PaperContract: A Structured Contract for Consistent and Editable Paper-to-Video Generation

Anonymous ACL submission

Abstract

Converting a research paper into a presentation video requires co-generating multiple synchronized modalities, including slides, speech, subtitles, and a talking head. Existing pipelines decouple these modalities into independent prompt-driven branches, which causes cross-modal drift and forces a full system re-run for any subsequent edit. We address it with **PaperContract**, a framework built around a persistent *structured contract* that every downstream module consumes as a single source of truth. During generation, slides are produced through QA-guided refinement, speech is synthesized once per slide, and word-level forced-alignment timestamps are projected onto a shared global timeline, so speech, subtitles, cursor, and talking-head are synchronized by construction. During revision, natural-language edits are resolved by updating the contract and selectively invalidating only the affected asset caches, avoiding redundant re-computation. On the Paper2Video benchmark, PaperContract improves cross-modal consistency and video quality over stage-wise and agentic baselines, while reducing editing overhead through contract-level cache reuse.

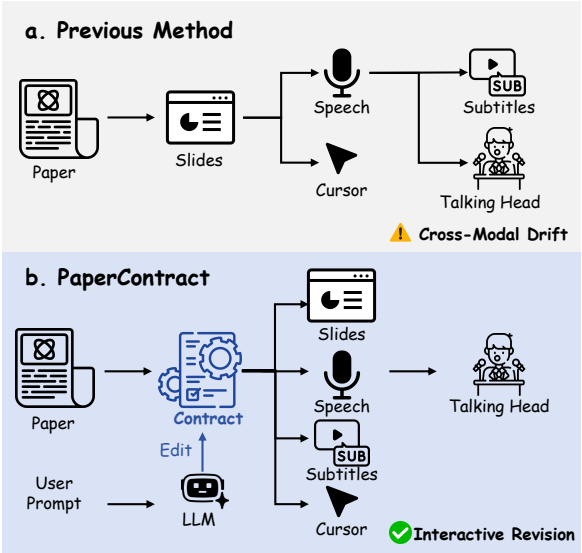

Figure 1: Comparison between conventional stage-wise paper-to-video pipelines and PaperContract. **(a)** Existing pipelines route the paper through a cascade of stages—slides, speech, subtitles, cursor, talking-head—each interpreting its upstream input with its own prompt and no shared semantic record, so cross-modal drift accumulates along the chain. **(b)** PaperContract inserts a single structured contract that holds the per-slide semantics; downstream modules read from the contract (talking-head remains audio-driven), and revision is applied by editing the contract rather than rerunning the full pipeline.

1 Introduction

Scientific communication increasingly relies on multimodal formats such as presentation videos, which combine slides, speech, subtitles, and visual guidance to convey complex ideas. Automatically generating such videos from research papers (Zhu et al., 2025; Shi et al., 2025a; Zheng et al., 2026) sits at the intersection of a broader effort to make scientific artifacts machine-actionable, alongside paper-to-code (Seo et al., 2025), paper-to-poster (Pang et al., 2025), and paper-to-agent (Miao et al., 2025) systems, with downstream applications in education, dissemination, and accessibility (Chen et al., 2025a).

Existing systems (Zhu et al., 2025; Shi et al., 2025a) decompose the task into a cascade of loosely coupled stages—slide synthesis (Zheng et al., 2025; Ge et al., 2025; Liang et al., 2025), speech (Chen et al., 2025b), subtitle alignment (Bain et al., 2023), cursor grounding (Qin et al., 2025; Lin et al., 2025), and talking-head rendering (Li et al., 2025; Cui et al., 2024)—each implemented as an independent module prompted from its upstream output, following multi-agent design patterns such as MetaGPT (Hong et al., 2024) and CAMEL (Li et al., 2023). Semantic interpretation of the paper is thus dispersed across

stages, with no module retaining the others' decisions (Fig. 1**a**).

We call this design pattern the *semantic re-derivation problem*: each stage re-interprets its upstream input with its own prompt, so the paper's semantics are repeatedly re-extracted along the chain with no shared intermediate representation across stages. Two structural drawbacks follow. First, dispersed semantic reconstruction across stages causes cross-modal inconsistencies: narration may introduce details absent from the slide, subtitles drift from the spoken track, or the cursor lands on an unrelated region. Second, the resulting video is effectively frozen: even a minor revision—*"shorten the methodology section"* or *"add a comparison slide"*—forces a full pipeline re-run, because the system retains no record of which semantic units actually changed.

We address both drawbacks with **PaperContract** (Fig. 1**b**). The core idea is to replace the stage-wise reinterpretation chain with a single *structured contract*: a persistent, machine-readable specification of per-slide narration, visible text, and visual elements that is constructed once from the paper and consumed by every downstream module as its authoritative source.

Given the contract, PaperContract proceeds in two phases. In generation, slides are rendered with QA-guided refinement, narration is synthesized once per slide, and word-level forced-alignment timestamps are lifted into a shared global timeline. As a result, subtitles, cursor grounding, and talking-head synthesis are synchronized by construction. In revision, a natural-language author request is resolved by updating the contract and selectively invalidating only the affected renders, audio, and animations; everything else is reused from cache.

Contributions. We make the following contributions:

- Contract-centered formulation. We identify the *semantic re-derivation problem* in existing paper-to-video pipelines and address it by recasting the task around a persistent *structured contract* that every downstream modality consumes as a single source of truth, replacing repeated prompt-level re-derivation.
- Contract-driven generation. We design a generation pipeline of QA-guided slide rendering, and forced-alignment-based temporal realization, which together guarantee cross-modal synchronization by construction.
- Contract-guided interactive revision. We turn natural-language edits into local updates on the contract and propagate them through selective cache invalidation, reducing edit-time cost without sacrificing output consistency.
- Empirical validation. We show on the Paper2Video benchmark (Zhu et al., 2025) that PaperContract improves cross-modal consistency and overall presentation quality over existing stage-wise and agent-based baselines.

2 Related Work

2.1 Document-to-Slide Generation

Generating presentation slides from scientific documents has been studied as structured summarization. Early work such as D2S (Sun et al., 2021) casts the task as query-driven summarization, while Doc2PPT (Fu et al., 2021) adds hierarchical sequence-to-sequence models with coarse layout prediction. More recent systems improve visual quality and personalization through iterative editing, program-guided layout control, and multi-agent collaboration (Zheng et al., 2025; Ge et al., 2025; Liang et al., 2025; Zeng et al., 2026). These systems treat the slide deck as the final artifact with no explicit semantic layer for downstream modalities to consume, making it hard to extend the same output into narration, subtitles, or interaction without re-deriving the underlying structure from scratch.

2.2 Presentation Video Generation

A more recent line of work extends document-to-slide pipelines into narrated presentation videos. PresentAgent (Shi et al., 2025a) produces synchronized slides and speech from diverse document sources, and PaperTalker (Zhu et al., 2025) focuses on academic papers and additionally integrates subtitles, cursor grounding, and a talking-head channel. Despite their broader scope, both generate each modality through an independent prompting pipeline conditioned on the source paper—what we term the *semantic re-derivation problem*. This surfaces as content-level mismatches when narration introduces facts absent from the slide, temporal drift when subtitles diverge from the spoken track, and spatial grounding errors when the cursor mispoints. The resulting video is also effectively read-only: any author revision must propagate through

every independent pipeline, forcing a full re-run. PaperContract addresses both issues by replacing prompt-level reinterpretation with a single structured contract that all modalities consume and that authors can directly edit.

2.3 Structured Intermediate Representations

Structured intermediate representations have improved consistency in adjacent multi-stage generation settings. Program sketches and typed APIs bridge natural-language intent and executable outputs in code generation (Austin et al., 2021; Wang et al., 2024), and typed communication protocols reduce inter-module ambiguity in multi-agent systems (Hong et al., 2024). These designs share our intuition that an explicit machine-readable interface is more reliable than repeated natural-language reinterpretation. They differ from our setting in two ways. First, they target reasoning and task decomposition rather than the alignment of heterogeneous modalities such as slide pixels, narration audio, word-level timings, cursor positions, and talking-head frames. Second, they do not support localized post-hoc revision.

PaperContract instantiates the structured-interface idea for paper-to-video generation. The contract serves as a single authoritative source for every modality, explicitly binds visual elements to the narration units that reference them, and exposes a timeline structure consumed by subtitles, cursor animation, and talking-head synthesis by construction. Because it is persistent and structured, revision reduces to a cache-invalidation problem rather than a fresh generation pass.

3 Method

Overview. We propose **PaperContract**, a contract-centered framework for paper-to-video generation (Fig. 2). Instead of letting each modality re-derive the paper's semantics independently, we extract a *structured contract*—a persistent, machine-readable specification of per-slide narration, visible text, and visual assets—that every downstream module consumes as the single source of truth. This design directly targets the two structural drawbacks of stage-wise pipelines. First, when slides, narration, subtitles, cursor, and talking-head are each prompted independently from the paper, semantic drift accumulates across stages and surfaces as cross-modal inconsistency. Reading from a single contract removes that source of error, turning synchronization into a property enforced by construction rather than recovered post hoc. Second, because the contract is explicit and persistent, an author revision is a localized edit on $\mathcal{C}$ rather than a full pipeline re-run: only the affected units are re-realized, while everything else is reused from cache.

Given a paper $\mathcal{D}$, a speaker image $\mathcal{I}$, and a reference voice $\mathcal{A}$, **PaperContract** operates in four sequential stages (Fig. 2): **(1) Contract Construction** (§3.1) parses $\mathcal{D}$ into a Markdown intermediate and prompts an LLM to emit the structured contract $\mathcal{C}$, which encodes paper metadata and per-slide units. **(2) Contract-Constrained Slide Rendering** (§3.2) expands each slide unit into a layout-ready specification and wraps VLM rendering in a QA loop, where a judge scores candidate layouts and a learned action scorer chooses between REFINE and REGENERATE. **(3) Contract-Driven Temporal Realization** (§3.3) synthesizes narration audio and runs a single forced-alignment pass that produces a shared word-level timing reference for subtitles, cursor animation, and talking-head clips. **(4) Contract-Guided Interactive Revision** (§3.4) maps user edits onto $\mathcal{C}$ and propagates them via selective invalidation, so unchanged units reuse cached artifacts.

3.1 Contract Construction

We construct a *structured contract* $\mathcal{C}$ from the input paper $\mathcal{D}$, turning a long, loosely structured document into a bounded sequence of presentation units that downstream modules can consume without further parsing. Concretely, we convert the rendered PDF into a Markdown intermediate that linearizes the paper text and records paths to extracted figures, so downstream modules never reopen the source PDF and the method applies even when the LaTeX source is unavailable. An LLM then induces the contract from this representation, with each unit defined as

$$c_i = \left(\mathbf{z}_i^{\text{title}},\ \mathbf{z}_i^{\text{nar}},\ \mathbf{z}_i^{\text{text}},\ \mathbf{z}_i^{\text{figure}}\right), \tag{1}$$

where $\mathbf{z}_i^{\text{title}}$, $\mathbf{z}_i^{\text{nar}}$, $\mathbf{z}_i^{\text{text}}$, and $\mathbf{z}_i^{\text{figure}}$ are the slide title, narration, on-slide text (caption and bullets), and figure paths.

The full contract is emitted as a strict JSON object

$$\mathcal{C} = \{\texttt{title},\ \texttt{authors},\ \texttt{institutions},\ c_i\ \forall i\}, \tag{2}$$

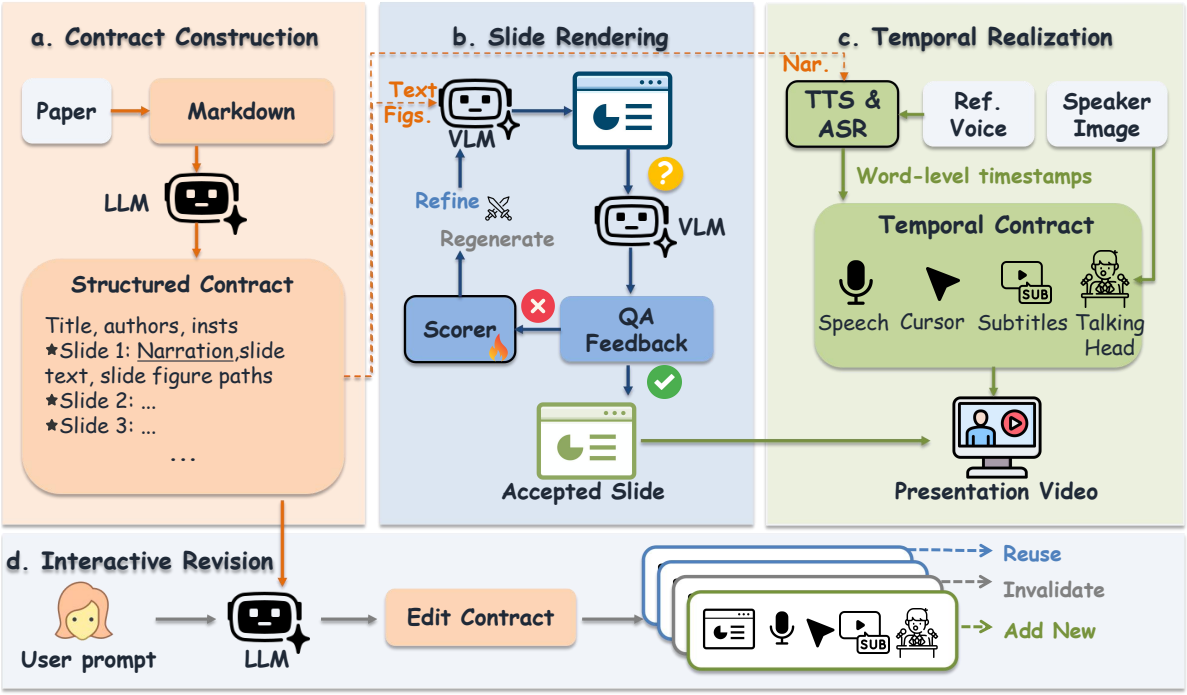

Figure 2: **Overview of PaperContract. (a) Contract Construction** converts the input paper into Markdown; an LLM then compiles this data into a unified *Structured Contract* containing per-slide narration, text, and asset references. **(b) Slide Rendering** inputs these elements into a VLM, entering a QA-guided verification loop where a VLM judge provides feedback and a scorer decides to REFINE or REGENERATE the layout until it passes quality thresholds. **(c) Temporal Realization** synthesizes speech via TTS using a reference voice, extracts word-level timestamps, and projects them into a *Temporal Contract* that simultaneously anchors subtitles, cursor motion, and talking-head animations for final video composition. **(d) Interactive Revision** processes natural-language user prompts through an LLM to update the contract, allowing unchanged assets to be *reused*, modified assets to be *invalidated*, and new assets to be *added* incrementally.

constrained to preserve document order and to attach each figure path to the slide where it is discussed, so later modules need not recover these alignments. The serialized contract is persisted once and consumed by all downstream stages.

3.2 Contract-Constrained Slide Rendering

We deterministically unpack each unit c_i into a slide specification

$$d_i = (\mathbf{z}_i^{\text{title}},\ \mathbf{z}_i^{\text{text}},\ \mathbf{z}_i^{\text{figure}}), \tag{3}$$

and render a slide image $x_i = \text{VLM}(d_i, \mathcal{T})$ from a layout template $\mathcal{T}$. Raster figures are injected from $\mathbf{z}_i^{\text{figure}}$, and a retrieved institution logo is pasted onto each slide.

QA-Guided Refinement. Because a single VLM call is unreliable, we wrap rendering in a bounded multi-attempt search with a learned repair policy, shown as Fig. 2 **b**. Each attempt a is reviewed by a VLM judge $q_{i,a} = \text{J}(x_{i,a}, \mathcal{T})$ that returns content/style scores $(s^c_{i,a}, s^s_{i,a})$ together with structured feedback on content, style, layout, and text gaps. We track the current best attempt $\hat{a}_i$ via the lexicographic key $\left(\min(s^c, s^s),\ \frac{s^c+s^s}{2},\ -a\right)$. A usable slide must clear *both* rubrics, so we let the bottleneck axis govern ranking—this prevents a strong content score from masking poor style, and vice versa, instead of letting one axis compensate for the other. The mean then rewards higher overall quality among equally balanced attempts, and $-a$ resolves the remainder in favor of the earliest attempt, anchoring the search to a known-good candidate rather than drifting through equivalent re-renders.

If $\hat{a}_i$ has not yet passed the rubric thresholds τ^c and τ^s, we route the judge's feedback to a learned *action scorer* that decides how to spend the next attempt. *Refine* re-conditions VLM on the current image and feedback, while *regenerate* resamples from d_i. Since neither action dominates a priori—refinement is cheap when the layout is nearly correct, regeneration is necessary when it is not—we treat the choice as a contextual bandit and score

each action u by

$$\text{score}(u) = b_u + w_u^\top \phi(q_{i,\hat{a}_i}), \tag{4}$$

where ϕ encodes the four feedback gaps into a context vector. We execute the higher-scoring action, then update its parameters with reward $\rho = \frac{1}{2}\big((s^c - \tau^c)/\tau^c + (s^s - \tau^s)/\tau^s\big)$ via clipped gradient steps

$$\begin{aligned} w_u &\leftarrow w_u + \eta\, \rho\, \phi, \\ b_u &\leftarrow b_u + \eta\, \rho. \end{aligned} \tag{5}$$

The two scorers serve complementary roles: the judge scores *slides* to determine acceptance, while the action scorer scores *actions* to allocate the remaining attempts. Search terminates once both thresholds are met or the retry budget is exhausted, at which point we accept the best-so-far candidate as $x_i^\star$.

3.3 Contract-Driven Temporal Realization

Coordinating subtitles, cursor animation, and talking-head clips requires a shared temporal reference. Prior systems (Zhu et al., 2025) align each channel independently, which yields only approximate synchronization. We instead derive per-slide temporal contracts $\mathcal{V} = \{\mathcal{V}_i\}_{i=1}^N$ from a single forced-alignment pass, and let every downstream module consume them as immutable timing ground truth.

Speech synthesis and alignment. For each slide i, we synthesize narration audio as $a_i = \text{TTS}(\mathbf{z}_i^{\text{nar}}, \mathcal{A})$, with the sampler seeded deterministically from a hash of $(\mathbf{z}_i^{\text{nar}}, \mathcal{A})$. Determinism guarantees that identical narration under the same speaker reference yields byte-identical audio across runs, a property we exploit in Section 3.4 to reuse cached clips and preserve voice consistency across revisions. Forced alignment then produces a word-level timing record

$$\mathcal{V}_i = \text{Align}(a_i,\ \mathbf{z}_i^{\text{nar}}) = \{(w_{i,j},\ t_{i,j}^s,\ t_{i,j}^e)\}_{j=1}^{m_i}. \tag{6}$$

We lift these local timestamps onto a global timeline via the cumulative offset $\Delta_i = \sum_{k=1}^{i-1} \text{dur}(a_k)$:

$$\hat{t}_{i,j}^s = t_{i,j}^s + \Delta_i, \qquad \hat{t}_{i,j}^e = t_{i,j}^e + \Delta_i, \tag{7}$$

so subtitles are globally synchronized without a second alignment pass.

Cursor and talking-head. The same temporal contract drives the remaining visual channels. For cursor animation, we segment the narration into half-sentence units $\{g_{i,k}\}$ and ground each unit on the rendered slide via UI-TARS (Qin et al., 2025): $p_{i,k} = \text{Ground}(x_i^\star, g_{i,k})$. The cursor moves to $p_{i,k}$ over the aligned interval $[\hat{t}_{i,\ell_k}^s, \hat{t}_{i,r_k}^e]$. We synthesize talking-head clips as $h_i = \text{M}(\mathcal{I}, a_i)$ using Ditto (Li et al., 2025) and composite them in a picture-in-picture layout. Because subtitles, cursor, and talking-head all reference the same $\mathcal{V}$, cross-modal synchronization is guaranteed by construction rather than recovered post hoc.

3.4 Contract-Guided Interactive Revision

Authors typically refine a generated video through natural-language instructions such as "add a comparison slide" or "shorten the methodology section." Re-running the full pipeline for such edits is wasteful, and because the contract $\mathcal{C}$ is explicit and persistent, we localize edits at the contract level and propagate only the necessary changes downstream.

Contract editing. Given the current contract and a user request r, an LLM produces a revised contract together with a *change plan* Π that lists the affected assets and their operations (modify, add, or delete). The prompt enforces minimal edits and the contract schema, so a localized request does not perturb unrelated slides.

Structural diff. Because model-emitted plans can under-report changes, we cross-check Π with a deterministic diff. Let $\sigma_i = \text{Hash}(c_i)$ denote the content signature of slide c_i, and let $\mathcal{C}' = \{c_i'\}_{i=1}^{N'}$ with signatures $\{\sigma_i'\}$ be the revised contract. Writing $i^\star = \min\{i | \sigma_i \neq \sigma_i'\}$ for the first divergence, the structural diff set is

$$\mathcal{S}_{\text{diff}} = \begin{cases} \{\, i | \sigma_i \neq \sigma_i' \,\}, & N = N', \\ \{\, i | i \geq i^\star \,\}, & N \neq N', \end{cases} \tag{8}$$

and the final affected set is $\mathcal{S} = \mathcal{S}_\Pi \cup \mathcal{S}_{\text{diff}}$, where $\mathcal{S}_\Pi$ collects the slide indices reported in Π. This design serves as a crucial safeguard, guaranteeing that no silent omissions escape detection.

Selective invalidation. Given the affected set $\mathcal{S}$, we invalidate rendered slides and talking-head clips for all $i \in \mathcal{S}$. Narration audio is keyed by content hash rather than slide index, so a slide whose narration is byte-identical to its previous version reuses its cached audio even when an upstream insertion shifts its position; this preserves the audible voice of unchanged content across revisions. Cross-slide artifacts (the global timeline and anything indexed against it) are dropped unconditionally, since any

change in audio duration shifts that timeline. The pipeline then resumes with the surviving cache as a memo table, re-executes only for $\mathcal{S}$, and recomputes the global timeline once at the end. Persisting contracts and change plans across sessions yields an auditable edit history, turning PaperContract into a conversational editor in which authors refine a presentation by editing its contract rather than its pixels.

4 Experiments

In this section, we evaluate our method on paper-to-video generation and analyze its key components.

4.1 Setup

Implementation Details. We use *GPT-4.1* (OpenAI, 2025) for contract construction and interactive revision, and *Gemini-3.1-Flash-Image-Preview* (Google DeepMind, 2026) as the VLM for both slide rendering and judging, balancing efficiency and performance. In QA-guided refinement, the rubric thresholds are set to $\tau^c = 10$ and $\tau^s = 8$, and the contextual-bandit action scorer is updated with a learning rate of 0.08. For temporal realization, narration is synthesized once per slide with deterministic hash-based seeding, after which a single ASR-based forced-alignment pass (Bain et al., 2023) produces the word-level timestamps shared by subtitles, cursor grounding, and talking-head composition. We use UI-TARS (Qin et al., 2025) for cursor grounding and Ditto (Li et al., 2025) for talking-head synthesis. GPT-4.1 and Gemini are accessed through APIs, while local modules run on two NVIDIA A100-SXM4-80GB GPUs.

Baselines. We compare against four systems from two categories: **(i)** *end-to-end generators* (Wan et al., 2025; DeepMind, 2025), in which a native video model produces the presentation directly from a paper-derived prompt; and **(ii)** *multi-agent frameworks* (Shi et al., 2025b; Zheng et al., 2025; Zhu et al., 2025), which generate slides, narration, and presentation assets in separate stages before composing them into a final video.

Benchmark and metrics. We evaluate on the Paper2Video benchmark (Zhu et al., 2025), which contains 100 peer-reviewed conference papers spanning machine learning, computer vision, and natural language processing, each paired with an author-recorded presentation video.[1]

We report seven metrics drawn from two sources. From Paper2Video (Zhu et al., 2025): **Meta Similarity** measures surface-level alignment between generated and reference outputs; **PresentArena** uses a VideoLLM as a proxy audience to perform double-order pairwise comparison against the human reference; **PresentQuiz** assesses how effectively the video conveys the paper's core content via question answering; and **IP Memory** evaluates author visibility and presentation memorability. From PPTEval (Zheng et al., 2025): **Content**, **Design**, and **Coherence** are scored by GPT-4.1, where content and design are averaged across slides and coherence is assessed at the presentation level.

4.2 Main Results

Table 1 reports the overall paper-to-video results. Among all automatic systems, **PaperContract** is the strongest on nearly every metric. Compared with the strongest full-pipeline baseline, $\text{PaperTalk}_{\text{GPT4.1}}$ (Zhu et al., 2025), our method lifts speech and content similarity from 0.646/1.97 to 0.722/2.22 and IP Memory from 50.0% to 72.5%. PresentArena win rate jumps from 17.0% to 43.0%, and PresentQuiz detail/understanding rises from 0.842/0.951 to 0.866/0.960. The largest gain appears on PresentArena, which evaluates the final video as a whole rather than any single channel; this indicates that the contract-centered design improves global presentation quality rather than overfitting one metric.

Two patterns emerge across baseline families. End-to-end generators such as Wan2.2 and Veo3 are fast but struggle to preserve the paper's technical structure, which drags their similarity and arena scores down. Multi-agent systems such as PresentAgent and PaperTalk recover more paper content, yet they still lag on cross-modal alignment and audience preference. Both observations match our hypothesis: when slides, narration, and visual guidance are derived from separate interpretations of the paper, drift accumulates across stages, and a single persistent contract removes that source of error.

Table 2 zooms in on the slide deck before video composition. PaperContract with Gemini Flash is best on all three PPTEVAL dimensions (4.80 / 4.74 / 4.91 on content / design / coherence), with the largest margin on coherence—both PaperContract variants leave PPTAgent and PaperTalk well behind.

[1] We exclude one tutorial paper ("All Things ViTs: Understanding and Interpreting Attention in Vision Transformers") that does not conform to the standard research paper format, leaving 100 papers for evaluation.

Table 1: **Detailed evaluation results across four baselines. Bold** and underline indicate the best and second-best results. NA means the method is not applicable to the metric.

Method	Similarity↑		Arena↑	PresentQuiz Acc.↑		IP Memory↑	Avg. Duration(s)
	Speech	Content		Detail	Under.		
HumanMade	**1.00**	**5.00**	**50.0%**	0.738	0.908	-	375.15
Wan2.2 (Wan et al., 2025)	NA	NA	1.1%	0.251	0.551	11.5%	4.00
Veo3 (DeepMind, 2025)	0.133	NA	1.2%	0.367	0.585	31.3%	8.00
PresentAgent$_{\text{GPT4.1}}$ (Shi et al., 2025b)	0.045	1.47	2.0%	0.548	0.654	12.5%	430.20
PaperTalk$_{\text{GPT4.1}}$ (Zhu et al., 2025)	0.646	1.97	17.0%	0.842	0.951	50.0%	234.36
PaperContract$_{\text{GPT4.1+Gemini Flash}}$	0.722	2.22	43.0%	0.866	0.960	**72.5%**	239.48
+ **Interactive Revision**	0.729	-	-	**0.926**	**0.976**	-	286.91

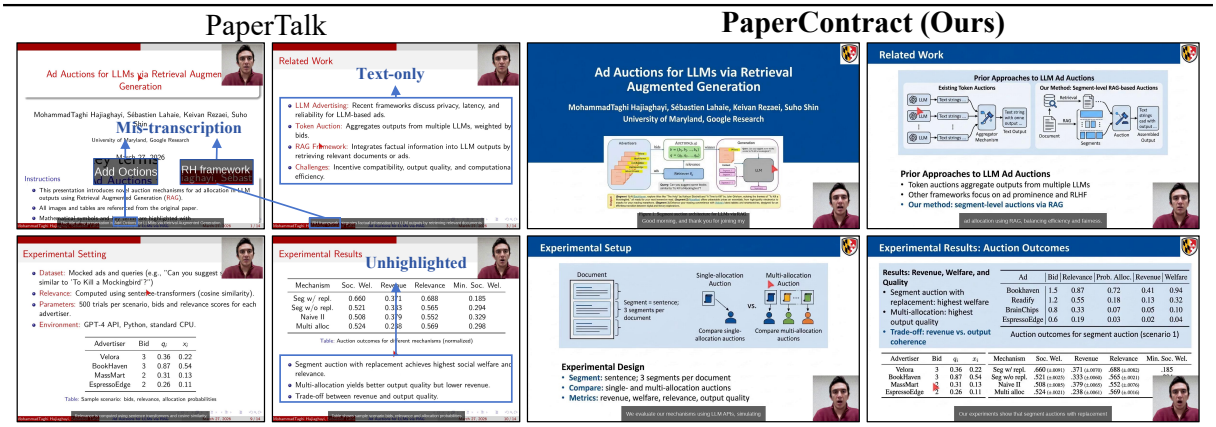

Figure 3: Qualitative comparison between PaperTalk (Zhu et al., 2025) and PaperContract. PaperTalk suffers from subtitle mis-transcription, text-only layouts, and unhighlighted results. PaperContract produces accurate subtitles, visually structured slides with an institution logo, richer visual elements, and emphasized key findings.

Table 2: Evaluation result on slide quality. ⋆ represents a simple version without QA-Guided Refinement.

Method	Content	Design	Coherence
HumanMade	4.43	2.85	2.73
PPTAgent (Zheng et al., 2025)	4.07	2.02	2.06
PaperTalk (Zhu et al., 2025)	4.34	2.85	3.84
PaperContract$_{\text{Python-ppt}}$	3.55	3.70	4.15
PaperContract$^{\star}_{\text{Gemini Flash}}$	**4.80**	4.71	**4.91**
PaperContract$_{\text{Gemini Flash}}$	**4.80**	**4.74**	**4.91**

This is exactly what the contract is meant to deliver: it improves not only slide-level visual quality but also the global narrative consistency of the deck.

Fig. 3 gives a qualitative view on the same input paper. PaperContract produces more accurate subtitles, visually richer slide layouts with institution logos, and clearer emphasis on key findings, whereas PaperTalk shows subtitle mis-transcription, text-only slides, and unhighlighted experimental results.

4.3 Ablation and Efficiency

Table 2 also gives a component-level view of slide generation. Replacing the Python-ppt renderer with Gemini Flash lifts content / design / coherence from

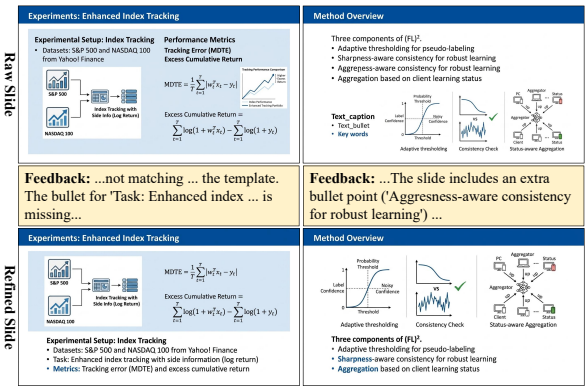

Figure 4: Slide visualization of QA-guided refinement. Guided by the QA feedback, the refined slides correctly recover missing items, remove extraneous bullets, and apply typographic emphasis to key terms.

3.55 / 3.70 / 4.15 to 4.80 / 4.71 / 4.91, showing that an image-native renderer handles dense scientific visuals, equations, and layout constraints far better than template-based composition. Adding QA-guided refinement on top of Gemini Flash mainly tightens design quality (4.71 to 4.74) while leaving the already strong content and coherence scores untouched. This is the behavior we want from the

Table 3: Generation cost for each method. ⋆ represents a simple version without temporal contract.

Method	Token (K)	Time (min.)	Cost ($)
Veo3 (DeepMind, 2025)	NA	0.4	1.667
Wan2.2 (Wan et al., 2025)	NA	8.1	0.280
PresentAgent (Shi et al., 2025b)	241	39.5	0.003
PaperTalk (Zhu et al., 2025)	62	48.1	0.001
PaperContract⋆	84	25.9	1.333
PaperContract	84	18.3	1.333
Interactive Revision	24	8.6	0.274

refinement loop: it acts as a lightweight repair step for layout and formatting, not a rewrite that perturbs slide semantics.

Fig. 4 visualizes this repair behavior on two representative slides. When a raw slide violates the contract specification—either by omitting required bullets or including redundant content—the verification module emits targeted QA feedback and triggers selective regeneration. The refined slides recover the missing items, drop the extraneous bullets, and apply keyword emphasis as specified in the contract, confirming that QA-guided refinement enforces structural correctness without disrupting the overall semantics established by the renderer.

Table 3 examines generation efficiency. Compared with its simplified variant without a temporal contract, the full system cuts average generation time from 25.9 to 18.3 minutes at identical token usage and API cost. This 29.3% runtime saving comes from reusing a shared temporal representation rather than re-aligning each downstream modality. PaperContract is also substantially faster than the multi-agent baselines PresentAgent (39.5 minutes) and PaperTalk (48.1 minutes), suggesting that a shared intermediate representation is useful not only for output quality but also for system efficiency.

4.4 Interactive Revision Analysis

The last row of Table 1 reports one additional round of contract-guided revision driven by PresentQuiz errors. Starting from the base PaperContract output, we feed the incorrectly answered quiz items back as revision prompts and let the model update only the affected contract units. PresentQuiz detail rises from 0.866 to 0.926 and understanding from 0.960 to 0.976, while speech similarity stays essentially flat at 0.729. The performance gain is concentrated in comprehension-oriented metrics (e.g., missing explanations, unclear reasoning, and insufficient slide emphasis), rather than in overall visual style. This variance is consistent with the semantic nature of the user's revision prompts. Note that because the prompts are derived from PresentQuiz failures, the quiz scores measure in-distribution recovery rather than out-of-distribution generalization; we treat this experiment as a proof-of-concept demonstrating that contract-level edits can translate evaluator feedback into targeted improvements, and leave a fully held-out evaluation to future work.

Crucially, this quality gain comes at a small fraction of a full re-run. The bottom row of Table 3 shows that one revision round consumes only 24K tokens, 8.6 minutes, and $0.274 per paper. This is roughly 29% of the tokens, 47% of the wall-clock time, and 21% of the API cost of regenerating the video from scratch (84K, 18.3 min, $1.333). This asymmetry is a direct consequence of selective invalidation: unchanged contract units reuse their cached slides, audio, and talking-head clips, so the marginal cost of an edit scales with the size of the diff rather than the length of the deck.

This result supports the central claim of PaperContract. Once a video is represented as an editable contract, the system stops behaving like a one-shot generator; specific failure cases surfaced by an evaluator can be translated into localized semantic edits that leave the rest of the deck untouched. In practice, this turns paper-to-video generation into an evaluation-driven editing process, where refinement is localized, auditable, and compatible with downstream reuse.

5 Conclusion

In this paper, we presented **PaperContract**, a contract-centered framework for paper-to-video generation. Instead of relying on repeated prompt-based reinterpretation, our approach introduces a single structured representation shared across all presentation modalities. By decoupling content specification from modal realization, PaperContract significantly improves cross-modal consistency while reducing redundant computation. Furthermore, it enables highly efficient post-generation editing through selective, contract-level cache invalidation. Comprehensive experiments demonstrate that PaperContract outperforms existing stage-wise and agent-based baselines in presentation quality, alignment, and editability. Broadly, our findings suggest that persistent structured representations offer a powerful and promising foundation for controllable, reusable multimodal generation.

6 Limitations

PaperContract improves consistency and editability for paper-to-video generation, but it still has several practical limitations. First, the current system depends on a stack of strong external models for contract construction, slide rendering, and evaluation, including frontier LLM/VLM APIs as well as local synthesis modules. As a result, overall performance is partly coupled to model availability, version changes, and inference cost rather than to the contract design alone. This dependency also makes exact reproduction harder across time and compute environments, especially because our reported setup uses API-based models together with GPU-heavy local modules.

Second, our empirical study is still limited in scope. We evaluate on the Paper2Video benchmark with 100 research papers paired with author-recorded talks, which provides a realistic testbed for scientific presentations but does not cover other presentation genres such as tutorials, lectures, product demos, or highly interactive talks. The method also assumes that a paper can be mapped into a mostly linear slide narrative. For papers with unusually dense mathematics, large tables, or figure-heavy layouts, the contract may still omit fine-grained details, and the renderer may produce crowded or suboptimal slides.

Third, the contract representation does not guarantee factual correctness. If the contract construction stage misses an important claim, over-compresses a derivation, or assigns the wrong emphasis to a figure, the resulting error can propagate consistently to slides, narration, subtitles, and cursor behavior. In this sense, the contract can reduce inconsistency without eliminating source-level semantic mistakes. A related limitation is that our evaluation still relies heavily on automated model-based judges, including VLM-based judging and PPTEVAL-style scoring, which may not fully capture human preferences, pedagogical quality, or subtle factual distortions.

Finally, the current revision mechanism is most effective for localized edits, but larger structural changes still incur non-trivial recomputation. In particular, when earlier narration changes alter audio duration, downstream global timeline artifacts must be recomputed because later timestamps shift accordingly. This means that the system is not yet a constant-time editor for arbitrary revisions. Future work should therefore study stronger factual grounding, broader human evaluation, and more fine-grained temporal reuse under large structural edits.

7 Ethical Considerations

Beyond technical limitations, the system also raises potential misuse risks. Because PaperContract can automatically produce polished presentation videos with narration, subtitles, cursor motion, and talking-head clips, factual errors or misleading emphasis in the generated content may be perceived as authoritative and spread more easily in educational or scientific settings. In addition, the talking-head component could be misused to create presenter-like videos without adequate disclosure or consent. These risks suggest that human verification, clear provenance, and explicit disclosure of AI-generated content remain important when deploying such systems in practice.

Our experiments also rely on existing artifacts, including the Paper2Video benchmark, third-party APIs, and pretrained external modules, whose use remains subject to the licenses and terms specified by their respective creators. We use these resources only for research and evaluation in a paper-to-video generation setting that is consistent with their intended role as benchmarking data or general-purpose model/tooling support. Correspondingly, the system and its generated outputs should be treated as research artifacts, and any downstream release or deployment should continue to respect the original access conditions of the underlying data, models, and services.

The benchmark data used in our experiments consists of author-recorded research presentation videos paired with published papers, so any names, voices, faces, affiliations, or other identifying information originate from materials that are already public in the academic communication setting. We do not augment the benchmark with new identity annotations, attempt to infer additional sensitive attributes, or release additional raw personal media beyond what is already contained in the original benchmark. For components that rely on speaker images or reference voices, use should remain limited to research settings with appropriate rights and consent, and any broader deployment should include stronger review of privacy, identity, and content-safety considerations.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman. 2023. Whisperx: Time-accurate speech transcription of long-form audio. *arXiv preprint arXiv:2303.00747*.

Qiguang Chen, Mingda Yang, Libo Qin, Jinhao Liu, Zheng Yan, Jiannan Guan, Dengyun Peng, Yiyan Ji, Hanjing Li, Mengkang Hu, and 1 others. 2025a. Ai4research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. 2025b. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6255–6271.

Jiahao Cui, Hui Li, Yao Yao, Hao Zhu, Hanlin Shang, Kaihui Cheng, Hang Zhou, Siyu Zhu, and Jingdong Wang. 2024. Hallo2: Long-duration and high-resolution audio-driven portrait image animation. *arXiv preprint arXiv:2410.07718*.

DeepMind. 2025. Veo 3 technical report. Technical report, DeepMind. Technical Report.

Tsu-Jui Fu, William Yang Wang, Daniel J. McDuff, and Yale Song. 2021. Doc2ppt: Automatic presentation slides generation from scientific documents. In *AAAI Conference on Artificial Intelligence*.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. 2025. Autopresent: Designing structured visuals from scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2902–2911.

Google DeepMind. 2026. Gemini 3.1 Flash Image Model Card. https://deepmind.google/models/model-cards/gemini-3-1-flash-image/. Model card. Published: 2026-02-26. Accessed: 2026-05-25.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tianqi Li, Ruobing Zheng, Minghui Yang, Jingdong Chen, and Ming Yang. 2025. Ditto: Motion-space diffusion for controllable realtime talking head synthesis. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9704–9713.

Xin Liang, Xiang Zhang, Yiwei Xu, Siqi Sun, and Chenyu You. 2025. Slidegen: Collaborative multimodal agents for scientific slide generation. *Preprint*, arXiv:2512.04529.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. 2025. Showui: One vision-language-action model for gui visual agent. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19498–19508.

Jiacheng Miao, Joe R Davis, Jonathan K Pritchard, and James Zou. 2025. Paper2agent: Reimagining research papers as interactive and reliable ai agents. *arXiv preprint arXiv:2509.06917*.

OpenAI. 2025. GPT-4.1 Model. https://platform.openai.com/docs/models/gpt-4.1. OpenAI API documentation. Accessed: 2026-05-25.

Wei Pang, Kevin Qinghong Lin, Xiangru Jian, Xi He, and Philip Torr. 2025. Paper2poster: Towards multimodal poster automation from scientific papers. *arXiv preprint arXiv:2505.21497*.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, and 1 others. 2025. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*.

Minju Seo, Jinheon Baek, Seongyun Lee, and Sung Ju Hwang. 2025. Paper2code: Automating code generation from scientific papers in machine learning. *arXiv preprint arXiv:2504.17192*.

Jingwei Shi, Zeyu Zhang, Biao Wu, Yanjie Liang, Meng Fang, Ling Chen, and Yang Zhao. 2025a. PresentAgent: Multimodal agent for presentation video generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 760–773, Suzhou, China. Association for Computational Linguistics.

Jingwei Shi, Zeyu Zhang, Biao Wu, Yanjie Liang, Meng Fang, Ling Chen, and Yang Zhao. 2025b. Presentagent: Multimodal agent for presentation video generation. *arXiv preprint arXiv:2507.04036*.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. 2021. D2S: Document-to-slide generation via query-based text

summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1405–1418, Online. Association for Computational Linguistics.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, and 1 others. 2025. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*.

Zhiruo Wang, Daniel Fried, and Graham Neubig. 2024. Trove: Inducing verifiable and efficient toolboxes for solving programmatic tasks. In *International Conference on Machine Learning*.

Wenzheng Zeng, Mingyu Ouyang, Langyuan Cui, and Hwee Tou Ng. 2026. Slidetailor: Personalized presentation slide generation for scientific papers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):34584–34592.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2025. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14413–14429, Suzhou, China. Association for Computational Linguistics.

Hao Zheng, Guozhao Mo, Xinru Yan, Qianhao Yuan, Wenkai Zhang, Xuanang Chen, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. 2026. Deeppresenter: Environment-grounded reflection for agentic presentation generation. *Preprint*, arXiv:2602.22839.

Zeyu Zhu, Kevin Qinghong Lin, and Mike Zheng Shou. 2025. Paper2video: Automatic video generation from scientific papers. *arXiv preprint arXiv:2510.05096*.