the next section formalizes Agentic Generative Systems, identifies their core components and key properties, and introduces a capability-oriented hierarchy that serves as the conceptual foundation for the taxonomy developed in the remainder of this survey.

3. Agentic Generative Systems: Formalization and Principles

Building on the background reviewed in Sec. 2, we now formalize the concept of Agentic Generative Systems (AGS). We first give a formal definition, then identify the recurring functional components used by existing systems, define a three-level capability hierarchy (L1–L3), and clarify the relationship between AGS and adjacent paradigms.

3.1. Formal Definition of Agentic Generative Systems

An *Agentic Generative System (AGS)* is a generative system that produces content through an explicit process of goal-driven planning and tool-oriented execution, with feedback-based refinement and persistent state emerging in higher-capability systems. Unlike conventional generative models that operate in a single forward pass, AGS perform generation as a structured, multi-step decision-making process.

Definition 1 (Agentic Generative System). *An Agentic Generative System is a generation-centered system that explicitly represents a content creation goal, decomposes it into intermediate steps or structured representations, and uses an agent or controller to orchestrate generative tools, models, or executable operations along a multi-step generation trajectory. Depending on its capability level, the trajectory may be executed open-loop, locally revised through feedback, or strategically replanned using persistent state and memory.*

Formally, an AGS can be described as a tuple $\mathcal{S} = (\mathcal{G}, \mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{M})$, where $\mathcal{G}$ denotes the goal representation, $\mathcal{P}$ the planner or controller, $\mathcal{T} = \{t_1, t_2, \ldots, t_K\}$ a set of generative tools or executable operations, $\mathcal{F}$ an optional feedback mechanism, and $\mathcal{M}$ an optional memory mechanism. The optionality of $\mathcal{F}$ and $\mathcal{M}$ is important: L1 systems may instantiate them as empty, post-hoc, or weakly specified components, whereas L2/L3 systems make them active parts of control. Under this formulation, an

 doi:10.20944/preprints202607.1997.v1

AGS produces a *generation trajectory* $\tau = (s_0, a_0, o_0, s_1, a_1, o_1, \ldots, s_T)$, where s_t is the system state at step t (encompassing the current artifact, active goals, available intermediate representations, and memory contents when present), a_t is an action over $\mathcal{T}$ (e.g., a tool invocation, model call, code execution, or editing operation with specific parameters), and o_t is the returned observation, artifact, diagnostic signal, or user response. The planner $\mathcal{P}$ selects actions conditioned on the current state and goal, $a_t = \mathcal{P}(s_t; \mathcal{G})$. In L1 systems, the sequence of actions may be fixed after the initial plan and observations need not alter the current trajectory; in L2/L3 systems, later actions are conditioned on intermediate observations, feedback, and memory. The state transitions according to $s_{t+1} = \delta(s_t, a_t, o_t)$, where δ incorporates the effect of tool outputs and, when applicable, feedback interpretation or memory updates. Generation terminates when a stopping criterion is satisfied, which may be defined by the goal (e.g., all subgoals completed), a pre-specified budget or workflow endpoint, an evaluator in $\mathcal{F}$, or an external signal such as user approval.

This definition emphasizes several essential distinctions from conventional generation. First, AGS are *goal-driven* rather than prompt-driven: the system maintains an explicit representation of the generation objective beyond surface-level instructions [16,24]. Second, generation is realized through *explicit planning and control*, rather than implicit reasoning hidden inside a single model inference [18,22]. Third, AGS treat generative models as *tools* within a larger decision loop, instead of as monolithic end-to-end solvers [67,92].

A system qualifies as an AGS when it externalizes generation into an explicit, goal-conditioned multi-step process and uses an agent or controller to plan over tools, models, executable operations, or structured intermediate representations. Feedback-conditioned action is not required for L1 systems; rather, it marks the transition from planned orchestration to closed-loop AGS. Conversely, systems that merely rewrite prompts, select a model variant, or execute a fixed developer-coded pipeline without explicit decomposition or tool/representation-level orchestration fall below this threshold and are better characterized as conventional or tool-augmented pipelines (cf. Sec. 2.2). Native agentic generators [43,44] occupy a related position: they may internalize planning or reflection inside a model, but they qualify as AGS only when such internal agency participates in an explicit generation trajectory with identifiable goals, intermediate states, and controllable revision behavior.

3.2. Core Components of AGS

Despite diverse implementations, existing AGS can be analyzed through a common set of five functional components. These components are instantiated to different degrees across capability levels; lower-level systems may include only weak or degenerate forms of feedback and memory. Figure 5 illustrates how they interact in the strongest closed-loop form of agentic generation, and the key properties that each component gives rise to are summarized alongside.

Goal Representation ($\mathcal{G}$). AGS maintain an explicit representation of the generation objective, which may include high-level intent, constraints, style requirements, or long-term consistency goals. This representation can be static (derived from the initial user instruction) or dynamic (updated during execution). Explicit goal modeling enables the system to reason about progress, failure, and completion, distinguishing AGS from prompt-conditioned systems where objectives are implicitly encoded and not explicitly tracked. In practice, goals may take forms as diverse as dialogue-state structures [24], scene-graph specifications [16], or narrative outlines [14,15].

Planning and Control ($\mathcal{P}$). Planning refers to the process by which the system decomposes a goal into a sequence or structure of subtasks. Control governs the selection, ordering, and termination of these subtasks during execution. Planning strategies range from simple linear decomposition [69,93] to tree- or graph-based planning with backtracking [18,22] and explicit search procedures such as Monte Carlo Tree Search (MCTS) [94]. Control policies determine whether the system merely follows a front-loaded plan or adapts subsequent actions based on intermediate outcomes and feedback. This component gives rise to the defining *multi-step* property of AGS: generation unfolds over multiple decision steps rather than a single inference pass.

 doi:10.20944/preprints202607.1997.v1

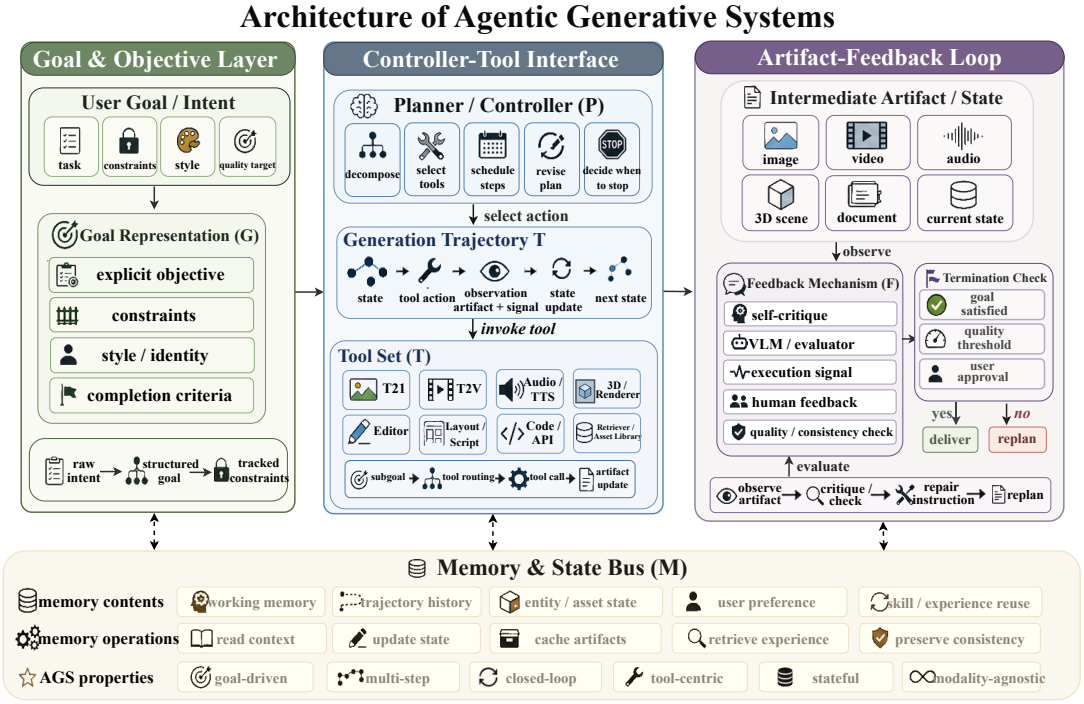

Figure 5. Schematic architecture of an Agentic Generative System. Given a user goal, the *Goal Representation* ($\mathcal{G}$) maintains explicit objectives. The *Planner* ($\mathcal{P}$) decomposes goals into subtasks and selects tools from the *Tool Set* ($\mathcal{T}$). Intermediate outputs may be evaluated by the *Feedback Mechanism* ($\mathcal{F}$), which informs plan revision in closed-loop systems. *Memory* ($\mathcal{M}$) can preserve state across steps, enabling long-horizon consistency and experience reuse. Arrows indicate information flow; L1 systems may execute a pre-generated trajectory open-loop, while L2/L3 systems close the loop through feedback and memory.

Tool Orchestration ($\mathcal{T}$). In AGS, content generation is performed by invoking generative tools or executable operations, such as text-to-image models, video generators, audio synthesizers, 3D asset creators, retrieval modules, layout optimizers, editing APIs, rendering engines, or native agentic generators. These tools are treated as callable modules with inspectable inputs and outputs. The system decides *when*, *how*, and *which* tools to invoke, rather than embedding all generation capability within a single model. Tool orchestration may operate through goal-conditioned linear pipelines [72], dynamic routing [67,92,95], workflow graph synthesis [22,73,96], or code- and software-grounded execution [16,97,98]. The intelligence of AGS lies in this orchestration rather than in raw generation capacity, making the system inherently *tool-centric*.

Feedback and Refinement ($\mathcal{F}$). Higher-level AGS incorporate mechanisms to evaluate intermediate or final outputs and adjust subsequent actions accordingly. Feedback may originate from internal self-critique [20,99], external evaluators or vision-language model (VLM) critics [65,100], execution and environment signals [16,73], or human input [101,102]. This enables iterative refinement and error correction, which are critical for long-form, multi-modal, and high-stakes generation tasks. The feedback component is what turns AGS from planned open-loop orchestration into *closed-loop* generation: the system can respond to errors, inconsistencies, or unmet constraints rather than executing blindly.

Memory ($\mathcal{M}$). Stronger AGS maintain persistent or semi-persistent state across generation steps to support long-horizon consistency. Memory ranges from short-term working memory that preserves recent context [101,103] to trajectory and interaction memory that tracks the evolving history of actions and revisions [19,24,104], and further to entity and asset memory that preserves character identity, scene structure, or reusable generation skills [16,17,105]. Memory allows AGS to be *stateful* across steps, accumulating constraints and experience rather than treating each action as independent.

 doi:10.20944/preprints202607.1997.v1

Together, these five components define the design space of AGS. As more components become active, AGS progress from goal-driven multi-step orchestration toward closed-loop, stateful, and strategically replanned generation. In addition, because the framework places no constraint on the output modality of $\mathcal{T}$, AGS are inherently *modality-agnostic*: the same agentic principles apply across image, video, audio, 3D, and document-centric generation.

3.3. Capability Levels of AGS: L1 to L3

While all AGS can be described using the above components, existing systems differ significantly in their degree of autonomy, adaptivity, and planning depth. To capture this variation, we use a three-level capability hierarchy. This section defines only the boundary conditions of the hierarchy; Sec. 4 expands it into a mechanism-level taxonomy with representative systems and design patterns. Table 1 summarizes the criteria.

L1: Linear-Chain AGS execute a goal-conditioned plan through a fixed, selected, or pre-generated sequence of tool invocations. The key threshold for L1 is explicit decomposition or orchestration beyond a single forward generation pass, even if the resulting trajectory is executed open-loop. Feedback, if present, is post-hoc and does not alter the current trajectory.

L2: Reactive-Loop AGS add feedback-conditioned iteration. At least one subsequent action depends on an observed intermediate artifact, diagnostic signal, evaluator judgment, or user response, enabling critique, repair, or local refinement. However, control remains primarily reactive: the system fixes observed failures within the current trajectory rather than revising a long-horizon strategy.

L3: Goal-Oriented Orchestrated AGS maintain richer state and support strategic replanning. They use feedback and memory not only to repair local errors, but also to revise the overall plan, compare alternative sub-strategies, coordinate specialized tools or agents, and preserve goals across longer generation horizons.

These levels are comparative rather than absolute thresholds. They describe increasing control over the generation trajectory: from one-way orchestration, to closed-loop local correction, and finally to stateful long-horizon control. The next section uses this hierarchy as a backbone, while analyzing the concrete mechanisms that realize each level.

Table 1. Capability-level criteria for Agentic Generative Systems. The table summarizes the boundary conditions used in this section; Sec. 4 expands them into mechanism-level design patterns and representative systems.

Dimension	L1: Linear Chain	L2: Reactive Loop	L3: Goal-Oriented Orchestrated
Control pattern	Executes an explicit generation plan once.	Corrects local failures using intermediate observations.	Tracks state and can revise the overall strategy.
Planning	Front-loaded decomposition into a sequence or workflow.	Reactive revision within the current trajectory.	Hierarchical, search-based, or globally revisable planning.
Feedback	Absent or post-hoc; does not alter the current trajectory.	Step-level critique, repair, or refinement.	Multi-granularity feedback can alter the plan.
Memory	Minimal or none.	Short-term working state for the current correction loop.	Persistent trajectory, entity, asset, or skill memory.
Goal handling	Static goal representation.	Static goal with local updates.	Explicit goal tracking over longer horizons.
Transition	Explicit goal-conditioned decomposition beyond single-shot generation.	At least one later action is conditioned on feedback.	Strategic replanning based on accumulated evidence.

3.4. Comparison with Related Paradigms

To further clarify the position of AGS, Table 2 contrasts them with four adjacent paradigms along the dimensions introduced above. The central distinction is that AGS treat *media artifacts and the generation trajectory* as first-class objects, whereas other paradigms center on either direct prediction (single-shot generation), developer-specified chaining (tool-augmented generation), or general task completion (LLM agents, robotic agents). This focus on artifact-centric orchestration gives rise to generation-specific concerns—editability of intermediate state, perceptual feedback that resists scalar reduction, and planning

 doi:10.20944/preprints202607.1997.v1

over structural representations such as layouts, scene graphs, and shot lists—that do not naturally arise in adjacent paradigms. Native agentic generators are compatible with this framework when they expose or participate in such artifact-centric control; otherwise, they are better viewed as powerful generative components whose internal reasoning remains outside the system-level trajectory analyzed here.

Table 2. AGS compared with adjacent paradigms. AGS are distinguished by treating media artifacts and the generation trajectory as first-class objects of control, feedback, and state.

Paradigm	Central Object	Control Logic	Feedback / State	Tools or Actions	Scope
Single-shot generation	Output artifact.	Direct prompt-to-artifact mapping.	None.	One implicit generator.	Usually one media modality.
Tool-augmented generation	Output artifact.	Developer-specified or shallow workflow.	Limited or post-hoc.	Chained models or APIs.	One or few modalities.
LLM agents	Task completion.	General task decomposition.	Task-level feedback; optional memory.	General-purpose APIs or tools.	Domain-agnostic tasks.
Robotic agents	Physical state.	Perception-action control.	Sensor feedback; maps or state estimators.	Embodied actuators.	Physical environments.
AGS (ours)	Media artifact plus generation trajectory.	Goal-conditioned generation control.	L2/L3 feedback; L3 persistent memory.	Generative tools, models, or operations.	Any media modality.

The formalization and capability levels introduced in this section provide the conceptual foundation for the mechanism-centered taxonomy developed in Sec. 4 and the cross-modal analysis in Sec. 5.

4. Taxonomy of Agentic Generative Systems

This section expands the formalization in Sec. 3 into a mechanism-centered taxonomy of Agentic Generative Systems (AGS). Rather than grouping methods only by modality or task, we characterize how they represent goals, plan generation trajectories, orchestrate tools or executable operations, maintain state, and use feedback to refine outputs over time. This perspective is necessary because two systems may both target image, video, or 3D generation while differing substantially in autonomy, controllability, robustness, and interaction style.

We structure the taxonomy around one primary capability axis and several mechanism axes. The primary axis is the *capability hierarchy* from L1 to L3 introduced in Sec. 3.3, which captures increasing control over the generation trajectory. The mechanism axes then explain how this control is implemented: architectural organization, planning and control strategy, tool orchestration mechanism, memory design, and feedback/evaluation loop. Together, these dimensions provide a more complete description of AGS than modality or task labels alone. For example, systems such as ChatGen [93], ComfyGen [72], and GPT4Motion [97] all involve explicit planning, but they remain largely linear systems; by contrast, Idea2Img [20], RPG [21], and CountLoop [65] introduce iterative correction; and systems such as ComfyMind [22], GenArtist [18], Talk2Image [24], SceneCraft [16], and WorldCraft [17] operate with richer long-horizon control and more explicit state tracking.

4.1. Overview of the Taxonomy

The taxonomy is designed around three principles. First, the dimensions should be *separable*: a system's architectural form should not be conflated with its planning depth or feedback mechanism. For instance, a single-controller system may still perform long-horizon search, while a multi-agent system may remain a shallow pipeline. Second, the dimensions should be *mechanism-centered* rather than application-centered. What matters for AGS is not only whether a method generates images, videos, 3D scenes, or presentations, but how it converts user intent into a multi-step generative trajectory. Third, the taxonomy should be *comparative*, allowing systems with different output modalities to be analyzed under a shared lens.

 doi:10.20944/preprints202607.1997.v1

Under this view, AGS occupy a design space defined by six diagnostic questions. (1) How much autonomy does the system have in decomposing and revising a goal? (2) Is control concentrated in one controller or distributed across multiple agents? (3) Does the system rely on linear planning, reactive local correction, or explicit long-horizon search? (4) Are tools or executable operations invoked through goal-conditioned pipelines, dynamic routing, workflow synthesis, code generation, or software interaction? (5) What kinds of memory are maintained across steps or turns? (6) How are intermediate and final outputs evaluated and used for refinement? These questions recur across image generation, video storytelling, 3D scene creation, document authoring, restoration, and editing, suggesting that AGS are best understood as *systems of coordinated decision-making for generation*.

The capability hierarchy defined in Sec. 3.3 serves as the backbone of the taxonomy. It captures the progression from *one-way orchestration* to *reactive closed-loop generation* and finally to *goal-oriented, stateful, and often hierarchical control*. This section therefore does not redefine AGS, but expands the hierarchy into concrete mechanisms and representative system patterns. Table 3 summarizes the resulting taxonomy matrix. The remaining subsections unpack how architecture, planning, tool orchestration, memory, and feedback realize these differences in practice.

Table 3. Mechanism-centered taxonomy matrix for Agentic Generative Systems. The capability axis captures the degree of control over the generation trajectory, while the mechanism axes explain how this control is implemented.

Taxonomy axis	Diagnostic question	Common design choices	Representative examples
Capability level	How much control does the system exert over the generation trajectory?	L1 linear-chain orchestration; L2 feedback-conditioned local revision; L3 goal-oriented, stateful, or hierarchical orchestration.	L1: ChatGen [93]; L2: RPG [21]; L3: ComfyMind [22], SceneCraft [16].
Architecture	Where is control located, and how are responsibilities divided?	Centralized controller; collaborative multi-agent roles; planner–executor–critic hierarchy; native agentic generator.	GenArtist [18], StoryAgent [14], MovieAgent [25], VisionCreator [43].
Planning and control	How is user intent converted into an executable generation trajectory?	Prompt or layout planning; script, shot, or scene planning; reactive repair; search or backtracking over alternatives.	LayoutGPT [69], GPT4Motion [97], CountLoop [65], ComfySearch [106].
Tool orchestration	What actions can the system invoke, and how are tools composed?	Goal-conditioned pipelines; dynamic routing; workflow graph synthesis; code or GUI execution; model-internal orchestration.	HuggingGPT [67], ComfyGen [72], ComfyGPT [73], RETOUCHLLM [107].
Memory and state	What information persists across steps, turns, or branches?	Short-term working state; trajectory history; asset or entity memory; user preference memory; reusable skill memory.	Talk2Image [24], VideoMemory [105], WorldCraft [17], Agent Banana [19].
Feedback loop	How are intermediate or final artifacts evaluated and used for refinement?	Self-critique; VLM/evaluator feedback; executable verification; human feedback; multi-stage review and repair.	Idea2Img [20], ReasonEdit [99], PreGenie [108], Evaluation Agent [100].

4.2. Capability Levels: L1–L3

Building on the boundary conditions in Sec. 3.3, we use three capability levels to describe how much explicit control a system exerts over its generation trajectory. Here the emphasis is on how these levels appear in existing systems. The levels should be interpreted as a functional hierarchy rather than a strict historical timeline: a later system is not automatically more agentic than an earlier one; instead, the relevant question is what kind of control loop the system actually implements.

L1: linear-chain AGS. L1 systems introduce explicit goal-conditioned decomposition or orchestration, but execution remains largely linear. The system typically rewrites prompts, selects components, generates a workflow, or produces an executable intermediate representation before

 doi:10.20944/preprints202607.1997.v1

invoking downstream generators, with little or no runtime correction. Representative examples include ChatGen [93], which converts free-form dialogue into structured text-to-image generation instructions; ComfyGen [72], which maps prompts to adaptive ComfyUI workflows; LayoutGPT [69], which turns language into explicit layout constraints; GPT4Motion [97], which translates motion intent into Blender-oriented scripts; and FreeAudio [26], which first plans sound-event timing before audio synthesis. These systems are more agentic than single-shot generation because they externalize planning, but their control flow is still mostly one-way. They rarely revise strategy after observing execution outcomes, and memory is usually limited to the current request.

L2: reactive-loop AGS. L2 systems extend L1 by introducing explicit feedback-conditioned iteration. They monitor intermediate artifacts or diagnostic signals, diagnose local failures, and revise prompts, layouts, masks, tool calls, or editing actions accordingly. However, planning remains largely local or reactive rather than globally strategic. This level includes Idea2Img [20], which iteratively critiques generated drafts with GPT-4V; RPG [21], which combines recaptioning, planning, and complementary regional generation; CountLoop [65], which uses detector- and VLM-based feedback to repair multi-instance generation; MIRA [109] and ReasonEdit [99], which cast editing as a perception–reasoning–action loop; V-Stylist [110], which adjusts video stylization through self-reflection; and WeGen [101] and AutoStudio [102], which maintain multi-turn interaction state and revise outputs incrementally. The defining property of L2 is not merely repetition, but *feedback-conditioned iteration*: the system decides what to fix based on observed discrepancies. Still, these corrections are often local in scope and do not usually involve persistent world models, global goal tracking, or long-range replanning.

L3: goal-oriented orchestrated AGS. L3 systems move beyond local correction and operate with explicit long-horizon objectives, richer internal state, and more flexible coordination structures. They often use hierarchical planning, multi-agent decomposition, search, dynamic replanning, or persistent memory. Typical examples include ComfyMind [22], which performs tree-based planning over multimodal workflows; GenArtist [18], which builds and traverses planning trees for image generation and editing; Talk2Image [24], which maintains dialogue state and performs adaptive graph replanning in multi-turn generation; SceneCraft [16] and WorldCraft [17], which combine code generation, visual verification, and accumulated experience for 3D scene creation; StoryAgent [14] and MAViS [15], which coordinate long-form narrative video production across multiple specialized agents; and Agent Banana [19], which adds explicit context folding and long-range state tracking for multi-turn high-fidelity editing. At this level, generation is no longer a sequence of isolated fixes. Instead, it becomes a controlled process in which the system tracks goals over time, compares alternative sub-strategies, and revises execution plans under changing evidence.

The hierarchy is comparative rather than absolute. The L1–L3 taxonomy is intended to clarify capability differences, not to impose rigid boundaries. For example, a single-agent system with strong search and persistent state may be more capable than a shallow multi-agent pipeline. Similarly, a highly interactive editing framework may remain L2 if it mainly performs local reactive correction, while a 3D or video system becomes L3 once it maintains explicit global structure and supports strategic replanning. The value of the hierarchy is therefore analytical: it offers a shared language for comparing systems that differ in modality but confront similar control challenges. Borderline cases should be classified by the strongest control behavior that is routinely exercised, not by whether a paper uses the word "agent" or by how many modules are present.

4.3. Architectures of Agentic Generative Systems

Architecturally, AGS range from centralized controllers to distributed and hierarchical multi-agent organizations. These architectural choices strongly affect scalability, modularity, and controllability, even when the underlying generation models are similar. Architecture explains where control is located, how responsibilities are divided, and how intermediate state is shared; it complements, but does not by itself determine, the L1–L3 capability level.

Centralized single-agent architectures. A common design is to use one dominant controller—usually an LLM, MLLM, or native agentic generator—to manage the generation process. Examples

 doi:10.20944/preprints202607.1997.v1

include GenArtist [18], AgenticIR [95], Q-Agent [111], and VisionCreator-style systems [43,44], where a central reasoning module diagnoses the problem, selects tools or actions, and evaluates outcomes. This architecture is attractive because it keeps decision-making coherent and simplifies coordination. It also makes it easier to attribute errors to a single reasoning trace. However, centralized systems may struggle as tasks become longer, tool spaces become larger, or subtasks require heterogeneous expertise.

Collaborative multi-agent architectures. Many recent AGS instead distribute functionality across specialized roles. BannerAgency [112], PreGenie [108], MM-StoryAgent [113], and SlideGen [29] allocate different parts of the pipeline to agents responsible for strategy, layout, content generation, evaluation, or code realization. In long-form video systems such as StoryAgent [14], MAViS [15], MovieAgent [25], and AniME [114], distinct agents handle script writing, shot planning, character consistency, audio design, and video assembly. This role specialization can improve modularity and interpretability, since each agent has a narrower objective and interface. The cost is increased coordination complexity: message passing, conflict resolution, credit assignment, and shared state management become central design issues.

Hierarchical and planner–executor–critic designs. A particularly important architectural pattern is hierarchical control. Here, a high-level planner or coordinator decomposes the task, while lower-level executors perform concrete actions and critics verify results. WorldCraft [17], Agent Banana [19], M3 [115], and VULCAN [116] are representative. This organization is well suited to long-horizon generation because it separates strategic reasoning from tactical execution. It also maps naturally onto workflows where some modules reason in language while others act through APIs, code, or software interfaces.

Architecture as a capability enabler. Architectural sophistication alone does not determine capability level, but it often enables it. L1 systems are usually centralized and shallow. L2 systems often add reviewer or evaluator roles. L3 systems frequently require hierarchical decomposition, explicit coordination, or durable shared state. Thus, architecture is best treated as one axis of the taxonomy: it does not replace the capability hierarchy, but it helps explain how higher capability is implemented in practice.

4.4. Planning and Control Strategies

Planning is one of the defining dimensions of AGS because it mediates the gap between abstract intent and executable generation. Across the literature, planning ranges from prompt preprocessing to explicit search over workflow or scene trajectories. It operationalizes the transition from L1 to L3: stronger systems move from front-loaded decomposition toward feedback-aware revision, alternative exploration, and strategic replanning.

Linear pre-generation planning. The simplest form of planning occurs before generation begins. Systems such as ChatGen [93], PromptEnhancer [117], GPT4Motion [97], FreeAudio [26], and LLM-grounded Video Diffusion Models [118] first transform user intent into a better prompt, script, layout, timing plan, or motion description. This form of planning is valuable because it exposes structure that downstream generators cannot infer reliably from raw prompts alone. Yet it remains limited when intermediate failures require revision.

Reactive local control. L2 systems typically plan and control in a reactive manner. Idea2Img [20], CountLoop [65], Self-correcting LLM-controlled Diffusion Models [119], ReasonEdit [99], and Edit-Thinker [120] all exemplify a local control loop: generate, inspect, diagnose, and revise. This style is effective when errors are localized, such as missing objects, incorrect counts, distorted regions, or style mismatches. Its limitation is that it may optimize individual steps without maintaining a globally coherent trajectory or reconsidering whether the original plan remains appropriate.

Hierarchical and goal-oriented planning. As tasks become longer and more structured, AGS increasingly adopt hierarchical planning. Video storytelling systems such as StoryAgent [14], MAViS [15], MM-StoryAgent [113], and ScriptAgent [121] separate global narrative planning from shot-level or frame-level execution. 3D systems such as SceneCraft [16], WorldCraft [17], and CityX [28] decompose generation into subgoals like layout specification, asset retrieval, code synthesis, and physical arrangement. Presentation and educational video systems such as Code2Video [122], Paper2Video [13], and SlideGen [29] similarly separate discourse planning from rendering and refinement. In these

 doi:10.20944/preprints202607.1997.v1

systems, planning is not merely a better prompt; it is a structured decision process over intermediate representations such as outlines, scripts, layouts, scene graphs, code, and shot plans.

Search-based control. Some of the strongest AGS further extend planning into explicit search. ComfyMind [22] uses tree-based planning for workflow composition, ComfySearch [106] performs autonomous exploration in workflow space, AniMaker [94] uses MCTS for clip generation, CoSTA* [104] explores cost-sensitive tool paths for multi-turn editing, and PhotoAgent [123] frames retouching as exploratory aesthetic search. These methods are important because they shift AGS from deterministic decomposition toward deliberative control. Rather than committing to one plan, the system evaluates alternatives, reuses prior evidence, and backtracks when needed.

4.5. Tool Orchestration Mechanisms

Tool orchestration is another central axis of AGS. In these systems, intelligence often lies less in a monolithic generator than in the ability to compose and control heterogeneous tools, models, intermediate representations, and executable operations. This dimension defines the action space over which the planner operates, ranging from goal-conditioned model chains to executable workflows, code, GUI actions, and model-internal agentic generators.

Goal-conditioned pipelines and prompt-adaptive workflows. Some methods orchestrate tools through predefined but prompt-adaptive pipelines. ComfyGen [72] is a canonical example: the system selects and instantiates an appropriate workflow template based on prompt semantics. This design offers modularity and reliability, but the orchestration space is still constrained; it is most naturally associated with L1 unless runtime feedback can alter the trajectory.

Dynamic routing and model selection. Other systems perform explicit tool routing. HuggingGPT [67] and InternGPT [68] are broad multimodal examples, while generative systems such as SPAgent [92], AgenticIR [95], and JarvisIR [124] dynamically match tasks to specialized models based on semantics, degradation type, or tool capability. This routing-based design is crucial when no single model is sufficient across tasks, but routing alone becomes agentic only when it is tied to explicit goals and multi-step generation state.

Workflow graph synthesis. A more agentic form of orchestration appears in workflow-generation systems. ComfyGPT [73], ComfyMind [22], ComfySearch [106], ComfyUI-R1 [96], and ComfyBench [125] treat generation as the construction of an executable graph whose nodes and edges must satisfy functional constraints. Here, orchestration is itself the output of planning. This is an important departure from conventional prompt engineering, because the system must reason about compatibility, ordering, and execution validity in addition to visual quality.

Model-internal agentic orchestration. Native agentic generators such as VisionCreator and VisionCreator-R1 [43,44] internalize parts of planning, reflection, and generation inside a unified model. From the perspective of this taxonomy, they do not eliminate tool orchestration; rather, they change the implementation substrate of orchestration. The "tool" being controlled may be an internal reasoning-and-generation policy, but the same questions remain: whether goals are explicit, whether intermediate states are inspectable, whether feedback changes later actions, and whether the trajectory can be revised.

Code- and software-grounded orchestration. Another major branch uses code or real software as the orchestration substrate. SceneCraft [16], GPT4Motion [97], Code2Video [122], RETOUCHLLM [107], Chat2SVG [126], and Blender Alchemy [127] synthesize executable code or scripts to control rendering or editing tools. GUI-based benchmarks such as PSBench [98] go further by evaluating agents that operate real software interfaces directly. These settings reveal that tool orchestration is not just about choosing models; it can involve controlling external environments with stateful actions, constraints, and side effects.

Why tool orchestration matters. Tool orchestration is a key source of AGS flexibility. It enables systems to combine the strengths of specialist models, operate across modalities, and expose intermediate representations that are more editable and verifiable than pure pixel generation. At the same time, it introduces new challenges, including tool hallucination, invalid workflows, brittle interfaces, hidden tool state, and coordination overhead. These trade-offs explain why tool design and orchestration policy should be considered central taxonomic dimensions rather than implementation details.

 doi:10.20944/preprints202607.1997.v1

4.6. Memory Mechanisms in Agentic Generative Systems

Memory determines whether AGS can maintain consistency beyond the immediate step. Although many systems are still weak in this respect, memory has become increasingly important as generation tasks shift from one-shot artifact creation to long-horizon co-creation. In the L1–L3 hierarchy, memory is one of the main mechanisms that turns local reactive correction into persistent long-horizon control.

Short-term working memory. At the most basic level, AGS use working memory to preserve recent dialogue context, local editing state, or currently active constraints. Interactive systems such as WeGen [101], MagicQuill [103], and X-Planner [128] depend on such short-horizon state to remain responsive across local edits. This form of memory is usually transient and closely tied to the current turn or subtask.

Trajectory and interaction memory. A stronger form stores the evolving trajectory of the task. Talk2Image [24] maintains structured dialogue-state memory across multi-turn generation and editing; Agent Banana [19] uses context folding to compress interaction history while preserving editing intent; and search-based systems such as CoSTA* [104] track previously explored branches and tool decisions. This kind of memory is important because agentic generation often involves revising earlier choices rather than simply appending new ones. It also provides the evidence base needed for search, rollback, and cost-aware tool selection.

Entity, asset, and scene memory. Long-form video and 3D systems often require more explicit memory about persistent entities. VideoMemory [105] introduces dynamic memory banks for characters, props, and backgrounds; CoAgent [129] maintains explicit global context for long-video consistency; AniME [114] uses a global asset memory; and story visualization systems such as AutoStudio [102] and DreamStory [130] rely on persistent subject representations to reduce identity drift. Without such memory, AGS frequently fail on one of their most important promises: preserving coherence across long horizons.

Experience and skill memory. Some L3 systems go further by remembering not only content state but also procedural experience. SceneCraft [16] stores reusable generation skills in an external library, WorldCraft [17] accumulates tool-use experience through self-corrective modules, and AesopAgent [131] uses retrieved prior expertise to improve future story-to-video production. This is especially important for AGS because repeated interaction with tools or environments can generate reusable operational knowledge.

Memory and capability level. Memory is one of the clearest differentiators between L2 and L3 systems. L1 systems typically have little or no persistent memory. L2 systems may preserve short-term state for iterative correction. L3 systems, by contrast, often require explicit memory abstractions to support long-horizon consistency, multi-turn interaction, personalization, or cumulative self-improvement. For this reason, when long-horizon control is claimed, memory should be viewed not as a peripheral add-on but as a major driver of capability scaling in AGS.

4.7. Feedback and Evaluation Loops

Feedback is the mechanism that turns orchestration into closed-loop generation. Without feedback that changes later decisions, even sophisticated planning remains essentially open-loop. Recent AGS differ substantially in what feedback they use, how often they invoke it, and how tightly it is coupled to planning. In capability terms, feedback marks the transition from L1 to L2, while richer feedback integrated with memory and planning enables L3-style strategic replanning.

Self-critique and internal reflection. Many L2 and L3 systems use internal model-based reflection to inspect intermediate outputs. Idea2Img [20], ReasonEdit [99], EditThinker [120], TWIG [132], and VisionCreator-R1 [44] all rely on a form of self-critique in which the model or an auxiliary reasoning module explains what went wrong and proposes revisions. This internalization of evaluation is important because it makes feedback part of the generative trajectory itself, rather than a score reported only after generation has finished.

 doi:10.20944/preprints202607.1997.v1

External evaluators and critics. A second family uses dedicated evaluators. Evaluation Agent [100] and VideoGen-Eval [133] treat assessment as an explicit agentic process; CountLoop [65], ShowTable [134], and many story or video systems use VLM-based critics or specialized scoring modules to compare outputs against semantic, structural, or aesthetic criteria. Such critics provide stronger modularity and often better interpretability than implicit self-judgment, especially when the criteria involve counting, grounding, temporal continuity, or source fidelity.

Execution and environment feedback. In tool-heavy AGS, execution itself becomes a feedback source. ComfyGPT [73] and ComfyUI-R1 [96] expose whether a workflow is executable; GUI benchmarks such as PSBench [98] expose whether a software action succeeds and whether the current interface state matches the intended plan. Code-based systems such as SceneCraft [16] and Code2Video [122] similarly obtain feedback from rendered outputs, syntax validity, or downstream tool behavior. This kind of feedback is especially valuable because it grounds reasoning in the real operational environment and detects failures that may not be visible from prompts or generated media alone.

Environment, simulator, and physics feedback. Some video and embodied-generation systems obtain feedback from simulators or external world models. GPT4Motion [97] is better viewed as physics-aware open-loop planning through Blender-oriented scripts rather than as a feedback-loop method; more explicitly closed-loop settings include VideoAgent [135], which uses external feedback for embodied planning video generation, and ORCA-style closed-loop avatar systems [136], which couple generation with active world modeling. These systems highlight an important direction for AGS: feedback need not come only from textual critique or perceptual comparison, but also from interaction with a simulated or partially observable environment.

Reward-driven and learned feedback. Finally, several systems incorporate learned reward or preference signals. RePrompt [137], MIRA [109], ImageEdit-R1 [138], and VTool-R1 [139] use reinforcement learning or reward-based optimization to shape editing trajectories, prompt adaptation, or multimodal tool use. This direction is important because it moves AGS toward scalable policy improvement rather than heuristic correction alone.

Feedback as the core of closed-loop generation. Across these variants, the common pattern is clear: feedback determines whether AGS can recover from errors, maintain consistency, and improve beyond first-pass generation. The strongest systems do not treat evaluation as a terminal score; they integrate it as a control signal that influences planning, tool selection, and memory updates. In this sense, feedback is not just another module. It is one of the defining properties that turns multi-step generation into adaptive, closed-loop agentic generation.

Summary. Taken together, the taxonomy shows that AGS are best understood as systems organized along interacting dimensions rather than as a single method family. Capability level captures how far a system goes from linear orchestration to stateful long-horizon control. Architecture explains how decision-making is distributed. Planning and tool orchestration explain how goals are converted into executable trajectories. Memory and feedback determine whether those trajectories remain coherent, adaptive, and revisable. This separation between capability and mechanism is important: a system is not L3 because it has many agents or many tools; it is L3 when those mechanisms support strategic control over the evolving generation trajectory. The taxonomy therefore provides a unified framework for analyzing current AGS and for identifying where future progress is likely to emerge: not only in stronger generators, but in better control structures for generation.

5. Applications and Domain Instantiations of Agentic Generative Systems

Agentic Generative Systems (AGS) appear across a wide range of generation domains, but the form of agency changes with the target artifact and the representation that must be controlled. Sec. 4 classified AGS by their control mechanisms; this section complements that view by asking how those mechanisms instantiate in concrete application domains. We organize the discussion by the *primary output artifact*: cross-domain orchestration, images, videos, audio, 3D scenes/assets, and document–presentation authoring. Across these domains, the central variable is the *object of control*: spatial

 doi:10.20944/preprints202607.1997.v1

composition for images, temporal coherence for video, event timing and affect for audio, editable world state for 3D, and information structure plus editability for communicative media. We retain the L1–L3 hierarchy in an operational sense. L1 systems externalize planning or workflow specification, but do not substantially revise plans based on generated outputs. L2 systems add output-conditioned local correction loops, such as critique, reprompting, repair, or iterative refinement. L3 systems maintain richer cross-step state and support strategic replanning over long horizons, often through persistent memory, role-specialized collaboration, or search over tool trajectories. To avoid taxonomy drift, we treat benchmarks, evaluation agents, and safety-specific attack or moderation systems as supporting material rather than core application instances, unless they directly instantiate a generation loop. Table 4 gives a system-level overview of representative AGS across these domains, summarizing their capability levels, architectures, planning, memory, feedback, and tool choices at a glance.

Table 4. Representative Agentic Generative Systems organized by output domain and capability level (**L1** linear-chain, **L2** reactive-loop, **L3** goal-oriented orchestrated). **Arch**: Single-controller, Multi-agent, Hierarchical/planner–executor–critic. **Plan**: Linear, Layout, Workflow, Reactive, Hierarchical (Hier), Search, Code. **Mem**: – none, ST short-term, Traj trajectory, Ent entity/asset, Skill. **Feedback**: – none, Self self-critique, VLM evaluator, Exec execution, Multi multi-critic. ✓ = public code available.

System	Lvl	Arch	Plan	Mem	Feedback	Venue	Code
CROSS-DOMAIN ORCHESTRATION							
Visual ChatGPT [66]	L1	S	Reactive	ST	–	arXiv'23	✓
HuggingGPT [67]	L1	S	Reactive	ST	Exec	NeurIPS'23	✓
ComfyGPT [73]	L2	M	Workflow	ST	Exec	arXiv'25	✓
ComfyMind [22]	L3	H	Search	Traj	Exec+VLM	NeurIPS'25	✓
IMAGE GENERATION & EDITING							
ChatGen [93]	L1	S	Linear	–	–	CVPR'25	✓
LayoutGPT [69]	L1	S	Layout	–	–	NeurIPS'23	✓
Idea2Img [20]	L2	S	Reactive	ST	Self	ECCV'24	✓
RPG [21]	L2	S	Reactive	ST	Self	ICML'24	✓
CountLoop [65]	L2	M	Reactive	ST	VLM	arXiv'25	
GenArtist [18]	L3	S	Search	Traj	Self+VLM	NeurIPS'24	✓
Talk2Image [24]	L3	M	Hier	Traj	VLM	AAAI'26	
Agent Banana [19]	L3	H	Hier	Traj	Self	arXiv'26	✓
4KAgent [140]	L3	H	Hier	Traj	VLM	NeurIPS'25	✓
VIDEO GENERATION							
GPT4Motion [97]	L1	S	Code	–	–	CVPRW'24	✓
LLM-grounded VDM [118]	L1	S	Layout	–	–	ICLR'24	✓
GENMAC [141]	L2	M	Reactive	ST	VLM	AAAI'26	✓
VideoAgent [135]	L2	S	Reactive	ST	VLM+Exec	NeurIPSW'25	✓
StoryAgent [14]	L3	M	Hier	Ent	Self	arXiv'24	
MovieAgent [25]	L3	M	Hier	Ent	Multi	arXiv'25	✓
AUDIO GENERATION							
WavJourney [23]	L1	S	Linear	–	–	TASLP'25	✓
FreeAudio [26]	L1	S	Linear	–	–	ACMMM'25	
PodAgent [142]	L2	M	Reactive	ST	Self	ACL'25	✓
Dopamine Audiobook [27]	L2	M	Reactive	ST	Self	arXiv'25	
ReelWave [143]	L3	M	Hier	–	Self	arXiv'25	
3D SCENE & ASSET GENERATION							
SceneX [144]	L1	S	Code	–	–	AAAI'25	
Blender Alchemy [127]	L2	S	Search	ST	Exec	ECCV'24	✓
SceneCraft [16]	L3	H	Code	Skill	VLM	ICML'24	
WorldCraft [17]	L3	H	Code	Skill	Self	arXiv'25	
CityX [28]	L3	M	Code	–	Self	arXiv'24	✓
DOCUMENT & PRESENTATION AUTHORING							
POSTA [64]	L1	S	Linear	–	–	CVPR'25	
PresentAgent [145]	L1	S	Linear	–	–	EMNLP'25	✓
PreGenie [108]	L2	M	Reactive	ST	Self	EMNLP'25	
SlideGen [29]	L3	M	Hier	Traj	VLM	arXiv'25	✓
Paper2Video [13]	L3	M	Hier	Traj	VLM	NeurIPSW'25	✓

 doi:10.20944/preprints202607.1997.v1

5.1. Cross-Domain and General-Purpose AGS

Before discussing individual modalities, it is useful to note that some systems instantiate AGS most directly as *general-purpose orchestration layers* over heterogeneous generation tools. Here the controlled object is not a single image, clip, or document, but a workflow graph that binds user goals to models, tools, intermediate representations, and execution traces. Early systems such as *Visual ChatGPT* and *InternGPT* demonstrated that language models can act as central coordinators over multiple visual foundation models, enabling multi-turn generation, editing, and visual reasoning through explicit tool selection and sequential execution [66,68]. At a broader systems level, *HuggingGPT* generalized this idea by turning model invocation into a planning-and-scheduling problem over a large external model ecosystem, making generation one instance of a larger tool-centric control loop [67].

More recent work pushes this orchestration view toward integrated AGS controllers. *MAGiC*, *ComfyMind*, *ComfySearch*, and *SPAgent* treat multimodal generation as structured workflow construction and coordination rather than isolated model calls [22,92,106,146]. Other recent frameworks such as *UniVA*, *MultiMedia-Agent*, *MAGUS*, and *LLM-I* extend this pattern toward broader multimodal understanding and generation under shared planning, memory, and tool-use abstractions [147–150]. Finally, native agentic generation models such as *VisionCreator* and *VisionCreator-R1* suggest a complementary trajectory in which part of planning, reflection, and creation is internalized into a unified generator rather than implemented purely as an external wrapper [43,44].

These cross-domain systems are important for two reasons. First, they make the modality-agnostic nature of AGS concrete: the same principles of goal tracking, tool orchestration, and feedback-driven revision can support images, video, audio, 3D, and document-centric media. Second, they clarify that AGS are not merely "LLMs plus tools," but systems in which generation is explicitly embedded in a controllable decision process.

5.2. Image Generation and Editing

Image generation and editing remain one of the clearest domains for AGS because the gap between user intent and executable visual specification is often large. The controlled object is a spatial–semantic image state: composition, layout, identity, style, local edit scope, and repair priorities must be made explicit enough for a generator or editing tool to act. Since users rarely provide all of this information in a single prompt, image AGS often externalize goal representation into structured layouts, editable plans, or repair trajectories before or during synthesis.

L1: explicit specification before synthesis. L1 image AGS improve controllability by making visual intent explicit before generation, while keeping execution largely one-way. A first cluster focuses on prompt and workflow construction. *ChatGen*, *PromptEnhancer*, and *ComfyGen* translate free-form requests into model-friendly prompts, parameter choices, or workflow templates, thereby turning underspecified user intent into executable generation inputs [72,93,117]. A second cluster externalizes structural control. *LayoutGPT* predicts explicit 2D or 3D layouts from language, *MultiRef* organizes multiple visual references into controllable generation constraints, and *POSTA* decomposes poster creation into background, layout, and typography planning [64,69,151]. Other systems expose narrower but still useful control interfaces. *EmoCtrl* injects explicit affective control signals into generation, while *MagicQuill* converts user interaction into localized edit intent without requiring the user to specify every operation textually [103,152]. Taken together, these systems are best understood as linear-chain AGS: they externalize goal understanding and planning, but the plan is mostly fixed before synthesis begins.

L2: reactive generation and editing loops. L2 systems move beyond front-loaded planning by adding output-conditioned revision. A prominent branch treats image synthesis as a plan–generate–evaluate–revise loop. *Idea2Img* uses iterative self-refinement with visual feedback, while *Draw ALL Your Imagine*, *RPG*, *LayerCraft*, and *CoT-lized Diffusion* combine structured reasoning with spatial or regional control to progressively correct compositional errors [20,21,153–155]. *Rare-to-Frequent*, *RePrompt*, *TWIG*, *Uni-CoT*, and *Visual-Aware CoT* further illustrate a broader L2 pattern in which

 doi:10.20944/preprints202607.1997.v1

reasoning is interleaved with generation or used to rewrite prompts, check constraints, and repair mismatches during inference [132,137,156–161]. For difficult compositional or foreground-conditioned cases, *CountLoop*, *Self-correcting LLM-controlled Diffusion*, *Maestro*, and *Marmot* show that explicit critique is particularly useful for counting, attribute binding, object integrity, and layout repair [65,119,162–166].

A second L2 branch focuses on instruction-based editing. *X-Planner* decomposes complex edits into subtasks with automatically generated spatial control signals [128]. *MURE*, *MIRA*, *ReasonEdit*, *ARTIE*, *CoEditor++*, and *EditThinker* all instantiate some form of perception–reasoning–action loop, where intermediate edits are inspected and revised rather than accepted in one pass [99,109,120,167–171]. Interactive systems such as *WeGen*, *AutoStudio*, and *T2I-Copilot* strengthen this reactive pattern by explicitly resolving ambiguity, maintaining user-facing dialogue state, or responding to multi-turn corrections [101,102,172–174].

A third L2 branch appears in restoration, retouching, and domain-constrained repair. *Clarity ChatGPT*, *LLMRA*, *Hybrid Agents for Image Restoration*, *JarvisIR*, and *Restore-R1* treat restoration as sequential diagnosis plus tool routing under perceptual feedback [124,175–179]. *Agentic Retoucher*, *PhotoArtAgent*, *RETOUCHIQ*, and *RETOUCHLLM* apply similar logic to artistic retouching, where the system iteratively balances semantic fidelity, local defect repair, and aesthetic quality [107,180–182]. Compared with L1 systems, the defining difference here is not simply that more modules are present, but that intermediate images actively influence the next action.

L3: long-horizon orchestration for creative workflows and professional editing. L3 image AGS differ from L2 primarily in the persistence of goal state and the flexibility of execution strategy. Instead of only correcting local mistakes, they maintain structured subgoals, search over tool trajectories, or preserve cross-turn editing state over longer horizons. *GenArtist* and *GenAgent* are representative: both separate reasoning from raw image synthesis and emphasize iterative tool-mediated optimization under reflection [18,183]. In multi-turn settings, *Talk2Image* and *Agent Banana* make persistent intent tracking explicit by maintaining dialogue-aware state and triggering adaptive replanning when accumulated edits drift away from the user goal [19,24]. Recent multi-agent systems such as *coDrawAgents*, *M3*, *CRAFT*, and collaborative reinforcement-learning-based generation push this direction further by decomposing compositional generation into verifiable sub-constraints and coordinating specialized agents for planning, checking, editing, and validation [115,184–186].

Mature L3 behavior often appears in professional editing and restoration, where quality constraints are high and the space of useful actions is large. *4KAgent*, *AgenticIR*, *MAIR*, *Q-Agent*, and *RestoreAgent* formulate restoration as a long-horizon planning problem over diagnosis, routing, execution, quality testing, and replanning [95,111,140,187–189]. *JarvisArt*, *JarvisEvo*, and *PhotoAgent* treat photo retouching as exploratory search over professional operators, with reflection or evaluator models deciding whether to continue, revise, or roll back edits [123,190,191]. Another notable trend is the move toward software-operating or tool-combinatorial agents, including *ImageEdit-R1*, *Lego-Edit*, GUI-oriented editing agents studied in *PSBench*, *CoSTA**, and *VTool-R1*, where the system no longer merely rewrites prompts but actively navigates editable tool spaces [98,104,138,139,192]. Finally, knowledge-seeking systems such as *Mind-Brush* and *World-To-Image* indicate that L3 image AGS may also need external retrieval and world-grounded reasoning when user requests involve long-tail concepts beyond the prior of any single generator [193,194].

Taken together, image AGS evolve from explicit prompt and layout specification, to reactive repair and iterative editing, and finally to goal-oriented orchestration over editing tools, memory, and search. This progression is especially revealing because it separates prompt improvement from stronger agency: robust image AGS must inspect, diagnose, and strategically revise the generation process itself.

5.3. Video Generation

Compared with image generation, video generation places much stronger pressure on long-horizon planning, temporal consistency, multimodal synchronization, and cross-scene identity preservation. The controlled object is a temporally extended production state that links script, shot order,

 doi:10.20944/preprints202607.1997.v1

motion, camera behavior, characters, audio, and post-processing decisions. The agentic advantage is therefore especially visible in this domain: planning is needed to convert abstract intent into shots and motion, and feedback is needed to maintain coherence across time.

L1: temporal planning without deep replanning. L1 video AGS externalize temporal structure before rendering, but remain largely feed-forward during execution. *DirecT2V*, *LLM-grounded Video Diffusion Models*, and *FlowZero* all convert prompts into frame- or trajectory-level temporal instructions, thereby treating video generation as structured scene planning rather than direct one-shot synthesis [118,195,196]. *MotionZero* and *VLIPP* likewise inject motion or physical priors before synthesis, using language or multimodal reasoning to produce more plausible trajectories [197,198]. *GPT4Motion* makes the planning layer even more explicit by generating Blender scripts that simulate physical motion before downstream rendering [97]. Although these systems vary in representation—frame directives, dynamic syntax, motion priors, or executable code—they share the same L1 structure: generation quality improves because temporal intent becomes an explicit intermediate artifact.

L2: reactive refinement for physicality, composition, and controllability. L2 video systems add feedback-aware correction after the initial plan. A first branch targets physical plausibility and motion quality. *VideoAgent* uses VLM or environment feedback to iteratively improve generated videos for embodied planning, while physics-aware refinement systems such as *PhyT2V*, *Bootstrapping Physics-Grounded Video Generation through VLM-Guided Iterative Self-Refinement*, and *SketchVerify* detect or anticipate physical violations and revise generation accordingly [135,199–201]. *VISTA* extends this test-time self-improvement view to multiple experts that critique visual, audio, and contextual aspects jointly [202].

A second L2 branch focuses on compositional or domain-specialized control. *GENMAC* explicitly instantiates a design–generate–verify–redesign loop for compositional text-to-video generation [141]. *Compositional 3D-aware Video Generation with LLM Director* and *GenHSI* introduce structured 3D reasoning into video planning, enabling more reliable object- or human-scene interaction control [203,204]. *MotionAgent*, *V-Stylist*, *FairyGen*, *FilMaster*, *AutoVFX*, *MORA*, and *LangDriveCTRL* show that the same reactive logic extends to stylization, cinematic generation, visual effects, and structured scene editing [110,205–212]. Across these systems, the essential L2 move is to let generated video or intermediate motion plans influence the next decision, without yet requiring persistent long-horizon task state.

L3: long-horizon narrative orchestration and production workflows. Mature video AGS treat video creation as a production pipeline rather than a model invocation. This is clearest in long-form storytelling. *VGTeam*, *StoryAgent*, *DreamFactory*, *MovieAgent*, *MAViS*, *VideoGen-of-Thought*, *VideoDirectorGPT*, *CoAgent*, *ScriptAgent*, and *VideoMemory* all maintain cross-stage state as they move from script writing and storyboard planning to character control, clip generation, and final review [14,15,25,105,121,129,213–219]. Related animation-oriented systems such as *Anim-Director*, *AniMaker*, *AniME*, *AesopAgent*, *PersonaVlog*, and *Vlogger* similarly emulate creative teams, often with persistent asset stores, reviewer roles, or search procedures to preserve narrative and visual consistency across long horizons [94,114,131,220–223].

Another L3 cluster expands from narrative generation to more general multimodal or simulation-centered video production. *SPAgent* turns video generation and editing into adaptive task decomposition plus model selection over a growing tool library, while *EditDuet* extends this tool-oriented view to non-linear video editing with timeline state, critic feedback, and rendering tools [92,224]. *MoA-VR* shows that the same L3 pattern also applies to video restoration, decomposing all-in-one repair into degradation identification, agent routing and restoration, and quality assessment rather than treating restoration as a single model call [225]. *Kubrick*, *FilmAgent*, and *MoReGen* bring CGI or simulation tools into the loop, making camera control, physical motion, and code-level scene construction part of the agentic workflow [226–229]. Executable event-graph systems make the intermediate representation even more explicit, translating text into tool-constrained event graphs before video realization [230]. *Orator*, *UniVA*, *Hollywood Town*, and *Vibe AIGC* further stress that long-form video AGS increasingly behave like workflow managers over heterogeneous tools, evaluation nodes, and intermediate states

 doi:10.20944/preprints202607.1997.v1

[147,231–233]. Music-video systems such as *AutoMV* add a cross-modal production variant, coordinating screenwriting, directing, verification, music-information-retrieval tools, and visual generation under shared character state [234]. At the limit, systems such as active video-avatar world models and *SPIRAL* blur the line between generative video pipelines and closed-loop agents, suggesting a future in which video AGS are not only content producers but also planners acting inside generated environments [136,235].

Overall, the video domain makes the value of AGS especially visible. L1 methods show that explicit temporal planning already improves coherence; L2 methods add corrective loops for physics, composition, and editing; and L3 systems elevate video generation into narrative and multimodal production orchestration with memory, role specialization, and strategic replanning.

5.4. Audio Generation

Audio generation exposes a different but equally important face of AGS. Instead of spatial composition, the controlled object is an acoustic timeline: temporal structure, event alignment, emotional expressivity, multi-source synchronization, and long-form coherence across speech, sound effects, and music. These properties make audio a natural domain for agentic decomposition, because useful systems must often decide *what* should happen sonically, *when* it should happen, and *how* multiple generators should be coordinated.

L1: event and timing plans before synthesis. L1 audio AGS usually improve controllability by introducing explicit narrative or temporal plans while preserving mostly feed-forward execution. *WavJourney* uses LLM-generated scripts to compose long-form audio through expert modules, showing an early programmatic view of audio creation [23]. *FreeAudio* makes event timing explicit by converting text into structured time plans for long-form text-to-audio generation [26]. *AudioStory* extends this idea to narrative audio by decomposing long prompts into temporally organized sub-queries that preserve scene-to-scene continuity [236]. Related planning ideas also appear in multimodal music generation: *Mozart's Touch* introduces an LLM bridge between visual inputs and musical prompts, while *ComposerX* uses multiple symbolic composition agents to coordinate music generation [237–239]. These systems already instantiate AGS in a weak form, because they externalize high-level intent into executable temporal or semantic structure before audio synthesis begins.

L2: reactive refinement for affect, personalization, and alignment. L2 audio systems strengthen this foundation by adding feedback-aware adaptation. *Dopamine Audiobook* is representative: it uses role-specialized agents for speech and audio design, and explicitly evaluates naturalness, expressivity, and immersion in a refinement loop [27]. *PodAgent* brings similar ideas to podcast generation through host–guest–writer role playing, while *Audio-Agent* decomposes generation, editing, and composition into smaller tasks under language-model control [142,240]. A related cluster centers on personalization and cross-modal semantics. *VibeMus* proactively models user preference during music generation, while *Zero-Effort Image-to-Music Generation* uses multimodal retrieval and self-refinement to synthesize more semantically aligned music from images [241,242]. In all these cases, audio is no longer produced from a fixed script alone; the system adapts after evaluating or disambiguating what it has already generated.

L3: long-horizon orchestration across speech, music, sound effects, and video context. Mature audio AGS treat sound creation as a long-horizon production problem. This is clearest in long-video soundtrack generation. *Long-Video Audio Synthesis with Multi-Agent Collaboration* and *Orchestrating Audio* break long-form audio generation into scene segmentation, script generation, audio design, and synthesis, using multi-agent coordination and retrieval to preserve acoustic continuity across scenes [243–245]. *ReelWave* pushes this direction further by explicitly separating functions such as sound directing, foley, dubbing, music, and mixing through multimodal LLM conversation [143]. *AudioGenie* broadens the same idea into multi-audio generation with generation and supervision teams plus explicit correction loops [246]. Finally, open systems such as *WeaveMuse* suggest a broader future in which music generation, understanding, editing, and multimodal coordination are unified under a single agentic substrate [247].

 doi:10.20944/preprints202607.1997.v1

In sum, audio AGS show that agency matters not only when artifacts are visually complex, but also when outputs are temporally layered, emotionally expressive, and multimodally conditioned. L1 systems improve control through event and timing plans, L2 systems add personalization and refinement loops, and L3 systems turn audio creation into long-horizon orchestration across speech, music, sound effects, and audiovisual context.

5.5. 3D Scene and Asset Generation

3D scene and asset generation makes the agentic nature of generation especially explicit. Here the controlled object is not merely a perceptual output but an editable world with geometry, layout, affordances, and often downstream simulation requirements. Compared with 2D generation, 3D creation more naturally exposes planning, code synthesis, asset retrieval, physical verification, and persistent world-state revision.

L1: layout, program, and procedural planning. L1 systems improve controllability by making scene structure explicit before realization. *Open-Universe Indoor Scene Generation* uses an LLM to synthesize programs for room layout and object relations, then retrieves assets from open object databases through multimodal search [248]. *SceneX* similarly uses an LLM planner to orchestrate terrain generation, asset retrieval, and placement over a procedural content generation toolkit [144]. *FreeScene* and *LayoutGPT* convert free-form language or multimodal inputs into structured layouts that constrain downstream 3D scene synthesis [69,249]. These methods are clearly agentic in the sense of goal decomposition and tool use, but they remain mostly linear once the plan has been produced.

L2: feedback-guided layout optimization and interactive editing. L2 3D AGS add evaluators or local correction loops on top of initial plans. Several works focus on scene layout refinement. *Agentic 3D Scene Generation with Spatially Contextualized VLMs*, *LayoutVLM*, *Scenethesis*, *Direct Numerical Layout Generation*, and *ReSpace* all combine language-guided planning with visual, geometric, or preference-based feedback to iteratively correct layout, object placement, or semantic alignment [250–255]. A second branch targets interactive editing and object-level grounding. *Blender Alchemy* uses rendered feedback and iterative search to improve 3D edits; *Chat-Edit-3D*, *EditRoom*, and *ScanEdit* decompose language instructions into structured editing operations and then refine the result through guided optimization or hierarchical execution; and error-driven scene-editing systems diagnose grounding failures before targeted 3D repair [127,256–259]. Related systems such as *ChatGarment*, *SmartAvatar*, and *SceneGenAgent* show that the same L2 pattern also applies to structured asset domains like garments, avatars, and industrial scenes: generation is no longer one-shot, but correction remains mostly local rather than globally strategic [260–262].

L3: world-building systems with persistent state and tool orchestration. Mature 3D AGS manage creation as a long-horizon workflow over code, assets, geometry, and evaluation. *SceneCraft* is a representative anchor: it synthesizes scenes as code, runs an inner loop of visual correction, and accumulates reusable skills in an outer loop [16]. *WorldCraft*, *ShapeCraft*, *LL3M*, *EZBlender*, *From Idea to Co-Creation*, and *VIGA* similarly treat 3D modeling as a planner–executor–critic style process over coding, rendering, verification, and user or critic feedback [17,263–269]. CAD-oriented systems extend the same code-and-verification view: *Physics-in-the-Loop* embeds CAD and FEA tools into iterative engineering-design verification, while *Zero-to-CAD* uses execution, validation, documentation lookup, and repair to synthesize interpretable CAD programs [270,271]. In these systems, the agent no longer outputs a scene directly; it maintains and revises an editable world representation across many dependent steps. Other scene-level systems such as *SceneWeaver*, *SceneSmith*, *SAGE*, *SceneReVis*, *VULCAN*, global-local tree-search-based generation, and *HSM* make this even clearer by combining hierarchical reasoning, explicit tool calls, physical constraints, and repeated self-correction to produce simulation-ready environments [116,272–278].

A second major L3 thread centers on city-scale and interactive world generation. *CityX*, *I-Design*, *CityGenAgent*, *RAISECity*, *ImmerseGen*, *Yo'City*, and *Majutsu City* all support large-scale world construction through some combination of multi-agent planning, code generation, self-critique, and multimodal editing [28,279–284]. *ChatDyn* extends this thread from static city assets to language-driven

 doi:10.20944/preprints202607.1997.v1

multi-actor dynamics in street scenes, where agents plan motions and stateful executors realize the evolving scene [285]. *ChatSim* further shows that long-horizon orchestration is essential when 3D generation must support interactive simulation and user-directed scene modification rather than static content alone [286]. Taken together, these works suggest that mature 3D AGS are not simply better geometric generators; they are world-building systems that maintain state, invoke heterogeneous tools, and revise scene structure under explicit semantic and physical constraints.

Overall, 3D generation is arguably the domain in which the AGS abstraction is most naturally realized. L1 methods externalize layout and program structure, L2 methods add visual or geometric refinement, and L3 methods transform generation into persistent world construction with code, assets, memory, and simulation-aware replanning.

5.6. Document, Presentation, and Multimedia Authoring

Document and presentation authoring form an important application frontier for AGS because the output is usually a *communicative artifact* rather than a purely perceptual one. The controlled object is an information design state: source fidelity, rhetorical hierarchy, layout, editable structure, modality selection, and audience-facing clarity must remain aligned. Compared with open-ended image or video generation, these tasks require systems to preserve information structure, support editability, coordinate multiple modalities, and optimize for communicative effectiveness. We therefore place here systems whose primary goal is to transform structured source content—papers, tables, figures, poster requirements, or presentation intent—into editable or presentable artifacts. By contrast, systems whose main contribution is open-ended cinematic storytelling are discussed under video.

L1: structure-first authoring. L1 systems improve authoring quality by externalizing content structure before rendering. *POSTA* is a representative case at the poster-design level: it decomposes generation into background synthesis, layout planning, and typography rendering, thereby treating poster creation as a structured design workflow rather than a single image-generation call [64]. *PresentAgent* instantiates a similar idea for presentation video generation by decomposing long documents into semantic segments, planned visual slides, and aligned narration [145]. These systems remain mostly feed-forward once the structure is fixed, but they show that document-centric AGS begin with explicit intermediate representations of communicative intent.

L2: editable communication artifacts and refinement loops. L2 systems strengthen this foundation by adding review, repair, and iterative refinement. *BannerAgency* is exemplary: it decomposes banner design into multiple specialized roles and keeps the final artifact editable as vector graphics or design code [112]. *ShowTable*, *VisPainter*, and *Chat2SVG* follow the same general principle that communicative artifacts should be revised under explicit structural or visual feedback rather than generated once and frozen [126,134,287]. A parallel line appears in presentation authoring. *PreGenie* uses a generate–review–optimize loop over slide content and layout, while slide-updating systems such as *SlideAgent* preserve user-defined templates through instruction grounding, data retrieval, and chart or text revision [108,288]. Scientific short-form systems such as *SciTalk* and educational media systems such as *Code2Video* show that planner–critic loops are especially useful when correctness, structure, and presentation design must all be satisfied at once [122,289].

L3: long-horizon document-to-media transformation and office-style production. Mature systems in this application area treat authoring as a long-horizon production workflow over many interdependent stages. *Preacher* and *Paper2Video* are central examples: both transform scientific papers into explanatory videos through coordinated long-document understanding, script extraction, visual planning, narration, and iterative alignment checks [12,13]. The same logic appears in more presentation-centric office systems. *SlideGen* combines outline planning, multimodal content mapping, layout optimization, note generation, and visual-in-the-loop refinement to produce slide decks as editable artifacts rather than static renderings [29]. *SlideTailor* extends this direction by modeling speaker- or user-specific presentation preferences, making personalization part of the maintained goal state rather than an afterthought [290]. Related design systems such as *PosterForest* and *APEX* similarly elevate poster generation and editing into multi-stage planning problems with reviewer-

 doi:10.20944/preprints202607.1997.v1

driven adjustment and tool-grounded editing operations [291,292]. Beyond slides and posters, recent scientific and educational media systems suggest that authoring AGS are gradually becoming full communication systems that manage source understanding, narrative sequencing, asset selection, and delivery style under a unified control loop [289,293,294].

In short, document and presentation authoring reveal a distinctive strength of AGS: they bridge content understanding with artifact realization. L1 systems externalize rhetorical or layout structure, L2 systems add iterative review to improve editability and communicative quality, and L3 systems orchestrate full production pipelines that transform papers, tables, poster requirements, and presentation intent into refined multimedia artifacts.

5.7. Summary Across Domains

Table 4 maps representative systems across all six domains onto their capability levels and mechanism choices. Across modalities, the same capability progression reappears in domain-specific form: L1 systems externalize intent into explicit plans, layouts, scripts, or workflows; L2 systems make generation reactive by conditioning subsequent actions on intermediate outputs; and L3 systems add persistent state, strategic replanning, and long-horizon coordination over tools, memory, and specialized roles. What changes across images, videos, audio, 3D worlds, and communicative media is not the underlying AGS principle, but the artifact representation that the system must keep controllable. This cross-domain view also motivates the evaluation discussion in Sec. 6: AGS should be assessed not only by final artifact quality, but also by whether their planning, feedback use, state maintenance, and tool decisions are appropriate for the domain-specific object of control. Thus, agentic generation is best understood not as a modality-specific trick, but as a system-level paradigm for autonomous multimodal content creation.

6. Benchmarks, Evaluation, and Datasets

Evaluating Agentic Generative Systems (AGS) is more difficult than evaluating conventional one-shot generators, because success depends not only on the final artifact but also on how the system decomposes goals, invokes tools, reacts to failures, and preserves coherence across multiple steps. As a result, evaluation in this area is increasingly built from two complementary sources. One source comes from generation-centric resources and benchmark ecosystems that already stress long-horizon consistency, compositional instruction following, and multimodal coordination, including story visualization, complex image generation and editing, movie-oriented audio generation, audio-video generation, and presentation or scientific video authoring [13,128,145,153,295–297]. The other source consists of explicitly agent-centered benchmarks that evaluate workflow synthesis, GUI-based operation, collaborative artifact production, and promptable evaluation itself [98,100,125,133,298,299]. Taken together, these developments suggest that AGS evaluation is moving from static output scoring toward layered protocols that also measure process quality, interaction quality, and long-horizon robustness. Table 5 summarizes representative resources and the AGS capabilities they expose.

 doi:10.20944/preprints202607.1997.v1

Table 5. Representative datasets, benchmarks, and evaluation suites relevant to AGS. The resources are grouped by the agentic capability they make visible rather than by modality alone.

Resource family	Modality / setting	Evaluated capability	Why it matters for AGS
LongBench-T2I (Draw ALL Your Imagine) [153]; X-Planner edit-evaluation setting [128]	Image generation and editing	Complex instruction following; compositional constraints	Tests whether a system can convert underspecified or multi-constraint goals into structured generation plans rather than relying on a single prompt.
ViStoryBench and ISG-BENCH [295,300]	Story visualization and interleaved images	Cross-image consistency; subject and relation persistence	Exposes whether an AGS maintains global entities, relations, and narrative structure across multiple generated artifacts.
AudioAtlas and VABench [296,297]	Audio and audiovisual generation	Event timing; semantic alignment; audio-video synchronization	Moves evaluation beyond perceptual quality toward temporal coordination, which is central to agentic planning over multimodal timelines.
DreamFactory (Multi-Scene Videos) and AnimeShooter [214, 301]	Long-form video and animation	Multi-scene continuity; script-to-shot coherence	Stresses long-horizon robustness, character persistence, and the ability to preserve intent across many intermediate generation states.
MM-StoryAgent and MUSE task settings [113,302]	Multimodal storytelling	Narrative planning; cross-modal alignment	Evaluates whether agents can coordinate scripts, visual assets, narration, and temporal structure into coherent story-level outputs.
PresentAgent (PresentEval) and Paper2Video [13,145]	Presentation and scientific video authoring	Content fidelity; slide-video-audio coordination	Measures document-grounded generation, where systems must preserve source information while planning visual explanation and narration.
ComfyBench [125]	ComfyUI workflow design	Workflow synthesis; node selection; execution validity	Directly evaluates tool orchestration by asking whether an agent can build executable generative workflows rather than only produce images.
PSBench [98]	GUI-based image editing	Long-horizon software actions; step cost; task completion	Tests AGS as software-operating agents that must translate user goals into ordered interface actions under stateful constraints.
DECKBench [298]	Collaborative slide creation	Revision behavior; interaction utility; layout quality	Makes collaborative artifact production measurable, including whether an agent can revise structured media under user feedback.
Evaluation Agent, VideoGen-Eval, and Agentic-DRS [100,133,303]	Agentic, video, and design evaluation	Promptable evaluation; diagnostic feedback; multi-step judging; design review	Shows how evaluation itself can become an agentic process, producing explanations, critiques, and correction signals for downstream revision.
SceneCraft and WorldCraft task settings [16,17]	3D scene and world generation	Code-grounded execution; rendered verification; reusable skills	Highlights an important benchmark gap: many 3D AGS rely on task-specific demonstrations and rendered checks, but lack shared trace-level benchmark suites.

 doi:10.20944/preprints202607.1997.v1

6.1. Benchmark and Dataset Landscape

Existing AGS-related resources are heterogeneous. Some are conventional benchmark datasets with fixed input-output tasks, some are task suites for interactive systems, and others are evaluation frameworks that generate or judge test cases. This heterogeneity reflects the nature of AGS: the object being evaluated is not only an output artifact, but also a trajectory of planning, tool use, feedback, and revision. For this reason, we group datasets, benchmarks, and evaluation suites by the capability they stress rather than by publication venue or data format. When a source is not a standardized benchmark but still defines an evaluation setting or releases task data, we describe it as a resource or task setting rather than treating it as a benchmark.

The landscape in Table 5 reveals three trends. First, benchmark difficulty is shifting from local perceptual quality toward *structured constraint satisfaction*: systems are expected to preserve entities, layouts, timelines, and source-document semantics across many generated elements. Second, agent-centered benchmarks increasingly expose the *process* behind the artifact, including workflow validity, GUI action sequences, revision behavior, and step cost. Third, coverage remains uneven. Image, video, audio, and document-centric generation now have emerging public benchmarks, but 3D world building, cross-tool workflow reuse, multi-session memory, and personalized co-creation still lack standardized datasets with reusable trajectories.

Open benchmark gaps. Current resources rarely provide complete trajectory logs, tool-call traces, intermediate artifacts, feedback signals, or revision histories. This limits our ability to compare two AGS that reach similar final outputs through very different processes. Future benchmarks should therefore release not only input-output pairs, but also process annotations: subgoal decompositions, tool invocations, failed attempts, recovery steps, user interventions, and cost or latency statistics. Such trace-level data would make it possible to evaluate the central promise of AGS: not merely whether a system can generate an artifact, but whether it can generate through controllable and recoverable decision-making.

6.2. Evaluation Protocols for Agentic Generation

Building on this benchmark landscape, a useful evaluation protocol for AGS should distinguish at least three levels of competence: *artifact quality*, *trajectory quality*, and *interactive utility*. Artifact quality asks whether the final output satisfies semantic, aesthetic, and structural constraints. Trajectory quality asks whether the system reaches that output through an effective sequence of planning, tool use, verification, and correction. Interactive utility asks whether the system remains helpful under ambiguity, revision, and extended user interaction. This separation is important because two systems may obtain similarly strong final outputs while differing substantially in cost, robustness, recoverability, or editability [98,100,125,133,298].

Artifact-level evaluation remains necessary, but is no longer sufficient. Most AGS still inherit end metrics from their application domains, but these metrics are becoming more structured and multi-constraint. For image generation and editing, resources such as *Draw ALL Your Imagine/LongBench-T2I* and *X-Planner* focus on complex, compositional, or indirect instructions, while *ViStoryBench* and *ISG-BENCH* evaluate cross-image logical consistency, subject persistence, and interleaved multimodal coherence [128,153,295,300]. For audio and audiovisual generation, *AudioAtlas* and *VABench* extend evaluation beyond generic fidelity toward event-level timing, semantic alignment, and audio-video synchronization [296,297]. For presentation and document-centric generation, resources such as *PresentAgent/PresentEval*, *DECKBench*, and *Paper2Video* emphasize content fidelity, layout quality, and multimodal coordination rather than surface appearance alone [13,145,298]. These protocols show that final-output quality remains essential, but the target artifact itself is increasingly long-horizon, structured, and interaction-aware.

AGS require process-sensitive evaluation. A distinctive property of AGS is that the generation process is itself part of the capability being measured. Accordingly, several recent benchmarks evaluate whether an agent selects appropriate tools, constructs executable workflows, and repairs failed inter-

 doi:10.20944/preprints202607.1997.v1

mediate states. *ComfyBench* measures autonomous workflow design in ComfyUI-like environments, making node selection, parameter configuration, and execution validity central evaluation targets rather than hidden implementation details [125]. *PSBench* does something similar for GUI-based image editing, where the agent must convert user requests into long-horizon interface actions under step-cost and task-completion constraints [98]. *DECKBench* extends this idea to collaborative slide generation and editing, where revision behavior and coordination quality matter alongside the final presentation itself [298]. *Agentic-DRS* extends the same process-sensitive view to graphic-design review by coordinating static and dynamic reviewer agents over alignment, composition, aesthetics, and color choices, positioning actionable critique as an explicit evaluation product rather than an informal afterthought [303]. More broadly, agentic evaluation frameworks such as *Evaluation Agent* and *VideoGen-Eval* indicate a shift toward promptable, multi-step, and diagnosis-oriented evaluation, where intermediate reasoning traces, correction behavior, and failure explanations become part of the protocol rather than auxiliary analysis [100,133,304].

Long-horizon robustness should be evaluated explicitly. Because AGS operate over extended trajectories, a strong system should not only succeed once, but also preserve constraints, identities, and multimodal consistency across many intermediate states. This is particularly important in story visualization, long-video generation, and multimodal narrative creation. *ViStoryBench* makes this explicit by stressing narrative continuity and character consistency across visual sequences [295]. Long-form story and video generation resources such as *DreamFactory*'s Multi-Scene Videos dataset, *AnimeShooter*, *TalkVerse*, and the task settings introduced alongside *MM-StoryAgent* and *MUSE* similarly foreground cross-scene identity persistence, script-to-shot coherence, and long-range audiovisual alignment [113,214,301,302,305,306]. *VABench* pushes the same issue into general audiovisual generation by explicitly testing whether audio and video remain synchronized and semantically coherent over longer durations and across multiple evaluation dimensions [297]. These settings make clear that AGS should be judged not only by local quality, but also by whether they sustain global structure over time.

Human-centered evaluation remains indispensable. For many AGS outputs, especially editable and collaborative media, automatic metrics underdetermine practical usefulness. A system can be locally accurate yet still be difficult to steer, costly to revise, or poorly aligned with user intent in multi-turn settings. This is why human-centered evaluation remains central in domains such as image co-creation, slide authoring, and interactive editing. *PSBench* and *DECKBench* already make usability, editability, and instruction satisfaction more explicit parts of the benchmark protocol [98,298]. At the system level, works such as *AutoStudio*, *Talk2Image*, and *T2I-Copilot* further suggest that ambiguity handling, clarification, and user-aligned revision should be treated as first-class evaluation targets rather than secondary interface issues [24,102,172].

A practical protocol should combine outcome, process, robustness, and efficiency. Taken together, recent benchmarks suggest that AGS should be evaluated with a layered protocol rather than a single scalar score. At minimum, such a protocol should report: (i) final artifact success under task-specific criteria; (ii) process quality, such as correctness of tool selection, workflow validity, and revision behavior; (iii) robustness under ambiguity, perturbation, or long-horizon multi-turn generation; and (iv) efficiency, including the number of steps, edits, tool calls, or retries required to reach a satisfactory result [98,100,125,133,297,298]. This perspective is especially important for AGS because the central question is no longer only whether a model can generate, but whether it can *generate through decision-making*.

7. Future Directions and Open Challenges

The preceding sections show that Agentic Generative Systems (AGS) are moving from prompt-driven generation toward controllable, multi-step content creation. However, the same properties that make AGS powerful also make them difficult to build, evaluate, and deploy. Unlike one-shot generators, AGS expose a trajectory of decisions: goals are decomposed, tools are selected, intermediate artifacts are inspected, feedback is interpreted, and later actions depend on earlier failures.

 doi:10.20944/preprints202607.1997.v1

Future progress therefore requires not only stronger component models, but also better system-level abstractions for control, verification, memory, reproducibility, and human oversight.

7.1. Trace-Level Evaluation Beyond Final Artifacts

The most immediate challenge is evaluation. Sec. 6 summarized current resources and argued that final artifact quality is necessary but insufficient. The remaining gap is protocol-level: two systems may reach similar outputs through very different processes, yet most benchmarks still lack reusable traces for comparing those processes. This gap becomes larger as AGS move from L1 planning to L2 feedback loops and L3 long-horizon orchestration. A future benchmark should therefore record not only inputs and final outputs, but also the full process trace behind each result—the subgoal decompositions, tool calls, feedback, failed attempts, and recovery actions discussed in Sec. 6—so that systems reaching similar artifacts through different processes can be told apart.

Existing resources already point in this direction. *ComfyBench*, *PSBench*, and *DECKBench* make workflow construction, GUI action sequences, and collaborative revision measurable rather than hidden implementation details [98,125,298]. Agentic evaluation frameworks such as *Evaluation Agent* and *VideoGen-Eval* further suggest that evaluation itself can become diagnostic and multi-step [100,133]. The open problem is to make such process-sensitive evaluation reusable across domains. Trace-level benchmarks should specify what constitutes a valid plan, a useful revision, an unnecessary tool call, a recoverable failure, and an acceptable trade-off between quality and cost. Without this information, AGS evaluation will remain difficult to compare across papers, because the most important behavior of an agentic system may happen before the final artifact appears.

7.2. Reliable Feedback and Self-Correction

Closed-loop generation depends on feedback, but current feedback sources are often noisy, incomplete, or poorly calibrated. Many AGS use LLMs or VLMs as critics, judges, or constraint checkers. This enables flexible self-correction, but it also creates a new failure mode: the system may revise confidently in the wrong direction because the critic misdiagnoses the artifact. For example, feedback loops for counting, layout repair, video physicality, or editing consistency are only useful when the evaluator can detect the relevant error and produce an actionable correction signal [65,119, 201].

Future work should separate at least three feedback problems. First, *detection*: can the system reliably identify that a generated artifact violates the user's goal or a domain constraint? Second, *localization*: can it determine which part of the artifact, workflow, or plan caused the failure? Third, *repair*: can it choose a next action that improves the artifact without damaging already correct parts? Most current systems blur these steps into a single critique prompt. More robust AGS will likely require calibrated multimodal evaluators, domain-specific verifiers, uncertainty-aware stopping rules, and mechanisms for escalating ambiguous failures to humans or stronger tools. This is especially important for L3 systems, where an early misdiagnosis can propagate across many later steps.

7.3. Memory, State, and Long-Horizon Controllability

Memory is central to strong AGS, but memory remains under-specified in much of the literature. Many systems claim multi-turn interaction, persistent preference modeling, or long-horizon consistency, yet the maintained state is often informal: a text history, a list of generated assets, or a hidden prompt summary. For complex generation, this is not enough. Video production requires persistent character, shot, and narrative state; 3D world building requires editable geometry, assets, code, and constraints; document-to-media authoring requires source fidelity, rhetorical hierarchy, and layout history [14,16,17,25,29,105].

A useful research direction is to design explicit memory abstractions for AGS. Such memory should distinguish goals, plans, tool states, intermediate artifacts, user preferences, and failure histories. It should also support operations that matter for generation: retrieval, update, rollback, comparison between alternatives, and transfer to related tasks. This would make memory more than a passive

 doi:10.20944/preprints202607.1997.v1

context store. It would become part of the control state that allows an AGS to preserve identity, maintain editability, recover from failed branches, and personalize outputs without forgetting earlier constraints. The challenge is to make this state compact enough for efficient reasoning while still faithful enough to prevent long-horizon drift.

7.4. Tool Orchestration, Reproducibility, and Cost

Tool use gives AGS flexibility, but it also creates reproducibility and cost challenges. An AGS may call multiple generators, editors, retrievers, layout engines, simulators, GUI tools, and evaluators. Each tool may have its own version, randomness, latency, failure modes, licensing terms, and hidden assumptions. As systems such as *HuggingGPT*, *ComfyMind*, and *ComfySearch* demonstrate, tool orchestration can itself become the central generation problem [22,67,106]. However, a workflow that works in one environment may be difficult to reproduce in another if the tool library, execution parameters, or intermediate states are not recorded.

Future AGS should therefore treat tool orchestration as an auditable part of the system. At minimum, papers should report the tool set, versions, call order, retry policy, failure handling strategy, and compute or monetary cost. Stronger systems should reason about cost and reliability during planning, rather than optimizing only for final perceptual quality. This matters because agentic workflows can easily become expensive: repeated critique, regeneration, search, and rollback may improve quality while making the system too slow or costly for practical use. Cost-aware planning, tool-call caching, deterministic replay, and standardized workflow logs will be necessary if AGS are to move from impressive demonstrations to dependable production systems.

7.5. Learning to Plan, Search, and Recover

Many current AGS rely on hand-designed prompts or fixed orchestration templates. This is practical, but it limits generalization. As the action space grows from prompt rewriting to tool selection, GUI operation, code generation, and world-state editing, manually specifying all useful behaviors becomes infeasible. Future systems need better ways to learn planning and recovery policies from traces, demonstrations, user feedback, and failed attempts.

The key challenge is that AGS learning signals are structured and delayed. A poor tool choice may not become visible until several steps later; a locally good edit may break global consistency; a failed branch may still contain reusable partial progress. This suggests that AGS need learning methods that operate over trajectories rather than isolated examples. Useful directions include imitation from successful workflows, preference learning over alternative traces, search over tool trajectories, automatic curriculum construction from failures, and skill libraries that can be reused across tasks. For 3D and software-grounded generation, systems such as *SceneCraft* already point toward reusable skills and iterative improvement over code-grounded creation [16]. The broader opportunity is to make AGS improve not only their outputs, but also their control policies.

7.6. Native Agentic Generators and Hybrid Architectures

Most existing AGS implement agency as an external wrapper around generative models. An LLM or MLLM plans, invokes tools, reads outputs, and decides whether to revise. This design is modular and interpretable, but it also introduces overhead: the controller may have limited access to the generator's internal uncertainty, latent structure, or intermediate reasoning. Recent native agentic generation models such as *VisionCreator* and *VisionCreator-R1* suggest a different direction, where planning, reflection, and creation are partly internalized into the generator itself [43,44].

The likely future is not a simple replacement of external agents by native generators, but a hybrid architecture. External orchestration remains valuable for tool use, auditability, modular upgrades, and human control. Native agentic generation may be valuable for tighter integration between reasoning and synthesis, lower latency, and better use of model-internal representations. The open question is how to combine these advantages. Future systems may use native agentic generators for local planning and artifact-level self-revision, while external controllers manage long-horizon goals, tool

 doi:10.20944/preprints202607.1997.v1

ecosystems, memory, safety checks, and user interaction. This hybrid view also sharpens the boundary of AGS: a model is not agentic merely because it produces better outputs; it becomes part of AGS when generation is organized as an explicit, stateful decision process.

7.7. Human Oversight, Safety, and Authorship

As AGS become more autonomous, human oversight becomes more important rather than less. Creative generation often involves ambiguity, preference, cultural context, and rights-sensitive material. Document and presentation systems must preserve source meaning; image and video editing systems can alter identity or factual appearance; audio and avatar systems may affect voice, likeness, and perceived authenticity. Therefore, AGS should expose controllable checkpoints, editable intermediate artifacts, provenance records, and meaningful user override mechanisms.

Human-centered AGS should also clarify authorship and responsibility. When a system decomposes a goal, selects tools, synthesizes assets, revises outputs, and possibly ignores or reinterprets user instructions, the final artifact is shaped by both user intent and system-level decisions. This creates practical questions: which edits should be reversible, which intermediate artifacts should be retained, how should generated content be labeled, and how should unsafe or rights-conflicting requests be blocked or transformed? Interactive systems such as *AutoStudio*, *Talk2Image*, and *T2I-Copilot* indicate the importance of clarification and user-aligned revision in creative workflows [24,102,172]. Future AGS should extend this idea from interface convenience to governance: users should be able to understand, steer, audit, and constrain the generation trajectory itself.

Overall, the next stage of AGS research will be defined less by whether agents can call more tools than by whether they can control generation in a reliable, inspectable, and recoverable manner. Progress will require benchmarks with traces, feedback mechanisms that are calibrated and actionable, memory structures that preserve long-horizon state, orchestration protocols that are reproducible and cost-aware, learning methods that improve decision policies, and hybrid architectures that combine native generation with external control. These challenges are system-level by nature, which is precisely why AGS should be studied as a distinct paradigm rather than as a loose collection of prompting or tool-use techniques.

8. Conclusion

This survey presents Agentic Generative Systems (AGS) as a unified paradigm for autonomous multimodal content creation. Instead of treating generation as a single prompt-to-output mapping, AGS formulate content creation as a goal-driven, multi-step decision process that ranges from planned tool orchestration to closed-loop refinement, in which agents decompose objectives, coordinate heterogeneous generative tools, and, in higher-capability systems, observe intermediate artifacts and refine outputs under feedback. We formalized this paradigm through core components including goal representation, planning and control, tool orchestration, feedback, and memory, and introduced a three-level capability hierarchy that distinguishes linear-chain, reactive-loop, and goal-oriented orchestrated systems.

Building on this formalization, we organized existing work through a mechanism-centered taxonomy covering architecture, planning strategy, tool interaction, memory, and feedback loops. We then reviewed representative systems across image, video, audio, 3D, and document-centric generation, showing that the same L1–L3 progression recurs across modalities, with the underlying object of control changing from one domain to the next. We also summarized emerging evaluation protocols, emphasizing that AGS should be assessed not only by final artifact quality, but also by trajectory quality, robustness, interaction utility, and efficiency.

Despite rapid progress, AGS remain far from mature. As discussed in Sec. 7, the next stage of the field will depend on trace-level evaluation, reliable feedback, explicit memory, reproducible tool orchestration, learned recovery policies, hybrid native–external architectures, and stronger human oversight. We hope this survey provides a foundation for understanding AGS as a distinct system-level

 doi:10.20944/preprints202607.1997.v1

direction and helps guide future research toward multimodal generation systems that are not only more capable, but also more controllable, inspectable, and recoverable.

References

1. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the CVPR, 2022. arXiv:2112.10752. Stable Diffusion backbone.
2. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.; et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In Proceedings of the NeurIPS, 2022. arXiv:2205.11487. Google Imagen.
3. Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; Rombach, R. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis, 2024, [2307.01952]. ICLR 2024.
4. Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; et al. Improving Image Generation with Better Captions, 2023. OpenAI DALL-E 3 technical report.
5. Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; et al. Video Generation Models as World Simulators, 2024. OpenAI Sora technical report, Feb 2024.
6. Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; et al. Make-A-Video: Text-to-Video Generation without Text-Video Data. In Proceedings of the ICLR, 2023. arXiv:2209.14792. Meta Make-A-Video.
7. Kondratyuk, D.; Yu, L.; Gu, X.; Lezama, J.; Huang, J.; Hornung, R.; et al. VideoPoet: A Large Language Model for Zero-Shot Video Generation. In Proceedings of the ICML, 2024. arXiv:2312.14125. Google VideoPoet.
8. Borsos, Z.; Marinier, R.; Vincent, D.; Kharitonov, E.; Pietquin, O.; Sharifi, M.; et al. AudioLM: A Language Modeling Approach to Audio Generation. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **2023**. arXiv:2209.03143. Google AudioLM.
9. Copet, J.; Kreuk, F.; Gat, I.; Remez, T.; Kant, D.; Synnaeve, G.; Adi, Y.; Défossez, A. Simple and Controllable Music Generation. In Proceedings of the NeurIPS, 2023. arXiv:2306.05284. Meta MusicGen.
10. Poole, B.; Jain, A.; Barron, J.T.; Mildenhall, B. DreamFusion: Text-to-3D using 2D Diffusion. In Proceedings of the ICLR, 2023. arXiv:2209.14988. Google DreamFusion.
11. Liu, R.; Wu, R.; Van Hoorick, B.; Tokmakov, P.; Zakharov, S.; Vondrick, C. Zero-1-to-3: Zero-shot One Image to 3D Object. In Proceedings of the ICCV, 2023. arXiv:2303.11328. Columbia University.
12. Liu, J.; Yang, L.; Luo, H.; Wang, F.; Li, H.; Wang, M. Preacher: Paper-to-Video Agentic System. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 17129–17139.
13. Zhu, Z.; Lin, K.Q.; Shou, M.Z. Paper2Video: Automatic Video Generation from Scientific Papers. In Proceedings of the Workshop on Scaling Environments for Agents, 2025.
14. Hu, P.; Jiang, J.; Chen, J.; Han, M.; Liao, S.; Chang, X.; Liang, X. StoryAgent: Customized Storytelling Video Generation via Multi-Agent Collaboration, 2024, [arXiv:cs.CV/2411.04925].
15. Wang, Q.; Huang, Z.; Jia, R.; Debevec, P.; Yu, N. MAViS: A Multi-Agent Framework for Long-Sequence Video Storytelling. In Proceedings of the Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers); Demberg, V.; Inui, K.; Marquez, L., Eds., Rabat, Morocco, 2026; pp. 2273–2295. https://doi.org/10.18653/v1/2026.eacl-long.101.
16. Hu, Z.; Iscen, A.; Jain, A.; Kipf, T.; Yue, Y.; Ross, D.A.; Schmid, C.; Fathi, A. SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning; Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; Berkenkamp, F., Eds. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 19252–19282.
17. Liu, X.; Tang, C.K.; Tai, Y.W. WorldCraft: Photo-Realistic 3D World Creation and Customization via LLM Agents, 2025, [arXiv:cs.CV/2502.15601].
18. Wang, Z.; Li, A.; Li, Z.; Liu, X. GenArtist: Multimodal LLM as an Agent for Unified Image Generation and Editing. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 128374–128395. https://doi.org/10.52202/079017-4077.
19. Ye, R.; Zhang, J.; Liu, Z.; Zhu, Z.; Yang, S.; Li, L.; Fu, T.; Dernoncourt, F.; Zhao, Y.; Zhu, J.; et al. Agent Banana: High-Fidelity Image Editing with Agentic Thinking and Tooling, 2026, [arXiv:cs.CV/2602.09084].

 doi:10.20944/preprints202607.1997.v1

20. Yang, Z.; Wang, J.; Li, L.; Lin, K.; Lin, C.C.; Liu, Z.; Wang, L. Idea2Img: Iterative Self-refinement with GPT-4V for Automatic Image Design and Generation. In Proceedings of the Computer Vision – ECCV 2024; Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; Varol, G., Eds., Cham, 2025; pp. 167–184.
21. Yang, L.; Yu, Z.; Meng, C.; Xu, M.; Ermon, S.; Cui, B. Mastering Text-to-Image Diffusion: Recaptioning, Planning, and Generating with Multimodal LLMs. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning; Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; Berkenkamp, F., Eds. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 56704–56721.
22. Guo, L.; Xu, X.; Wang, L.; Lin, J.; Zhou, J.; Zhang, Z.; Su, B.; Chen, Y. ComfyMind: Toward General-Purpose Generation via Tree-Based Planning and Reactive Feedback. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 45128–45164.
23. Liu, X.; Zhu, Z.; Liu, H.; Yuan, Y.; Huang, Q.; Cui, M.; Liang, J.; Cao, Y.; Kong, Q.; Plumbley, M.D.; et al. WavJourney: Compositional Audio Creation With Large Language Models. *IEEE Transactions on Audio, Speech and Language Processing* **2025**, *33*, 2830–2844. https://doi.org/10.1109/taslpro.2025.3574867.
24. Ma, S.; Guo, Y.; Su, J.; Huang, Q.; Zhou, Z.; Wang, Y. Talk2Image: A Multi-Agent System for Multi-Turn Image Generation and Editing. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 32437–32445. https://doi.org/10.1609/aaai.v40i38.40519.
25. Wu, W.; Zhu, Z.; Shou, M.Z. Automated Movie Generation via Multi-Agent CoT Planning, 2025, [arXiv:cs.CV/2503.07314].
26. Jiang, Y.; Chen, Z.; Ju, Z.; Li, C.; Dou, W.; Zhu, J. FreeAudio: Training-Free Timing Planning for Controllable Long-Form Text-to-Audio Generation. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025; Gurrin, C.; Schoeffmann, K.; Zhang, M.; Rossetto, L.; Rudinac, S.; Dang-Nguyen, D.; Cheng, W.; Chen, P.; Benois-Pineau, J., Eds. ACM, 2025, pp. 9871–9880. https://doi.org/10.1145/3746027.3755170.
27. Rong, Y.; Yang, S.; Li, C.; Yu, D.; Liu, L. Dopamine Audiobook: A Training-free MLLM Agent for Emotional and Immersive Audiobook Generation, 2025, [arXiv:cs.SD/2504.11002].
28. Zhang, S.; Zhou, M.; Wang, Y.; Luo, C.; Wang, R.; Li, Y.; Zhang, Z.; Peng, J. CityX: Controllable Procedural Content Generation for Unbounded 3D Cities, 2024, [arXiv:cs.CV/2407.17572].
29. Liang, X.; Zhang, X.; Xu, Y.; Sun, S.; You, C. SlideGen: Collaborative Multimodal Agents for Scientific Slide Generation, 2025, [arXiv:cs.AI/2512.04529].
30. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.H. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Computing Surveys* **2024**, *56*, 1–39. arXiv:2209.00796.
31. Zhang, C.; Zhang, C.; Zhang, M.; Kweon, I.S.; Kim, J. Text-to-Image Diffusion Models in Generative AI: A Survey, 2023, [2303.07909]. Comprehensive T2I diffusion survey.
32. Wang, Y.; Liu, X.; Pang, W.; Ma, L.; Yuan, S.; Debevec, P.; Yu, N. Survey of Video Diffusion Models: Foundations, Implementations, and Applications, 2025, [2504.16081]. TMLR 2025. Comprehensive video diffusion survey.
33. Li, X.; Zhang, Q.; Kang, D.; Cheng, W.; Gao, Y.; Zhang, J.; Liang, Z.; Liao, J.; Cao, Y.P.; Shan, Y. Advances in 3D Generation: A Survey, 2024, [2401.17807]. Comprehensive 3D generation survey covering NeRF/Gaussian/diffusion.
34. Cao, Y.; Li, S.; Liu, Y.; Yan, Z.; Dai, Y.; Yu, P.S.; Sun, L. A Comprehensive Survey of AI-Generated Content (AIGC): A History of Generative AI from GAN to ChatGPT, 2023, [2303.04226]. Widely-cited AIGC survey.
35. Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; et al. The Rise and Potential of Large Language Model Based Agents: A Survey, 2023, [2309.07864]. Fudan. Highly-cited LLM agent survey with brain/perception/action framework.
36. Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; et al. A Survey on Large Language Model based Autonomous Agents, 2023, [2308.11432]. Renmin Univ. LLM autonomous agent survey (profile/memory/planning/action).
37. Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large Language Model based Multi-Agents: A Survey of Progress and Challenges, 2024, [2402.01680]. Multi-agent LLM survey.
38. Yao, H.; Zhang, R.; Huang, J.; Zhang, J.; Wang, Y.; Fang, B.; Zhu, R.; Jing, Y.; Liu, S.; Li, G.; et al. A Survey on Agentic Multimodal Large Language Models, 2025, [2510.10991]. NTU/CUHK-SZ. Focus on agentic MLLMs as models (reasoning/reflection/memory via RL). Not content-creation focused.

 doi:10.20944/preprints202607.1997.v1

39. Schneider, J. Generative to Agentic AI: Survey, Conceptualization, and Challenges, 2025, [2504.18875]. Univ. Liechtenstein. Domain-agnostic conceptual survey on GenAI to Agentic AI transition.
40. Xie, J.; Chen, Z.; Zhang, R.; Wan, X.; Li, G. Large Multimodal Agents: A Survey, 2024, [2402.15116]. CUHK-SZ/SYSU. Broad LMA survey (Type I-IV by memory/training). Pre-Sora.
41. Durante, Z.; Huang, Q.; Wake, N.; Gong, R.; Park, J.S.; Sarkar, B.; Taori, R.; Noda, Y.; Terzopoulos, D.; Choi, Y.; et al. Agent AI: Surveying the Horizons of Multimodal Interaction, 2024, [2401.03568]. Microsoft/Stanford. Broad Agent AI survey covering embodied, multimodal, generative.
42. Lei, W.; Rong, Y.; Wang, J.; Xie, T.; Huang, G.; Liu, L. A Comprehensive Survey on Multi-Agent Systems for Audio-Visual Generation and Understanding, 2025. TechRxiv, DOI:10.36227/techrxiv.176583352.26143196/v1. HKUST-GZ. MAS for AV generation+understanding, modality-centered taxonomy, 4 coordination patterns.
43. Lai, J.; Lu, Z.; He, J.; Quan, R.; Zhao, W.; Yang, Q.; Chen, Q.; Lin, Q.; Li, C.; Gao, T.; et al. VisionCreator: A Native Visual-Generation Agentic Model with Understanding, Thinking, Planning and Creation, 2026. arXiv.
44. Lai, J.; Zhao, W.; Lu, Z.; Zhang, H.; Yang, Q.; Quan, R.; Li, Z.; Shao, S.; Guo, S.; Lu, Q. VisionCreator-R1: A Reflection-Enhanced Native Visual-Generation Agentic Model, 2026, [arXiv:cs.CV/2603.08812].
45. Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-Shot Text-to-Image Generation. In Proceedings of the ICML, 2021. arXiv:2102.12092. OpenAI DALL-E 1.
46. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the CVPR, 2019. arXiv:1812.04948. NVIDIA StyleGAN.
47. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the CVPR, 2020. arXiv:1912.04958. NVIDIA StyleGAN2.
48. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the NeurIPS, 2020. arXiv:2006.11239. Foundational DDPM paper.
49. Zhang, L.; Rao, A.; Agrawala, M. Adding Conditional Control to Text-to-Image Diffusion Models, 2023, [2302.05543]. ICCV 2023. ControlNet.
50. Ye, H.; Zhang, J.; Liu, S.; Han, X.; Yang, W. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models, 2023, [2308.06721]. IP-Adapter for image prompt conditioning.
51. Brooks, T.; Holynski, A.; Efros, A.A. InstructPix2Pix: Learning to Follow Image Editing Instructions. In Proceedings of the CVPR, 2023. arXiv:2211.09800.
52. Pan, X.; Tewari, A.; Leimkühler, T.; Liu, L.; Meka, A.; Theobalt, C. Drag Your GAN: Interactive Point-based Manipulation on the Generative Image Manifold, 2023, [2305.10973]. SIGGRAPH 2023. DragGAN.
53. Kawar, B.; Zada, S.; Lang, O.; Tov, O.; Chang, H.; Dekel, T.; Mosseri, I.; Irani, M. Imagic: Text-Based Real Image Editing with Diffusion Models, 2023, [2210.09276]. CVPR 2023.
54. Hong, W.; Ding, M.; Zheng, W.; Liu, X.; Tang, J. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers, 2023, [2205.15868]. ICLR 2023. Tsinghua CogVideo.
55. Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorber, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; et al. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets, 2023, [2311.15127]. Stability AI. SVD.
56. Guo, Y.; Yang, C.; Rao, A.; Liang, Z.; Wang, Y.; Qiao, Y.; Agrawala, M.; Lin, D.; Dai, B. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning, 2024, [2307.04725]. ICLR 2024.
57. Wang, C.; Chen, S.; Wu, Y.; Zhang, Z.; Zhou, L.; Liu, S.; Chen, Z.; Liu, Y.; Wang, H.; Li, J.; et al. VALL-E: Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers, 2023, [2301.02111]. Microsoft VALL-E.
58. Agostinelli, A.; Denk, T.I.; Borsos, Z.; Engel, J.; Verzetti, M.; Caillon, A.; Huang, Q.; Jansen, A.; Roberts, A.; Tagliasacchi, M.; et al. MusicLM: Generating Music From Text, 2023, [2301.11325]. Google MusicLM.
59. Kreuk, F.; Synnaeve, G.; Polyak, A.; Singer, U.; Défossez, A.; Copet, J.; Parikh, D.; Taigman, Y.; Adi, Y. AudioGen: Textually Guided Audio Generation, 2023, [2209.15352]. ICLR 2023. Meta AudioGen.
60. Lin, C.H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.Y.; Lin, T.Y. Magic3D: High-Resolution Text-to-3D Content Creation. In Proceedings of the CVPR, 2023. arXiv:2211.10440. NVIDIA Magic3D.
61. Nichol, A.; Jun, H.; Dhariwal, P.; Mishkin, P.; Chen, M. Point-E: A System for Generating 3D Point Clouds from Complex Prompts, 2022, [2212.08751]. OpenAI Point-E.
62. Kerbl, B.; Kopanas, G.; Leimkühler, T.; Drettakis, G. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. In Proceedings of the SIGGRAPH, 2023. arXiv:2308.14737.

 doi:10.20944/preprints202607.1997.v1

63. Yi, T.; Fang, J.; Wang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; Wang, X. GaussianDreamer: Fast Generation from Text to 3D Gaussians by Bridging 2D and 3D Diffusion Models, 2024, [2310.08529]. CVPR 2024.
64. Chen, H.; Xu, X.; Li, W.; Ren, J.; Ye, T.; Liu, S.; Chen, Y.C.; Zhu, L.; Wang, X. POSTA: A Go-to Framework for Customized Artistic Poster Generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 28694–28704.
65. Mondal, A.; Banerjee, A.; Nag, S.; Llados, J.; Zhu, X.; Dutta, A. CountLoop: Training-Free High-Instance Image Generation via Iterative Agent Guidance, 2025, [arXiv:cs.CV/2508.16644].
66. Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; Duan, N. Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models, 2023, [arXiv:cs.CV/2303.04671].
67. Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; Zhuang, Y. HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds. Curran Associates, Inc., 2023, Vol. 36, pp. 38154–38180.
68. Liu, Z.; He, Y.; Wang, W.; Wang, W.; Wang, Y.; Chen, S.; Zhang, Q.; Lai, Z.; Yang, Y.; Li, Q.; et al. InternGPT: Solving Vision-Centric Tasks by Interacting with ChatGPT Beyond Language, 2023, [arXiv:cs.CV/2305.05662].
69. Feng, W.; Zhu, W.; Fu, T.J.; Jampani, V.; Akula, A.; He, X.; Basu, S.; Wang, X.E.; Wang, W.Y. LayoutGPT: Compositional Visual Planning and Generation with Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds. Curran Associates, Inc., 2023, Vol. 36, pp. 18225–18250.
70. Lian, L.; Li, B.; Yala, A.; Darrell, T. LLM-grounded Diffusion: Enhancing Prompt Understanding of Text-to-Image Diffusion Models with Large Language Models, 2024, [2305.13655]. TMLR 2024.
71. Li, Y.; Liu, H.; Wu, Q.; Mu, F.; Yang, J.; Gao, J.; Li, C.; Lee, Y.J. GLIGEN: Open-Set Grounded Text-to-Image Generation, 2023, [2301.07093]. CVPR 2023.
72. Gal, R.; Haviv, A.; Alaluf, Y.; Bermano, A.H.; Cohen-Or, D.; Chechik, G. ComfyGen: Prompt-Adaptive Workflows for Text-to-Image Generation. In Proceedings of the ICLR 2025 Workshop on Modularity for Collaborative, Decentralized, and Continual Deep Learning, 2025.
73. Huang, O.; Ma, Y.; Zhao, Z.; Wu, M.; Ji, J.; Zhang, R.; Hu, Z.; Sun, X.; Ji, R. ComfyGPT: A Self-Optimizing Multi-Agent System for Comprehensive ComfyUI Workflow Generation, 2025, [arXiv:cs.MA/2503.17671].
74. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.R.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.
75. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools, 2023, [2302.04761]. NeurIPS 2023. Meta Toolformer.
76. Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An Open-Ended Embodied Agent with Large Language Models, 2023, [2305.16291]. NeurIPS 2023. NVIDIA Voyager.
77. Yang, J.; Jimenez, C.E.; Wettig, A.; Liber, K.; Narasimhan, S.Y.; Karthik. SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering, 2024, [2405.15793]. Princeton SWE-agent.
78. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2022, [2201.11903]. NeurIPS 2022. Chain-of-Thought.
79. Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; Narasimhan, K. Tree of Thoughts: Deliberate Problem Solving with Large Language Models, 2023, [2305.10601]. NeurIPS 2023. Tree-of-Thought.
80. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning, 2024, [2303.11366]. NeurIPS 2023. Reflexion.
81. Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. Self-Refine: Iterative Refinement with Self-Feedback, 2023, [2303.17651]. NeurIPS 2023. Self-Refine.
82. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. Tool Learning with Foundation Models, 2024, [2304.08354]. Tsinghua. Foundational tool-learning survey.
83. Park, J.S.; O'Brien, J.C.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative Agents: Interactive Simulacra of Human Behavior, 2023, [2304.03442]. UIST 2023. Stanford Generative Agents (Smallville).

 doi:10.20944/preprints202607.1997.v1

84. Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S.G.; Stoica, I.; Gonzalez, J.E. MemGPT: Towards LLMs as Operating Systems, 2024, [2310.08560]. UC Berkeley. MemGPT.
85. Ma, Y.; Feng, K.; Hu, Z.; Wang, X.; Wang, Y.; Zheng, M.; Wang, B.; Wang, Q.; He, X.; Wang, H.; et al. Controllable Video Generation: A Survey, 2025, [arXiv:cs.GR/2507.16869].
86. Xue, H.; Luo, X.; Hu, Z.; Zhang, X.; Xiang, X.; Dai, Y.; Liu, J.; Zhang, Z.; Li, M.; Yang, J.; et al. Human Motion Video Generation: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 10709–10730. https://doi.org/10.1109/TPAMI.2025.3594034.
87. Wen, B.; Xie, H.; Chen, Z.; Hong, F.; Liu, Z. 3D Scene Generation: A Survey, 2025, [arXiv:cs.CV/2505.05474].
88. Tang, X.; Li, R.; Fan, X. Recent Advances in 3D Object and Scene Generation: A Survey, 2025, [arXiv:cs.GR/2504.11734].
89. Wang, Z.; Bao, C.; Zhuo, L.; Han, J.; Yue, Y.; Tang, Y.; Huang, V.S.J.; Liao, Y. A Survey on Vision-to-Music Generation: Methods, Datasets, Evaluation, and Challenges. In Proceedings of the Proceedings of the 26th International Society for Music Information Retrieval Conference. ISMIR, 2025, pp. 223–234. https://doi.org/10.5281/zenodo.17811355.
90. Lin, Y.C.; Chen, K.C.; Li, Z.Y.; Wu, T.H.; Wu, T.H.; Chen, K.Y.; Lee, H.y.; Chen, Y.N. Creativity in LLM-based Multi-Agent Systems: A Survey. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 27584–27607. https://doi.org/10.18653/v1/2025.emnlp-main.1403.
91. He, Y.; Liu, Z.; Chen, J.; Tian, Z.; Liu, H.; Chi, X.; Liu, R.; Yuan, R.; Xing, Y.; Wang, W.; et al. LLMs Meet Multimodal Generation and Editing: A Survey, 2024, [arXiv:cs.AI/2405.19334].
92. Tu, R.; Sun, W.; Jin, Z.; Liao, J.; Huang, J.; Tao, D. SPAgent: Adaptive Task Decomposition and Model Selection for General Video Generation and Editing. *IEEE Trans. Image Process.* **2026**, *35*, 3085–3098. https://doi.org/10.1109/TIP.2026.3673949.
93. Jia, C.; Xia, C.; Dang, Z.; Wu, W.; Qian, H.; Luo, M. ChatGen: Automatic Text-to-Image Generation From FreeStyle Chatting. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 13284–13293.
94. Shi, H.; Li, Y.; Chen, X.; Wang, L.; Hu, B.; Zhang, M. AniMaker: Multi-Agent Animated Storytelling with MCTS-Driven Clip Generation. In Proceedings of the Proceedings of the SIGGRAPH Asia 2025 Conference Papers, SA Conference Papers 2025, Hong Kong, December 15-18, 2025; Komura, T.; Wimmer, M.; Fu, H., Eds. ACM, 2025, pp. 85:1–85:11. https://doi.org/10.1145/3757377.3764009.
95. Zhu, K.; Gu, J.; You, Z.; Qiao, Y.; Dong, C. An Intelligent Agentic System for Complex Image Restoration Problems. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
96. Xu, Z.; Wang, Y.; Yang, X.; Wang, L.; Luo, W.; Zhang, K.; Hu, B.; Zhang, M. ComfyUI-R1: Exploring Reasoning Models for Workflow Generation, 2025, [arXiv:cs.CL/2506.09790].
97. Lv, J.; Huang, Y.; Yan, M.; Huang, J.; Liu, J.; Liu, Y.; Wen, Y.; Chen, X.; Chen, S. GPT4Motion: Scripting Physical Motions in Text-to-Video Generation via Blender-Oriented GPT Planning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2024, pp. 1430–1440.
98. Zhang, Y.; Cheng, Z.; Zhao, Z.; Li, Z.; Liu, B.; Liu, Q.; Cai, J.; Chen, X.; Tu, Z.; Chu, D.; et al. PSBench: Editing Image via GUI Agents in Photoshop, 2026.
99. Yin, F.; Liu, S.; Han, Y.; Wang, Z.; Xing, P.; Wang, R.; Cheng, W.; Wang, Y.; Li, A.; Yin, Z.; et al. ReasonEdit: Towards Reasoning-Enhanced Image Editing Models, 2025, [arXiv:cs.CV/2511.22625].
100. Zhang, F.; Tian, S.; Huang, Z.; Qiao, Y.; Liu, Z. Evaluation Agent: Efficient and Promptable Evaluation Framework for Visual Generative Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 7561–7582. https://doi.org/10.18653/v1/2025.acl-long.374.
101. Huang, Z.; Zhuang, S.; Fu, C.; Yang, B.; Zhang, Y.; Sun, C.; Zhang, Z.; Wang, Y.; Li, C.; Zha, Z.J. WeGen: A Unified Model for Interactive Multimodal Generation as We Chat. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 23679–23689.
102. Cheng, J.; Lu, X.; Li, H.; Zai, K.L.; Yin, B.; Cheng, Y.; Yan, Y.; Liang, X. AutoStudio: Crafting Consistent Subjects in Multi-turn Interactive Image Generation, 2024. arXiv.
103. Liu, Z.; Yu, Y.; Ouyang, H.; Wang, Q.; Cheng, K.L.; Wang, W.; Liu, Z.; Chen, Q.; Shen, Y. MagicQuill: An Intelligent Interactive Image Editing System. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 13072–13082.

 doi:10.20944/preprints202607.1997.v1

104. Gupta, A.; Velaga, N.; Nguyen, D.; Zhou, T. CoSTA∗: Cost-Sensitive Toolpath Agent for Multi-turn Image Editing, 2025, [arXiv:cs.CV/2503.10613].
105. Zhou, J.; Du, Y.; Xu, X.; Wang, L.; Zhuang, Z.; Zhang, Y.; Li, S.; Hu, X.; Su, B.; cong Chen, Y. VideoMemory: Toward Consistent Video Generation via Memory Integration, 2026, [arXiv:cs.CV/2601.03655].
106. Su, J.; Lan, Q.; Wang, Z.; Xia, Y.; Wen, H.; Duan, Y.; Xiao, X.; Shi, T.; Jingsong, Y.; He, L. ComfySearch: Autonomous Exploration and Reasoning for ComfyUI Workflows, 2026, [arXiv:cs.AI/2601.04060].
107. Ye-Bin, M.; Miles, R.; Oh, T.H.; Elezi, I.; Deng, J. RetouchLLM: Training-free Code-based Image Retouching with Vision Language Models, 2025, [arXiv:cs.CV/2510.08054].
108. Xu, X.; Xu, X.; Chen, S.; Chen, H.; Zhang, F.; Chen, Y.C. PreGenie: An Agentic Framework for High-quality Visual Presentation Generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 3045–3063. https://doi.org/10.18653/v1/2025.findings-emnlp.165.
109. Zeng, Z.; Hua, H.; Luo, J. MIRA: Multimodal Iterative Reasoning Agent for Image Editing, 2025. arXiv.
110. Yue, Z.; Zhuang, S.; Li, K.; Ding, Y.; Wang, Y. V-Stylist: Video Stylization via Collaboration and Reflection of MLLM Agents. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 3195–3205.
111. Zhou, Y.; Cao, J.; Wen, F.; Zhang, Z.; Zhou, Y.; Shi, Y.; Liu, X.; Timofte, R.; Gool, L.V.; Zhai, G. Q-Agent: Quality-Driven Chain-of-Thought Image Restoration Agent through Robust Multimodal Large Language Model, 2025, [arXiv:eess.IV/2504.07148].
112. Wang, H.; Shimose, Y.; Takamatsu, S. BannerAgency: Advertising Banner Design with Multimodal LLM Agents. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 4304–4329. https://doi.org/10.18653/v1/2025.emnlp-main.214.
113. Xu, X.; Mei, J.; Li, C.; Wu, Y.; Yan, M.; Lai, S.; Zhang, J.; Wu, M. MM-StoryAgent: Immersive Narrated Storybook Video Generation with a Multi-Agent Paradigm across Text, Image and Audio, 2025, [arXiv:cs.CL/2503.05242].
114. Zhang, L.; Xu, B.; Yang, S.; Yin, M.; Liu, J.; Xu, C.; Wang, S.; Wu, Y.; Hong, Y.; Zhang, Z.; et al. AniME: Adaptive Multi-Agent Planning for Long Animation Generation. In Proceedings of the Proceedings of the SIGGRAPH Asia 2025 Posters, SA Posters 2025, Hong Kong, SAR, China, December 15-18, 2025; Komura, T.; Noh, J., Eds. ACM, 2025, pp. 3:1–3:3. https://doi.org/10.1145/3757374.3771455.
115. Yang, B.; Guo, R.; Fan, J.; Cheng, C.; Liu, G. M3: High-fidelity Text-to-Image Generation via Multi-Modal, Multi-Agent and Multi-Round Visual Reasoning, 2026, [arXiv:cs.CV/2602.06166].
116. Kuang, Z.; Lin, R.; Zhao, L.; Wetzstein, G.; Xie, S.; Woo, S. VULCAN: Tool-Augmented Multi Agents for Iterative 3D Object Arrangement, 2025. Project page.
117. Wang, L.; Xing, X.; Cheng, Y.; Zhao, Z.; Li, D.; Hang, T.; Tao, J.; Wang, Q.; Li, R.; Chen, C.; et al. PromptEnhancer: A Simple Approach to Enhance Text-to-Image Models via Chain-of-Thought Prompt Rewriting, 2025. arXiv.
118. Lian, L.; Shi, B.; Yala, A.; Darrell, T.; Li, B. LLM-grounded Video Diffusion Models. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
119. Wu, T.H.; Lian, L.; Gonzalez, J.E.; Li, B.; Darrell, T. Self-correcting LLM-controlled Diffusion Models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 6327–6336.
120. Li, H.; Zhang, M.; Zheng, D.; Guo, Z.; Jia, Y.; Feng, K.; Yu, H.; Liu, Y.; Feng, Y.; Pei, P.; et al. EditThinker: Unlocking Iterative Reasoning for Any Image Editor, 2025, [arXiv:cs.CV/2512.05965].
121. Mu, C.; He, X.; Yang, Q.; Chen, W.; Yao, J.; Liu, H.; Yi, Z.; Zhao, B.; Chen, X.; Ma, R.; et al. The Script is All You Need: An Agentic Framework for Long-Horizon Dialogue-to-Cinematic Video Generation, 2026, [arXiv:cs.CV/2601.17737].
122. Chen, Y.; Lin, K.Q.; Shou, M.Z. Code2Video: A Code-centric Paradigm for Educational Video Generation. In Proceedings of the NeurIPS 2025 Fourth Workshop on Deep Learning for Code, 2025.
123. Yao, M.; You, Z.; Tam, K.M.; Wang, M.; Xue, T. PhotoAgent: Agentic Photo Editing with Exploratory Visual Aesthetic Planning, 2026, [arXiv:cs.CV/2602.22809].
124. Lin, Y.; Lin, Z.; Chen, H.; Pan, P.; Li, C.; Chen, S.; Wen, K.; Jin, Y.; Li, W.; Ding, X. JarvisIR: Elevating Autonomous Driving Perception with Intelligent Image Restoration. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 22369–22380.

 doi:10.20944/preprints202607.1997.v1

125. Xue, X.; Lu, Z.; Huang, D.; Wang, Z.; Ouyang, W.; Bai, L. ComfyBench: Benchmarking LLM-based Agents in ComfyUI for Autonomously Designing Collaborative AI Systems. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 24614–24624.
126. Wu, R.; Su, W.; Liao, J. Chat2SVG: Vector Graphics Generation with Large Language Models and Image Diffusion Models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 23690–23700.
127. Huang, I.; Yang, G.; Guibas, L. BlenderAlchemy: Editing 3D Graphics with Vision-Language Models. In Proceedings of the Computer Vision – ECCV 2024; Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; Varol, G., Eds., Cham, 2025; pp. 297–314.
128. Yeh, C.H.; Wang, Y.; Zhao, N.; Zhang, R.; Li, Y.; Ma, Y.; Singh, K.K. Beyond Simple Edits: X-Planner for Complex Instruction-Based Image Editing. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 11991–11999. https://doi.org/10.1609/aaai.v40i14.38187.
129. Zeng, Q.; Cai, K.; Chen, R.; Lv, Q.; Wang, K. CoAgent: Collaborative Planning and Consistency Agent for Coherent Video Generation, 2025, [arXiv:cs.CV/2512.22536].
130. He, H.; Yang, H.; Tuo, Z.; Zhou, Y.; Wang, Q.; Zhang, Y.; Liu, Z.; Huang, W.; Chao, H.; Yin, J. DreamStory: Open-Domain Story Visualization by LLM-Guided Multi-Subject Consistent Diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 11874–11891. https://doi.org/10.1109/TPAMI.2025.3600149.
131. Wang, J.; Du, Z.; Zhao, Y.; Yuan, B.; Wang, K.; Liang, J.; Zhao, Y.; Lu, Y.; Li, G.; Gao, J.; et al. AesopAgent: Agent-driven Evolutionary System on Story-to-Video Production, 2024, [arXiv:cs.CV/2403.07952].
132. Guo, Z.; Zhang, R.; Li, H.; Zhang, M.; Chen, X.; Wang, S.; Feng, Y.; Pei, P.; Heng, P.A. Thinking-while-Generating: Interleaving Textual Reasoning throughout Visual Generation, 2025. arXiv.
133. Yang, Y.; Fan, K.; Sun, S.; Li, H.; Zeng, A.; Han, F.; Zhai, W.; Liu, W.; Cao, Y.; Zha, Z.J. VideoGen-Eval: Agent-based System for Video Generation Evaluation, 2025, [arXiv:cs.CV/2503.23452].
134. Liu, Z.; Bao, X.; Li, P.; Zhou, J.; Liao, Z.; He, Y.; Jiang, K.; Xie, C.W.; Zheng, Y.; Xie, H. ShowTable: Unlocking Creative Table Visualization with Collaborative Reflection and Refinement, 2026, [arXiv:cs.CV/2512.13303].
135. Soni, A.; Venkataraman, S.; Chandra, A.; Fischmeister, S.; Liang, P.; Dai, B.; Yang, S. VideoAgent: Self-Improving Video Generation for Embodied Planning. In Proceedings of the NeurIPS 2025 Workshop on Bridging Language, Agent, and World Models for Reasoning and Planning, 2025.
136. He, X.; Yang, T.; Cao, K.; Wu, R.; Meng, C.; Zhang, Y.; Kang, Z.; Wei, X.; Chen, Q. Active Intelligence in Video Avatars via Closed-loop World Modeling, 2025. arXiv.
137. Wu, M.; Wang, L.; Zhao, P.; Yang, F.; Zhang, J.; Liu, J.; Zhan, Y.; Han, W.; Sun, H.; Ji, J.; et al. RePrompt: Reasoning-Augmented Reprompting for Text-to-Image Generation via Reinforcement Learning. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
138. Zhao, Y.; Ye, Y.; Liu, X.; Shieh, M.Q.; Bui, T. ImageEdit-R1: Boosting Multi-Agent Image Editing via Reinforcement Learning, 2026, [arXiv:cs.CV/2603.08059].
139. Wu, M.; Yang, J.; Jiang, J.; Li, M.; Yan, K.; Yu, H.; Zhang, M.; Zhai, C.; Nahrstedt, K. VTool-R1: VLMs Learn to Think with Images via Reinforcement Learning on Multimodal Tool Use. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
140. Zuo, Y.; Zheng, Q.; Wu, M.; Jiang, X.; Li, R.; Wang, J.; Zhang, Y.; Mai, G.; Wang, L.; Zou, J.; et al. 4KAgent: Agentic Any Image to 4K Super-Resolution. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 164106–164175.
141. Huang, K.; Huang, Y.; Ning, X.; Lin, Z.; Wang, Y.; Liu, X. GENMAC: Compositional Text-to-Video Generation with Multi-Agent Collaboration. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 5049–5057. https://doi.org/10.1609/aaai.v40i7.37418.
142. Xiao, Y.; He, L.; Guo, H.; Xie, F.L.; Lee, T. PodAgent: A Comprehensive Framework for Podcast Generation. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025; Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 23923–23937. https://doi.org/10.18653/v1/2025.findings-acl.1226.
143. Wang, Z.; Tang, C.K.; Tai, Y.W. ReelWave: Multi-Agentic Movie Sound Generation through Multimodal LLM Conversation, 2025, [arXiv:cs.SD/2503.07217].
144. Zhou, M.; Wang, Y.; Hou, J.; Zhang, S.; Li, Y.; Luo, C.; Peng, J.; Zhang, Z. SceneX: Procedural Controllable Large-Scale Scene Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* **2025**, *39*, 10806–10814. https://doi.org/10.1609/aaai.v39i10.33174.

 doi:10.20944/preprints202607.1997.v1

145. Shi, J.; Zhang, Z.; Wu, B.; Liang, Y.; Fang, M.; Chen, L.; Zhao, Y. PresentAgent: Multimodal Agent for Presentation Video Generation. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; Habernal, I.; Schulam, P.; Tiedemann, J., Eds., Suzhou, China, 2025; pp. 760–773. https://doi.org/10.18653/v1/2025.emnlp-demos.58.
146. Wang, S.; Zhao, J.; Cui, Y.; Zhang, C.; Li, X. MAGiC: An LLM-Powered Multi-Agent Framework for Unleashing Visual Creativity. In Proceedings of the ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025, Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025); Lynce, I.; Murano, N.; Vallati, M.; Villata, S.; Chesani, F.; Milano, M.; Omicini, A.; Dastani, M., Eds. IOS Press, 2025, Frontiers in Artificial Intelligence and Applications, pp. 2146–2154. https://doi.org/10.3233/FAIA251054.
147. Liang, Z.; Zhang, D.; Zhou, H.; Huang, R.; Li, B.; Zhang, Y.; Wu, S.; Wang, X.; Luo, J.; Liao, L.; et al. UniVA: Universal Video Agent towards Open-Source Next-Generation Video Generalist, 2025, [arXiv:cs.CV/2511.08521].
148. Zhang, D.; Yao, W.; Wang, X.; Hu, Y.; Luo, J.; Yu, D. A Versatile Multimodal Agent for Multimedia Content Generation, 2026, [arXiv:cs.CV/2601.03250].
149. Li, J.; Huang, P.; Li, Y.; Chen, S.; Hu, J.; Tian, Y. A Unified Multi-Agent Framework for Universal Multimodal Understanding and Generation, 2025, [arXiv:cs.LG/2508.10494].
150. Guo, Z.; Zhang, F.; Jia, K.; Jin, T. LLM-I: LLMs are Naturally Interleaved Multimodal Creators, 2025, [arXiv:cs.LG/2509.13642].
151. Chen, R.; Chen, D.; Wu, S.; Wang, S.; Lang, S.; Sushko, P.; Jiang, G.; Wan, Y.; Krishna, R. MultiRef: Controllable Image Generation with Multiple Visual References. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025; Gurrin, C.; Schoeffmann, K.; Zhang, M.; Rossetto, L.; Rudinac, S.; Dang-Nguyen, D.; Cheng, W.; Chen, P.; Benois-Pineau, J., Eds. ACM, 2025, pp. 13325–13331. https://doi.org/10.1145/3746027.3758292.
152. Yang, J.; Luo, W.; Huang, H. EmoCtrl: Controllable Emotional Image Content Generation, 2026, [arXiv:cs.CV/2512.22437].
153. Zhou, Y.; Yuan, J.; Wang, Q. Draw ALL Your Imagine: A Holistic Benchmark and Agent Framework for Complex Instruction-based Image Generation, 2025, [arXiv:cs.CV/2505.24787].
154. Zhang, Y.; Li, J.; Tai, Y.W. LayerCraft: Enhancing Text-to-Image Generation with CoT Reasoning and Layered Object Integration. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 132139–132173.
155. Liu, Z.; Ning, M.; Zhang, Q.; Yang, S.; Wang, Z.; Yang, Y.; Xu, X.; Song, Y.; Chen, W.; Wang, F.; et al. CoT-lized Diffusion: Let's Reinforce T2I Generation Step-by-step. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 138729–138756.
156. Park, D.; Kim, S.; Moon, T.; Kim, M.; Lee, K.; Cho, J. Rare-to-Frequent: Unlocking Compositional Generation Power of Diffusion Models on Rare Concepts with LLM Guidance. In Proceedings of the International Conference on Learning Representations; Yue, Y.; Garg, A.; Peng, N.; Sha, F.; Yu, R., Eds., 2025, Vol. 2025, pp. 14650–14676.
157. Qin, L.; JIA, G.; Sun, Y.; Li, T.; Pan, H.; Yang, M.; Yang, X.; Qu, C.; Tan, Z.; Li, H. Uni-CoT: Towards Unified Chain-of-Thought Reasoning Across Text and Vision. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
158. Ye, Z.; Liu, Q.; Wei, C.; Zhang, Y.; Wang, X.; Wan, P.; Gai, K.; Luo, W. Visual-Aware CoT: Achieving High-Fidelity Visual Consistency in Unified Models, 2025, [arXiv:cs.CV/2512.19686].
159. Garg, S.; Singh, A.; Nayak, G.K. SIDiffAgent: Self-Improving Diffusion Agent, 2026, [arXiv:cs.AI/2602.02051].
160. Xiang, D.; Xu, W.; Chu, K.; Ding, T.; Shen, Z.; Zeng, Y.; Su, J.; Zhang, W. PromptSculptor: Multi-Agent Based Text-to-Image Prompt Optimization. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; Habernal, I.; Schulam, P.; Tiedemann, J., Eds., Suzhou, China, 2025; pp. 774–786. https://doi.org/10.18653/v1/2025.emnlp-demos.59.
161. Ye, W.; Liu, Z.; Yuwei, G.; Yuan, T.; Su, Y.; Fang, B.; Zhao, C.; Liu, Q.; Wang, L. GenPilot: A Multi-Agent System for Test-Time Prompt Optimization in Image Generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2025; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 929–958. https://doi.org/10.18653/v1/2025.findings-emnlp.49.

 doi:10.20944/preprints202607.1997.v1

162. Wan, X.; Zhou, H.; Sun, R.; Nakhost, H.; Jiang, K.; Sinha, R.; Arık, S.Ö. Maestro: Self-Improving Text-to-Image Generation via Agent Orchestration, 2025, [arXiv:cs.AI/2509.10704].
163. Sun, J.; Wang, H.; Cao, J.; Huang, H.; He, R. Marmot: Object-Level Self-Correction via Multi-Agent Reasoning, 2025, [2504.20054].
164. Akdemir, K.; Kazimi, T.; Yanardag, P. Audit & Repair: An Agentic Framework for Consistent Story Visualization in Text-to-Image Diffusion Models, 2025, [arXiv:cs.CV/2506.18900].
165. Li, M.; Hou, X.; Liu, Z.; Yang, D.; Qian, Z.; Chen, J.; Wei, J.; Jiang, Y.; Xu, Q.; Zhang, L. MCCD: Multi-Agent Collaboration-based Compositional Diffusion for Complex Text-to-Image Generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 13263–13272.
166. Xie, T.; Ma, R.; Wang, Q.; Ye, X.; Liu, F.; Tai, Y.; Zhang, Z.; Wang, L.; Yi, Z. Anywhere: A Multi-Agent Framework for User-Guided, Reliable, and Diverse Foreground-Conditioned Image Generation. In Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025; Walsh, T.; Shah, J.; Kolter, Z., Eds. AAAI Press, 2025, pp. 7410–7418. https://doi.org/10.1609/AAAI.V39I7.32797.
167. Zou, Z.; Yue, Z.; Du, K.; Bao, B.; Li, H.; Xie, H.; Xu, G.; Zhou, Y.; Wang, Y.; Hu, J.; et al. Beyond Textual CoT: Interleaved Text-Image Chains with Deep Confidence Reasoning for Image Editing, 2025, [arXiv:cs.CV/2510.08157].
168. Bandyopadhyay, D.; Mondal, A.K.; Dana, S.; Sharma, U.; AP, P.; Garg, D.; Singhee, A. A Plug-and-Play Agentic Framework for Text Guided Image Editing, 2026.
169. Ni, M.; Fan, Y.; Yang, Z.; Shen, Y.; Wei, Y.; Zhang, Y.; Wang, L.; Zhang, L.; Zuo, W. CoEditor++: Instruction-based Visual Editing via Cognitive Reasoning, 2026, [arXiv:cs.HC/2603.05518].
170. Pu, Y.; Zheng, H.; Mo, Z.; Zhang, H.; Fan, T.; Wu, S.; Wei, J. CAMEO: A Conditional and Quality-Aware Multi-Agent Image Editing Orchestrator, 2026, [arXiv:cs.CV/2604.03156].
171. Xu, Z.; Duan, H.; Nie, Y.; Du, M.; Wu, S.; Min, X.; Zheng, T.; Zhang, J.; Xu, S.; Chen, J.; et al. EditRefiner: A Human-Aligned Agentic Framework for Image Editing Refinement, 2026, [arXiv:cs.CV/2605.07457].
172. Chen, C.Y.; Shi, M.; Zhang, G.; Shi, H. T2I-Copilot: A Training-Free Multi-Agent Text-to-Image System for Enhanced Prompt Interpretation and Interactive Generation. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 19396–19405.
173. Liu, J.; Feng, R.; Wang, Y.; Zeng, W.; Jin, X. Generation Navigator: A State-Aware Agentic Framework for Image Generation, 2026, [arXiv:cs.CV/2605.17969].
174. Hahn, M.; Zeng, W.; Kannen, N.; Galt, R.; Badola, K.; Kim, B.; Wang, Z. Proactive Agents for Multi-Turn Text-to-Image Generation Under Uncertainty. In Proceedings of the Proceedings of the 42nd International Conference on Machine Learning; Singh, A.; Fazel, M.; Hsu, D.; Lacoste-Julien, S.; Berkenkamp, F.; Maharaj, T.; Wagstaff, K.; Zhu, J., Eds. PMLR, 13–19 Jul 2025, Vol. 267, *Proceedings of Machine Learning Research*, pp. 21591–21628.
175. Wei, Y.; Zhang, Z.; Ren, J.; Xu, X.; Hong, R.; Yang, Y.; Yan, S.; Wang, M. Clarity ChatGPT: An Interactive and Adaptive Processing System for Image Restoration and Enhancement, 2023, [arXiv:cs.CV/2311.11695].
176. Jin, X.; Shi, Y.; Xia, B.; Yang, W. LLMRA: Multi-modal Large Language Model based Restoration Assistant, 2024, [arXiv:cs.CV/2401.11401].
177. Li, B.; Li, X.; Lu, Y.; Chen, Z. Hybrid Agents for Image Restoration, 2025, [arXiv:cs.CV/2503.10120].
178. Lu, J.; Wu, Y.; Zhao, Z.; Wang, H.; Jimenez, F.; Majeedi, A.; Fu, Y. Restore-R1: Efficient Image Restoration Agents via Reinforcement Learning with Multimodal LLM Perceptual Feedback, 2025, [2512.18599].
179. Zhang, Y.; Jia, G.; Hu, H.; Zhao, S.; Zhao, K.; Sun, L.; Long, X.; Tian, K.; Jiang, C.; Liu, Z.; et al. TIR-Agent: Training an Explorative and Efficient Agent for Image Restoration, 2026, [arXiv:cs.CV/2603.27742].
180. Shen, S.; Liang, J.; Cai, C.; Geng, C.; Duan, H.; Zhang, X.; Hu, Q.; Zhai, G. Agentic Retoucher for Text-To-Image Generation, 2026. arXiv.
181. Chen, H.; Tao, K.; Wang, Y.; Wang, X.; Zhu, L.; Gu, J. PhotoArtAgent: Intelligent Photo Retouching with Language Model-Based Artist Agents, 2025, [arXiv:cs.CV/2505.23130].
182. Wu, Q.; Shi, J.; Jenni, S.; Kafle, K.; Wang, T.; Chang, S.; Zhao, H. RETOUCHIQ: MLLM Agents for Instruction-Based Image Retouching with Generalist Reward, 2026. arXiv.
183. Jiang, K.; Wang, Y.; Zhou, J.; Li, P.; Liu, Z.; Xie, C.W.; Chen, Z.; Zheng, Y.; Zhang, W. GenAgent: Scaling Text-to-Image Generation via Agentic Multimodal Reasoning, 2026, [arXiv:cs.CV/2601.18543].

 doi:10.20944/preprints202607.1997.v1

184. Li, C.; Wu, Q.; Pan, J.H.; Hui, K.H.; Hu, J.; Jiang, Y.; Sheng, B.; Liu, X.; Gong, W.; Liu, Z. coDrawAgents: A Multi-Agent Dialogue Framework for Compositional Image Generation, 2026, [arXiv:cs.CV/2603.12829].
185. Kovalev, V.; Kuvshinov, A.; Buzovkin, A.; Pokidov, D.; Timonin, D. CRAFT: Continuous Reasoning and Agentic Feedback Tuning for Multimodal Text-to-Image Generation, 2025, [arXiv:cs.CV/2512.20362].
186. Shi, J.; Qi, M.; Zhang, L.; Wang, D.; Zhao, Y.; Li, Z.; Xing, Y.; Li, N. Collaborative Text-to-Image Generation via Multi-Agent Reinforcement Learning and Semantic Fusion, 2025, [arXiv:cs.AI/2510.10633].
187. Jiang, X.; Li, G.; Chen, B.; Zhang, J. Multi-Agent Image Restoration, 2025, [arXiv:cs.CV/2503.09403].
188. Chen, H.; Li, W.; Gu, J.; Ren, J.; Chen, S.; Ye, T.; Pei, R.; Zhou, K.; Song, F.; Zhu, L. RestoreAgent: Autonomous Image Restoration Agent via Multimodal Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 110643–110666. https://doi.org/10.52202/079017-3512.
189. Wang, Y.; Yan, Q.; Zhou, J.; Dai, D.; Dong, W. PaAgent: Portrait-Aware Image Restoration Agent via Subjective-Objective Reinforcement Learning, 2026, [arXiv:cs.CV/2603.17055].
190. Lin, Y.; Lin, Z.; Lin, K.; Bai, J.; Pan, P.; Li, C.; Chen, H.; Wang, Z.; Ding, X.; Li, W.; et al. JarvisArt: Liberating Human Artistic Creativity via an Intelligent Photo Retouching Agent. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 52088–52130.
191. Lin, Y.; Wang, L.; Lin, K.; Lin, Z.; Gong, K.; Li, W.; Lin, B.; Li, Z.; Zhang, S.; Peng, Y.; et al. JarvisEvo: Towards a Self-Evolving Photo Editing Agent with Synergistic Editor-Evaluator Optimization, 2025. arXiv.
192. Jia, Q.; Liu, Y.; Chai, Y.; Yao, X.; Lu, Q.; Zhang, Y.; Shi, R.; Huang, Y.; Zhang, G. Lego-Edit: A General Image Editing Framework with Model-Level Bricks and MLLM Builder, 2025, [arXiv:cs.CV/2509.12883].
193. He, J.; Ye, J.; Huang, Z.; Jiang, D.; Zhang, C.; Zhu, L.; Zhang, R.; Zhang, X.; Li, W. Mind-Brush: Integrating Agentic Cognitive Search and Reasoning into Image Generation, 2026, [arXiv:cs.CV/2602.01756].
194. Son, M.H.; Oh, J.; Mun, S.B.; Roh, J.; Choi, S. World-To-Image: Grounding Text-to-Image Generation with Agent-Driven World Knowledge, 2025, [arXiv:cs.CV/2510.04201].
195. Hong, S.; Seo, J.; Shin, H.; Hong, S.; Kim, S. Large Language Models are Frame-Level Directors for Zero-Shot Text-to-Video Generation. In Proceedings of the First Workshop on Controllable Video Generation @ICML24, 2024.
196. Lu, Y.; Zhu, L.; Fan, H.; Yang, Y. FlowZero: Zero-Shot Text-to-Video Synthesis with LLM-Driven Dynamic Scene Syntax, 2023, [arXiv:cs.CV/2311.15813].
197. Su, S.; Guo, L.; Gao, L.; Shen, H.; Song, J. MotionZero:Exploiting Motion Priors for Zero-shot Text-to-Video Generation, 2023, [arXiv:cs.CV/2311.16635].
198. Yang, X.; Li, B.; Zhang, Y.; Yin, Z.; Bai, L.; Ma, L.; Wang, Z.; Cai, J.; Wong, T.T.; Lu, H.; et al. VLIPP: Towards Physically Plausible Video Generation with Vision and Language Informed Physical Prior. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 12360–12370.
199. Xue, Q.; Yin, X.; Yang, B.; Gao, W. PhyT2V: LLM-Guided Iterative Self-Refinement for Physics-Grounded Text-to-Video Generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 18826–18836.
200. Liu, Y.; Zhao, X.; Wen, P.; Dai, S.; Huang, Q. Bootstrapping Physics-Grounded Video Generation through VLM-Guided Iterative Self-Refinement, 2025, [arXiv:cs.CV/2511.20280].
201. Huang, Y.; Wang, Z.; Lin, H.; Kim, D.K.; Omidshafiei, S.; Yoon, J.; Zhang, Y.; Bansal, M. Planning with Sketch-Guided Verification for Physics-Aware Video Generation, 2025, [arXiv:cs.CV/2511.17450].
202. Long, D.X.; Wan, X.; Nakhost, H.; Lee, C.Y.; Pfister, T.; Arık, S.Ö. VISTA: A Test-Time Self-Improving Video Generation Agent, 2025. arXiv.
203. Zhu, H.; He, T.; Tang, A.; Guo, J.; Chen, Z.; Bian, J. Compositional 3D-aware Video Generation with LLM Director. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 131618–131644. https://doi.org/10.52202/079017-4184.
204. Li, Z.; Zhou, R.; Sajnani, R.; Cong, X.; Ritchie, D.; Sridhar, S. GenHSI: Controllable Generation of Human-Scene Interaction Videos. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2026, pp. 138–149.

 doi:10.20944/preprints202607.1997.v1

205. Liao, X.; Zeng, X.; Wang, L.; Yu, G.; Lin, G.; Zhang, C. MotionAgent: Fine-grained Controllable Video Generation via Motion Field Agent. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 11305–11316.
206. Zheng, J.; Cun, X. FairyGen: Storied Cartoon Video from a Single Child-Drawn Character. In Proceedings of the Proceedings of the SIGGRAPH Asia 2025 Conference Papers, SA Conference Papers 2025, Hong Kong, December 15-18, 2025; Komura, T.; Wimmer, M.; Fu, H., Eds. ACM, 2025, pp. 136:1–136:11. https://doi.org/10.1145/3757377.3764007.
207. Huang, K.; Huang, Y.; Wang, X.; Lin, Z.; Ning, X.; Wan, P.; ZHANG, D.; Wang, Y.; Liu, X. FilMaster: Bridging Cinematic Principles and Generative AI for Automated Film Generation. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
208. Tai, H.; Zhang, J.; Xu, Y.; Xue, Z.; Cao, W.; Wang, C.; Hu, X.; Liu, Y. Data-free VFX Self-Mining, 2026.
209. Yuan, Z.; Liu, Y.; Cao, Y.; Sun, W.; Jia, H.; Chen, R.; Li, Z.; Lin, B.; Yuan, L.; He, L.; et al. Mora: Enabling Generalist Video Generation via A Multi-Agent Framework, 2024, [arXiv:cs.CV/2403.13248].
210. He, Y.; Pittaluga, F.; Jiang, Z.; Zwicker, M.; Chandraker, M.; Tasneem, Z. LangDriveCTRL: Natural Language Controllable Driving Scene Editing with Multi-modal Agents, 2026, [arXiv:cs.CV/2512.17445].
211. Zhu, Z.; Wang, R.; Lyu, S.; Zhang, M.; Wu, B. BrandFusion: A Multi-Agent Framework for Seamless Brand Integration in Text-to-Video Generation, 2026, [arXiv:cs.CV/2603.02816].
212. Yang, C.; Li, P.; Qi, J.; Zhou, A.; Wu, J.; Liu, J. SCMAPR: Self-Correcting Multi-Agent Prompt Refinement for Complex-Scenario Text-to-Video Generation, 2026, [arXiv:cs.AI/2604.05489].
213. Fan, J.; Shen, J.; Yao, Y.; Wang, S.; Wang, Q.; Wang, Y. Communicative Agents for Slideshow Storytelling Video Generation based on LLMs, 2025, [arXiv:cs.AI/2509.01277].
214. Xie, Z.; Tang, D.; Tan, D.; Klein, J.; Bissyand, T.F.; Ezzini, S. DreamFactory: Pioneering Multi-Scene Long Video Generation with a Multi-Agent Framework, 2024, [arXiv:cs.AI/2408.11788].
215. Zheng, M.; Xu, Y.; Huang, H.; Ma, X.; Liu, Y.; Shu, W.; Pang, Y.; Tang, F.; Chen, Q.; Yang, H.; et al. VideoGen-of-Thought: Step-by-step generating multi-shot video with minimal manual intervention, 2025, [arXiv:cs.CV/2412.02259].
216. Lin, H.; Zala, A.; Cho, J.; Bansal, M. VideoDirectorGPT: Consistent Multi-Scene Video Generation via LLM-Guided Planning. In Proceedings of the First Conference on Language Modeling, 2024.
217. Hu, H.; Mao, Q.; Li, Y.; Jin, L. Camera Artist: A Multi-Agent Framework for Cinematic Language Storytelling Video Generation, 2026, [arXiv:cs.AI/2604.09195].
218. Song, Y.; Song, Y.; Losier, N.; Hodson, N.; Jin, Y.; Zhu, R.; Xu, Y.; Vlasic, D.; Claassen, C.; Leon, J.; et al. Co-Director: Agentic Generative Video Storytelling, 2026, [arXiv:cs.AI/2604.24842].
219. Song, Y.; Zhong, H.; Lin, K.Q.; Wang, H.; Shou, M.Z. Soap2Soap: Long Cinematic Video Remaking via Multi-Agent Collaboration, 2026, [arXiv:cs.CV/2605.17423].
220. Li, Y.; Shi, H.; Hu, B.; Wang, L.; Zhu, J.; Xu, J.; Zhao, Z.; Zhang, M. Anim-Director: A Large Multimodal Model Powered Agent for Controllable Animation Video Generation. In Proceedings of the SIGGRAPH Asia 2024 Conference Papers, SA 2024, Tokyo, Japan, December 3-6, 2024; Igarashi, T.; Shamir, A.; Zhang, H.R., Eds. ACM, 2024, pp. 34:1–34:11. https://doi.org/10.1145/3680528.3687688.
221. Hou, X.; Ma, B.; Cheng, J.; Ren, X.; Yu, K.; Li, W.; Zheng, T.; Lu, Q. PersonaVlog: Personalized Multimodal Vlog Generation with Multi-Agent Collaboration and Iterative Self-Correction, 2025, [arXiv:cs.CV/2508.13602].
222. Zhuang, S.; Li, K.; Chen, X.; Wang, Y.; Liu, Z.; Qiao, Y.; Wang, Y. Vlogger: Make Your Dream A Vlog. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 8806–8817.
223. Wang, W.F.; Lu, C.T.; Ng, J.P.; Chiu, Y.T.; Lee, T.Y.; Wang, M.; Chen, B.Y.; Chen, X.A. AnimAgents: Coordinating Multi-Stage Animation Pre-Production with Human-Multi-Agent Collaboration, 2025, [arXiv:cs.HC/2511.17906].
224. Sandoval-Castañeda, M.; Russell, B.C.; Sivic, J.; Shakhnarovich, G.; Heilbron, F.C. EditDuet: A Multi-Agent System for Video Non-Linear Editing. In Proceedings of the Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH Conference Papers 2025, Vancouver, BC, Canada, August 10-14, 2025; Alford, G.; Zhang, H.R.; Schulz, A., Eds. ACM, 2025, pp. 2:1–2:11. https://doi.org/10.1145/3721238.3730761.
225. Liu, L.; Cai, C.; Shen, S.; Liang, J.; Ouyang, W.; Ye, T.; Mao, J.; Duan, H.; Yao, J.; Zhang, X.; et al. MoA-VR: A Mixture-of-Agents System Toward All-in-One Video Restoration. *IEEE J. Sel. Top. Signal Process.* **2025**, *19*, 1822–1837. https://doi.org/10.1109/JSTSP.2025.3627128.

 doi:10.20944/preprints202607.1997.v1

226. He, L.; Song, Y.; Huang, H.; Liu, P.; Tang, Y.; Aliaga, D.; Zhou, X. Kubrick: Multimodal Agent Collaborations for Synthetic Video Generation, 2025, [arXiv:cs.CV/2408.10453].
227. Xu, Z.; Wang, J.; Wang, L.; Li, Z.; Shi, S.; Hu, B.; Zhang, M. FilmAgent: Automating Virtual Film Production Through a Multi-Agent Collaborative Framework. In Proceedings of the SIGGRAPH Asia 2024 Technical Communications, SA 2024, Tokyo, Japan, December 3-6, 2024; Igarashi, T.; Hu, R., Eds. ACM, 2024, pp. 15:1–15:4. https://doi.org/10.1145/3681758.3698014.
228. Bai, X.; Liang, H.; Galoaa, B.; Nandi, U.; Moezzi, S.; He, Y.; Ostadabbas, S. MoReGen: Multi-Agent Motion-Reasoning Engine for Code-based Text-to-Video Synthesis, 2025. arXiv.
229. Mu, L.; Qiang, W.; Jiang, F.; Wang, M.; Xu, M.; Zhang, K. FantasyHSI: Video-Generation-Centric 4D Human Synthesis in Any Scene Through a Graph-Based Multi-Agent Framework. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 8116–8124. https://doi.org/10.1609/aaai.v40i10.37758.
230. Cudlenco, N.; Masala, M.; Leordeanu, M. Agentic Video Generation: From Text to Executable Event Graphs via Tool-Constrained LLM Planning, 2026, [arXiv:cs.CV/2604.10383].
231. Chen, J.; Fu, Y.; Zeng, A.; Wang, Z.; Cen, S.; Yu, X.; Tanke, J.; Chen, Y.; Saito, K.; Mitsufuji, Y.; et al. Orator: LLM-Guided Multi-Shot Speech Video Generation, 2025.
232. Wei, Z.; Li, M.; Zhang, Z.; Yuan, R.; Hui, P.; Qu, H.; Evans, J.; Agrawala, M.; Rao, A. Hollywood Town: Long-Video Generation via Cross-Modal Multi-Agent Orchestration, 2025, [arXiv:cs.MA/2510.22431].
233. Liu, J.; Zhang, Y.; Li, S.; Lei, X. Vibe AIGC: A New Paradigm for Content Generation via Agentic Orchestration, 2026, [arXiv:cs.AI/2602.04575].
234. Tang, X.; Lei, X.; Zhu, C.; Chen, S.; Yuan, R.; Li, Y.; Oh, C.; Zhang, G.; Huang, W.; Benetos, E.; et al. AutoMV: An Automatic Multi-Agent System for Music Video Generation, 2025, [arXiv:cs.MM/2512.12196].
235. Yang, Y.; Liao, Y.; Mei, J.; Wang, B.; Yang, X.; Wen, L.; Zhang, J.; Li, X.; Lv, L.; Chen, H.; et al. SPIRAL: Self-Evolving Action-Conditioned Video Generation via Reflective Planning Agents, 2026, [arXiv:cs.CV/2603.08403].
236. Guo, Y.; Wang, T.; Ge, Y.; Ma, S.; Ge, Y.; Zou, W.; Shan, Y. AudioStory: Generating Long-Form Narrative Audio with Large Language Models, 2025, [arXiv:cs.CV/2508.20088].
237. Li, J.; Xu, T.; Chen, X.; Yao, X.; Han, J.; Liu, S. Mozart's Touch: a lightweight multimodal music generation framework based on pre-trained large models. In Proceedings of the International Conference on AI-Generated Content (AIGC 2024); Miao, D.; Zhao, F., Eds. SPIE, July 2025, p. 35. https://doi.org/10.1117/12.3067408.
238. Deng, Q.; Yang, Q.; Yuan, R.; Huang, Y.; Wang, Y.; Liu, X.; Tian, Z.; Pan, J.; Zhang, G.; Lin, H.; et al. ComposerX: Multi-Agent Symbolic Music Composition With LLMs. In Proceedings of the Proceedings of the 25th International Society for Music Information Retrieval Conference, ISMIR 2024, San Francisco, California, USA and Online, November 10-14, 2024; Kaneshiro, B.; Mysore, G.J.; Nieto, O.; Donahue, C.; Huang, C.A.; Lee, J.H.; McFee, B.; McCallum, M.C., Eds., 2024, pp. 669–679. https://doi.org/10.5281/ZENODO.14877425.
239. Bhandari, K.; Chang, S.; Roy, A.; Ronchini, F.; Benetos, E.; Herremans, D.; Colton, S. Text2Score: Generating Sheet Music From Textual Prompts, 2026, [arXiv:cs.SD/2605.13431].
240. Wang, Z.; Tang, C.K.; Tai, Y.W. Audio-Agent: Leveraging LLMs For Audio Generation, Editing and Composition, 2025, [arXiv:cs.SD/2410.03335].
241. Guo, Z.; Tu, T.; Ma, Y.; Yang, X. VibeMus: Proactive Agentic System for Music Personalization. In Proceedings of the Proceedings of the 7th ACM International Conference on Multimedia in Asia, MMAsia 2025, Kuala Lumpur, Malaysia, December 9-12, 2025; Chua, T.; Wong, L.; Chan, C.S.; Tang, J.; Ngo, C.; Schoeffmann, K.; Liu, J.; Ho, Y., Eds. ACM, 2025, pp. 162:1–162:3. https://doi.org/10.1145/3743093.3771663.
242. Zhao, Z.; Jin, D.; Zhou, Z. Zero-Effort Image-to-Music Generation: An Interpretable RAG-based VLM Approach, 2025, [arXiv:cs.SD/2509.22378].
243. Zhang, Y.; Xu, X.; Xu, X.; Liu, L.; Chen, Y. Long-Video Audio Synthesis with Multi-Agent Collaboration, 2025, [arXiv:cs.CV/2503.10719].
244. Zhang, Y.; Xu, X.; Xu, X.; Zhang, D.; Liu, L.; Chen, Y.C. Orchestrating Audio: Multi-Agent Framework for Long-Video Audio Synthesis. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 22267–22282. https://doi.org/10.18653/v1/2025.emnlp-main.1133.
245. Ren, Y.; Xu, X.; Zhang, Z.; Wu, W.; Li, B.; Zhang, C. AuDirector: A Self-Reflective Closed-Loop Framework for Immersive Audio Storytelling, 2026, [arXiv:cs.SD/2605.11866].
246. Rong, Y.; Wang, J.; Lei, G.; Yang, S.; Liu, L. AudioGenie: A Training-Free Multi-Agent Framework for Diverse Multimodality-to-Multiaudio Generation. In Proceedings of the Proceedings of the 33rd ACM International

 doi:10.20944/preprints202607.1997.v1

Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025; Gurrin, C.; Schoeffmann, K.; Zhang, M.; Rossetto, L.; Rudinac, S.; Dang-Nguyen, D.; Cheng, W.; Chen, P.; Benois-Pineau, J., Eds. ACM, 2025, pp. 8872–8881. https://doi.org/10.1145/3746027.3755758.

247. Karystinaios, E. WeaveMuse: An Open Agentic System for Multimodal Music Understanding and Generation. In Proceedings of the 1st Workshop on Large Language Models for Music & Audio (LLM4MA), 2025.
248. Aguina-Kang, R.; Gumin, M.; Han, D.H.; Morris, S.; Yoo, S.J.; Ganeshan, A.; Jones, R.K.; Wei, Q.A.; Fu, K.; Ritchie, D. Open-Universe Indoor Scene Generation using LLM Program Synthesis and Uncurated Object Databases, 2024, [arXiv:cs.CV/2403.09675].
249. Bai, T.; Bai, W.; Chen, D.; Wu, T.; Li, M.; Ma, R. FreeScene: Mixed Graph Diffusion for 3D Scene Synthesis from Free Prompts. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 5893–5903.
250. Liu, X.; Tai, Y.W.; Tang, C.K. Agentic 3D Scene Generation with Spatially Contextualized VLMs, 2025, [arXiv:cs.CV/2505.20129].
251. Sun, F.Y.; Liu, W.; Gu, S.; Lim, D.; Bhat, G.; Tombari, F.; Li, M.; Haber, N.; Wu, J. LayoutVLM: Differentiable Optimization of 3D Layout via Vision-Language Models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 29469–29478.
252. Ling, L.; Lin, C.H.; Lin, T.Y.; Ding, Y.; Zeng, Y.; Sheng, Y.; Ge, Y.; Liu, M.Y.; Bera, A.; Li, M. Scenethesis: A Language and Vision Agentic Framework for 3D Scene Generation. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.
253. Ran, X.; Li, Y.; Xu, L.; Yu, M.; Dai, B. Direct Numerical Layout Generation for 3D Indoor Scene Synthesis via Spatial Reasoning. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 125055–125081.
254. Bucher, M.J.; Armeni, I. ReSpace: Text-Driven Autoregressive 3D Indoor Scene Synthesis and Editing. In Proceedings of the The First Workshop on Efficient Spatial Reasoning, 2026.
255. Luo, J.; Tang, J.; Lu, R.; Zeng, G. SceneAssistant: A Visual Feedback Agent for Open-Vocabulary 3D Scene Generation, 2026, [arXiv:cs.CV/2603.12238].
256. Fang, S.; Wang, Y.; Tsai, Y.H.; Yang, Y.; Ding, W.; Zhou, S.; Yang, M.H. Chat-Edit-3D: Interactive 3D Scene Editing via Text Prompts. In Proceedings of the Computer Vision – ECCV 2024; Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; Varol, G., Eds., Cham, 2025; pp. 199–216.
257. Zheng, K.; Chen, X.; He, X.; Gu, J.; Li, L.; Yang, Z.; Lin, K.; Wang, J.; Wang, L.; Wang, X.E. EditRoom: LLM-parameterized Graph Diffusion for Composable 3D Room Layout Editing. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
258. El Amine Boudjoghra, M.; Laptev, I.; Dai, A. ScanEdit: Hierarchically-Guided Functional 3D Scan Editing. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 27105–27115.
259. Zhang, Y.; Wang, Z.; Lin, H.; Li, J.; Yang, J.; Bitton, Y.; Szpektor, I.; Bansal, M. Error-Driven Scene Editing for 3D Grounding in Large Language Models, 2025, [arXiv:cs.CV/2511.14086].
260. Bian, S.; Xu, C.; Xiu, Y.; Grigorev, A.; Liu, Z.; Lu, C.; Black, M.J.; Feng, Y. ChatGarment: Garment Estimation, Generation and Editing via Large Language Models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 2924–2934.
261. Huang-Menders, A.; Liu, X.; Xu, A.; Zhang, Y.; Tang, C.K.; Tai, Y.W. SmartAvatar: Text- and Image-Guided Human Avatar Generation with VLM AI Agents, 2025, [arXiv:cs.CV/2506.04606].
262. Xia, X.; Zhang, D.; Liao, Z.; Hou, Z.; Sun, T.; Li, J.; Fu, L.; Dong, Y. SceneGenAgent: Precise Industrial Scene Generation with Coding Agent. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 17847–17875. https://doi.org/10.18653/v1/2025.acl-long.873.
263. Zhang, S.; Jiang, C.; Li, Z.; Deng, J. ShapeCraft: LLM Agents for Structured, Textured and Interactive 3D Modeling. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 65116–65144.
264. Lu, S.; Chen, G.; Dinh, N.A.; Lang, I.; Holtzman, A.; Hanocka, R. LL3M: Large Language 3D Modelers, 2025, [arXiv:cs.GR/2508.08228].

 doi:10.20944/preprints202607.1997.v1

265. Wang, H.; Zhu, W.; Tang, S.; Wang, Z.; Dong, X.; Li, X.; Chen, X.; Bastola, A.; Huang, X.; Wang, Y.; et al. EZBlender: Efficient 3D Editing with Plan-and-ReAct Agent. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, March 2026, pp. 1343–1352.
266. Gao, J.; Juluri, S. From Idea to Co-Creation: A Planner-Actor-Critic Framework for Agent Augmented 3D Modeling. In Proceedings of the Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems. ACM, 2026, CHI EA '26, pp. 1–5. https://doi.org/10.1145/3772363.3798904.
267. Yin, S.; Ge, J.; Wang, Z.Z.; Wang, C.; Li, X.; Black, M.J.; Darrell, T.; Kanazawa, A.; Feng, H. Vision-as-Inverse-Graphics Agent via Interleaved Multimodal Reasoning, 2026, [arXiv:cs.CV/2601.11109].
268. Zhou, M.; Li, R.; Lyu, X.; Song, Z.; Huang, Z.; Zheng, C.; Rupprecht, C.; Vedaldi, A.; Wu, S. Articraft: An Agentic System for Scalable Articulated 3D Asset Generation, 2026, [arXiv:cs.CV/2605.15187].
269. Yang, Y.; Luo, Z.; Gan, W.; Hao, J.; Lu, J.; Yan, J.; Lyu, Z.; Xu, X. Code-as-Room: Generating 3D Rooms from Top-Down View Images via Agentic Code Synthesis, 2026, [arXiv:cs.CV/2605.18451].
270. Berger, E.; Usama, M.; Mehlstäubl, J.; Saske, B.; Paetzold-Byhain, K. Physics-in-the-Loop: A Hybrid Agentic Architecture for Validated CAD Engineering Design, 2026, [arXiv:cs.CV/2605.19717].
271. Ataei, M.; Askari, F.; Malekshan, K.R.; Jayaraman, P.K. Zero-to-CAD: Agentic Synthesis of Interpretable CAD Programs at Million-Scale Without Real Data, 2026, [arXiv:cs.CV/2604.24479].
272. Yang, Y.; Jia, B.; Zhang, S.; Huang, S. SceneWeaver: All-in-One 3D Scene Synthesis with an Extensible and Self-Reflective Agent. In Proceedings of the Advances in Neural Information Processing Systems; Belgrave, D.; Zhang, C.; Lin, H.; Pascanu, R.; Koniusz, P.; Ghassemi, M.; Chen, N., Eds. Curran Associates, Inc., 2025, Vol. 38, pp. 140319–140351.
273. Pfaff, N.; Cohn, T.; Zakharov, S.; Cory, R.; Tedrake, R. SceneSmith: Agentic Generation of Simulation-Ready Indoor Scenes, 2026, [arXiv:cs.RO/2602.09153].
274. Xia, H.; Li, X.; Li, Z.; Ma, Q.; Xu, J.; Liu, M.Y.; Cui, Y.; Lin, T.Y.; Ma, W.C.; Wang, S.; et al. SAGE: Scalable Agentic 3D Scene Generation for Embodied AI, 2026. arXiv.
275. Zhao, Y.; Sun, S.; Zhang, M.; Shi, Y.; Yang, X.; Bian, J. SceneReVis: A Self-Reflective Vision-Grounded Framework for 3D Indoor Scene Synthesis via Multi-turn RL, 2026, [arXiv:cs.CV/2602.09432].
276. Deng, W.; Qi, M.; Ma, H. Global-Local Tree Search in VLMs for 3D Indoor Scene Generation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2025, pp. 8975–8984.
277. Pun, H.I.D.; Tam, H.I.I.; Wang, A.; Huo, X.; Chang, A.X.; Savva, M. HSM: Hierarchical Scene Motifs for Multi-Scale Indoor Scene Generation. In Proceedings of the Thirteenth International Conference on 3D Vision, 2026.
278. Erkoç, Z.; Dai, A.; Nießner, M. WorldAgents: Can Foundation Image Models be Agents for 3D World Models?, 2026, [arXiv:cs.CV/2603.19708].
279. Çelen, A.; Han, G.; Schindler, K.; Van Gool, L.; Armeni, I.; Obukhov, A.; Wang, X. I-Design: Personalized LLM Interior Designer. In Proceedings of the Computer Vision – ECCV 2024 Workshops; Del Bue, A.; Canton, C.; Pont-Tuset, J.; Tommasi, T., Eds., Cham, 2025; pp. 217–234.
280. Liu, Z.; Tang, Z.; Wu, R.; Zheng, X.; Hu, J.; Hui, K.H.; Xie, H.; Dai, B.; Liu, Z. Imagine a City: CityGenAgent for Procedural 3D City Generation, 2026, [arXiv:cs.CV/2602.05362].
281. Wang, S.; Zheng, Z.; Shang, Y.; He, L.; Yu, Y.; Hangyu, F.; Feng, J.; Liao, Q.; Li, Y. RAISECity: A Multimodal Agent Framework for Reality-Aligned 3D World Generation at City-Scale, 2025, [arXiv:cs.CV/2511.18005].
282. Yuan, J.; Yang, B.; Wang, K.; Pan, P.; Ma, L.; Zhang, X.; Liu, X.; Cui, Z.; Ma, Y. ImmerseGen: Agent-Guided Immersive World Generation with Alpha-Textured Proxies. *IEEE Transactions on Visualization and Computer Graphics* **2026**, *32*, 3530–3540. https://doi.org/10.1109/tvcg.2026.3679097.
283. Lu, K.; Zhou, S.; Xu, H.; Xu, G.; Yang, Z.; Wang, Y.; Xiao, Z.; Long, J.; Li, M. Yo'City: Personalized and Boundless 3D Realistic City Scene Generation via Self-Critic Expansion, 2026, [arXiv:cs.CV/2511.18734].
284. Huang, Z.; He, J.; Huang, X.; Xiong, Z.; Luo, Y.; Ye, J.; Li, W.; Chen, Y.; Han, T. MajutsuCity: Language-driven Aesthetic-adaptive City Generation with Controllable 3D Assets and Layouts, 2025. arXiv.
285. Wei, Y.; Wang, J.; Du, Y.; Wang, D.; Pan, L.; Xu, C.; Feng, Y.; Dai, B.; Chen, S. ChatDyn: Language-Driven Multi-Actor Dynamics Generation in Street Scenes, 2024, [arXiv:cs.CV/2412.08685].
286. Wei, Y.; Wang, Z.; Lu, Y.; Xu, C.; Liu, C.; Zhao, H.; Chen, S.; Wang, Y. Editable Scene Simulation for Autonomous Driving via Collaborative LLM-Agents. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, pp. 15077–15087.

 doi:10.20944/preprints202607.1997.v1

287. Sun, J.; Zhang, F.; Feng, Y.; Li, C.; Li, Z.; Ai, J.; Chang, Y.; Dai, Y.; Zhang, K. From Pixels to Paths: A Multi-Agent Framework for Editable Scientific Illustration, 2025, [arXiv:cs.CV/2510.27452].
288. Zhou, K.; He, J.; Yang, W.; Wang, Z.; Zhang, Y.; Jia, W. Automatic Slide Updating with User-Defined Dynamic Templates and Natural Language Instructions, 2026, [arXiv:cs.CL/2604.17894].
289. Park, J.I.; Taneja, M.; Wang, Q.; Kang, D. Stealing Creator's Workflow: A Creator-Inspired Agentic Framework with Iterative Feedback Loop for Improved Scientific Short-form Generation, 2025, [arXiv:cs.CL/2504.18805].
290. Zeng, W.; Ouyang, M.; Cui, L.; Ng, H.T. SlideTailor: Personalized Presentation Slide Generation for Scientific Papers. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 34584–34592. https://doi.org/10.1609/aaai.v40i41.40758.
291. Choi, J.; Park, S.; Song, S.; Shim, H. PosterForest: Hierarchical Multi-Agent Collaboration for Scientific Poster Generation, 2026, [arXiv:cs.AI/2508.21720].
292. Shi, C.; Cai, Q.; Chen, Z.; Zeng, L.; Zhao, Y.; Yu, J.; Yu, J.; Li, X. APEX: Academic Poster Editing Agentic Expert, 2026, [arXiv:cs.AI/2601.04794].
293. Yan, L.; Wu, J.; Xie, D.; Shi, W.; Xia, D.; Huang, J. Beyond End-to-End Video Models: An LLM-Based Multi-Agent System for Educational Video Generation, 2026, [arXiv:cs.AI/2602.11790].
294. Ye, F.; Xie, Z.; Hu, Y.; Yin, Y.; Huang, S.; Dong, S.; Bao, J.; Yan, S. Deep-Reporter: Deep Research for Grounded Multimodal Long-Form Generation, 2026, [arXiv:cs.CL/2604.10741].
295. Zhuang, C.; Huang, A.; Hu, Y.; Wu, J.; Cheng, W.; Liao, J.; Wang, H.; Liao, X.; Cai, W.; Xu, H.; et al. ViStoryBench: Comprehensive Benchmark Suite for Story Visualization, 2026, [arXiv:cs.CV/2505.24862].
296. Wang, C.; Dai, Y.; Sun, L.; Du, J.; Gao, J. AudioAtlas: A Comprehensive and Balanced Benchmark Towards Movie-Oriented Text-to-Audio Generation. In Proceedings of the Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025; Gurrin, C.; Schoeffmann, K.; Zhang, M.; Rossetto, L.; Rudinac, S.; Dang-Nguyen, D.; Cheng, W.; Chen, P.; Benois-Pineau, J., Eds. ACM, 2025, pp. 13266–13272. https://doi.org/10.1145/3746027.3758284.
297. Hua, D.; Wang, X.; Zeng, B.; Huang, X.; Liang, H.; Niu, J.; Chen, X.; Xu, Q.; Zhang, W. VABench: A Comprehensive Benchmark for Audio-Video Generation, 2025. arXiv.
298. Jang, D.; Heisler, M.L.; Xing, L.; Li, Y.; Wang, E.; Xiong, Y.; Zhang, Y.; Fan, Z. DECKBench: Benchmarking Multi-Agent Frameworks for Academic Slide Generation and Editing, 2026, [arXiv:cs.AI/2602.13318].
299. Liu, Y.; Qian, Z.; Zhou, H.; Zhang, J.; Zhang, Y.; Li, Z.; Zhou, M.; Zhao, E.; Jiang, X.; Jiang, G. ATP-Bench: Towards Agentic Tool Planning for MLLM Interleaved Generation, 2026, [arXiv:cs.AI/2603.29902].
300. Chen, D.; Chen, R.; Pu, S.; Liu, Z.; Wu, Y.; Chen, C.; Liu, B.; Huang, Y.; Wan, Y.; Zhou, P.; et al. Interleaved Scene Graphs for Interleaved Text-and-Image Generation Assessment. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.
301. Qiu, L.; Li, Y.; Ge, Y.; Ge, Y.; Shan, Y.; Liu, X. AnimeShooter: A Multi-Shot Animation Dataset for Reference-Guided Video Generation, 2025, [arXiv:cs.CV/2506.03126].
302. Sun, W.; Wang, Z.; Hu, Z.; Wang, C.; Li, H.; Chen, W. Muse: A Multi-agent Framework for Unconstrained Story Envisioning via Closed-Loop Cognitive Orchestration, 2026. arXiv.
303. Nag, S.; J, J.K.; Goswami, K.; Morariu, V.I.; Srinivasan, B.V. Agentic Design Review System. *Proceedings of the AAAI Conference on Artificial Intelligence* **2026**, *40*, 1991–1999. https://doi.org/10.1609/aaai.v40i3.37180.
304. Wang, J.; Yang, X.; Wang, L.; Xu, Z.; Wang, Y.; Wang, Y.; Luo, W.; Zhang, K.; Hu, B.; Zhang, M. A Unified Agentic Framework for Evaluating Conditional Image Generation. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M.T., Eds., Vienna, Austria, 2025; pp. 12626–12646. https://doi.org/10.18653/v1/2025.acl-long.620.
305. Wang, Z.; Wang, J.; Ma, K.; Lin, D.; Zhou, B. TalkVerse: Democratizing Minute-Long Audio-Driven Video Generation, 2025, [arXiv:cs.CV/2512.14938].
306. Zhang, P.; Zhou, C.; Zhang, Z.; Liu, H.; Zhang, C.; Liu, J.; Zhou, X.; Chen, X.; Weng, S.; Li, S.; et al. A Benchmark and Multi-Agent System for Instruction-driven Cinematic Video Compilation, 2026, [arXiv:cs.CV/2604.10456].