arXiv:2607.00407v2 [cs.AI] 12 Aug 2026

Personalization as Inverse Planning: Learning Latent Design Intents for Agentic Slide Generation via Structural Denoising

Tianci Liu[1,*], Zihan Dong[2], Linjun Zhang[2], Haoyu Wang[3], Jing Gao[1], Emre Kıcıman[4], Ranveer Chandra[4], and Wei-Ting Chen[4†]

[1] Purdue University
[2] Rutgers University
[3] University at Albany
[4] Microsoft

Abstract. Slide design requires personalizing both deck themes and page layouts. Yet, current AI agent-based methods struggle with fine-grained, page-level design. Solely relying on prespecified templates or user verbose instructions, they fail to capture latent design intents, leaving Page-level Slide Personalization (PSP) unresolved. To close this gap, this work formulates PSP as an inverse planning problem. We propose to learn a design intent without assuming any knowledge of the specific executing tools (e.g., PowerPoint, Beamer) being used. However, relinquishing control over these tools makes the problem intractable to optimize end-to-end. To overcome this, we propose SPIRE, a principled framework to solve PSP approximately. By intentionally corrupting the visual structures of clean slides, SPIRE creates a verifiable task to denoise the corruption, whereby two agents learn to collaboratively refine executable designs via reinforcement learning (RL). We present a proof that structural denoising is a consistent surrogate for PSP, and that the multi-agent formulation strictly reduces policy gradient variance in RL. Extensive experiments demonstrate the superiority of SPIRE.

1 Introduction

Slide decks are a primary medium for communicating ideas in academia and industry [13, 34, 40]. Yet, creating a high-quality deck is a design-intensive process. Depending on the presenter and the target audience, the *same* content can call for different visual treatments. These treatments are often personalized to a speaker's habitual design, a lab's visual identity, or an organization's brand guidelines. In short, practical slide generation is inherently a *personalized* task.

[*] Work done during an internship at Microsoft.
[†] Corresponding author.

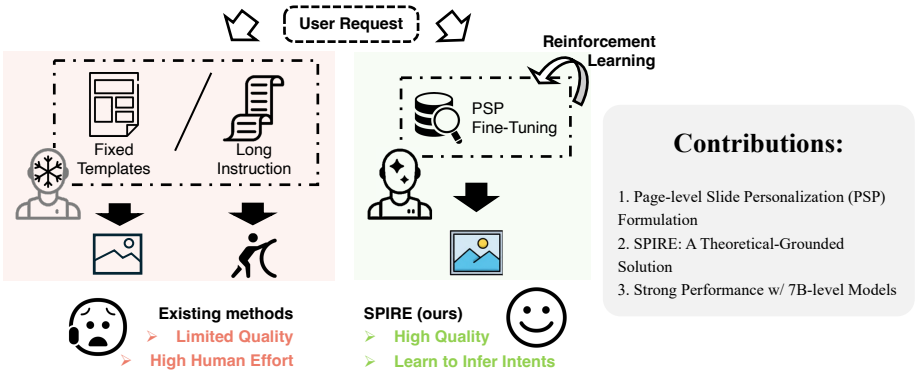


Fig. 1: Illustration of SPIRE. Trained with reinforcement learning, SPIRE learns to infer design intents instead of using prespecified layout templates or lengthy user instructions as existing methods do [10, 53], offering both *theoretical* and *empirical* advantages.

Recent advances in multi-modal large language models (MLLMs) have enabled agentic pipelines that automate slide generation for practical use [9, 10, 26, 30, 38, 45, 53]. These systems typically decompose slide creation into modular stages such as outlining, asset extraction, layout arrangement, and iterative refinement, thereby improving deck-level coherence via template selection, visual feedback, and multi-agent coordination [15, 16, 30, 42, 50, 53]. However, they struggle with fine-grained *page-level* design, which requires deciding visual hierarchy, element alignment, spacing, and styling choices. Nonetheless, existing agentic systems handle page-level layout and styling passively: Once the content is specified, the page-level design is largely overlooked, either by following generic system-defined templates [15, 24, 42, 53], or by requiring lengthy, explicit user instructions [10, 30], instead of inferring the user's latent intent. Consequently, they are too coarse to achieve satisfactory Page-level Slide Personalization (PSP).

To close this gap, we introduce the task of agentic PSP; the concept is depicted in Fig 1. Our formulation is inspired by how human designers teach slide making in practice: *Given* a detailed plan elaborating the page specification such as layout structure and visual hierarchy, a generally capable executor (e.g., an experienced designer unfamiliar with specific corporate styles) can reliably reproduce a high-quality personalized slide visual by following the plan step-by-step.

But such actionable plans are infeasible to collect at scale in practice. Detailed intent annotations are rarely available in real-world slide corpora [10], making direct supervised training impracticable. As a result, PSP requires one to *proactively infer* the plan as a latent intent in order to actively guide the generation process. Furthermore, this plan is naturally context-dependent: (enterprise) users often maintain more than one set of intents depending on their specific scenarios. Fortunately, such contextual intent can be largely specified by having the user provide a small number of reference slides. To this end, we formulate PSP as a probabilistic problem that explicitly models the design intent

as a latent random variable. Given (1) user-provided page assets and (2) a small set of reference slides, we infer the final, detailed plan, which is then seamlessly passed to a downstream visual designer for execution. Intuitively, our formulated PSP aims to infer a plan that, once executed by a capable, non-personalized executor, can most reliably reproduce the desired personalized slide given the user's references. *This probabilistic formulation treats design intent as a latent variable and provides a principled foundation for page-level slide personalization.*

Crucially, this latent-intent formulation of PSP is computationally challenging to solve for two reasons. First, it evaluates a proposed plan based on whether it enables the executor to reproduce the desired visual. This requires a rendering likelihood, which lacks a clear, direct objective that one can optimize. Naive image-level similarity provides an unreliable estimate for slide quality, which is governed by discrete, structured design decisions rather than purely pixel-wise closeness [10, 33, 39, 44]. Second, the executor that turns a plan into a slide is effectively a black box, which further impedes optimizing the PSP objective with standard numerical methods [17,35]. Therefore, an effective approximation is needed to make this objective tractable. We detail this formulation and its computational challenges in Sec. 2.1.

As a remedy, in this work we propose SPIRE (Structural Planning via Inverse REconstruction), a structural denoising framework that turns the hard-to-optimize *slide* personalization objective into a verifiable reconstruction problem that serves as a good proxy. Our solution extends existing multi-agent visual generation frameworks [11, 14, 15, 23, 41] to a principled training objective. With the aforementioned PSP goal of learning a reference-conditioned page-level plan, we alternate between a critic that provides semantic feedback and a planner that updates its plan accordingly. The key idea is to exploit the discrete structural nature of slides: instead of perturbing pixels, we intentionally corrupt a gold slide by applying random, element-level structural perturbations to its layout, hierarchy, and styling, and record the corresponding discrepancies. *This yields scalable self-supervision where suboptimal parts to improve are known by construction.*

Built upon the structural perturbation, SPIRE trains two complementary components separately on this shared signal. A critic learns to read the corrupted slide and the user's references, and to produce structured, actionable feedback that pinpoints the discrepancies as semantic edit suggestions. A planner learns to produce an executable, editable plan, either proposing a plan from scratch or revising an imperfect plan guided by the critic's feedback. As both components are trained to recover from controlled structural corruptions, the critic's feedback becomes verifiable, and the planner learns to improve plans without relying on gradients through the black-box executor, making latent-intent PSP tractable in practice. We resort to reinforcement learning [36,47] to train the two agents. Sec. 2.2 details the proposed SPIRE. We provide theoretical analysis of its superiority in Sec. 2.3. *This principled method and its theoretical advantage are the main technical contribution of this work.*

Our paper is organized as follows. Sec. 2 formulates PSP and details the proposed SPIRE. Extensive experimental results in Sec. 3 demonstrate the effec-

tiveness of our method. In the remaining part of this paper, we review related work in Sec. 4, and conclude the paper in Sec. 5.

2 Proposed Method

This section formalizes the task of reference-based page-level slide personalization (PSP), which entails an intractable optimization problem due to contextual intent being latent. We propose SPIRE as a principled approximate solution.

2.1 Problem Formulation

Suppose a (enterprise) user specifies a high-level instruction x about slide content (e.g., text content and visual elements)[5], and provides K reference slides

$$\mathcal{D}_{\text{ref}} = \{s_{\text{ref}}^{(1)}, \ldots, s_{\text{ref}}^{(K)}\},$$

to exemplify their contextually preferred style. We define PSP as producing a target slide s that can accurately reflect the user's exact intent that is not elaborated in the verbal x. To reflect the need for creative design, we allow $\mathcal{D}_{\text{ref}}$ not to cover the optimal layout s should use.

We refer to this intent as a *plan* and consider it as a latent variable denoted by z. Intuitively, z can be a step-by-step actionable instruction that contains a rich page-level specification (e.g., layout coordinates and element styling).

Following a well-formed z, a generally capable executor E can be unambiguously guided to produce a high quality visual s that is *personalized* to the user's needs, even if E itself is not directly tailored to the user's preference. From a probabilistic perspective, this implies that the visual realization s is *conditionally independent* of the user context given the plan z:

$$p(s \mid z, x, \mathcal{D}_{\text{ref}}) = p(s \mid z).$$

Here the randomness is induced by the use of black-box E, which can be a coding agent [1,6,7,16,27,29,52], an image generator [8,22,28,46], or a human designer. Note that disentangling the plan z and E is on purpose, so that PSP does not need to be conducted whenever a new executor E is introduced.

Based on this definition, letting π_θ denote a multi-modal language model *planner*, PSP can be formulated as learning π_θ from a personal corpus

$$\mathcal{D}_{\text{tr}} = \{(x^{(1)}, (s^*)^{(1)}, \mathcal{D}_{\text{ref}}^{(1)}), \ldots, (x^{(J)}, (s^*)^{(J)}, \mathcal{D}_{\text{ref}}^{(J)})\}.$$

Here s^* denotes the gold slide for x. We detail the construction of $\mathcal{D}_{\text{tr}}$ in the supplementary material.

[5] In this paper, we refer to page-level objects on a slide (e.g., text boxes, images, shapes, charts, and tables) uniformly as *visual elements*. A *text box* is a type of visual element that contains *text content*.

Given $\mathcal{D}_{\text{tr}}$, the goal of PSP is to maximize the marginal likelihood of s^* given $x, \mathcal{D}_{\text{ref}}$. This can be expressed as the following optimization objective:

$$\begin{aligned}\mathcal{J}(\theta) &= \mathbb{E}_{(x,s^*,\mathcal{D}_{\text{ref}})\sim\mathcal{D}_{\text{tr}}}\left[\log p_\theta(s^* \mid x, \mathcal{D}_{\text{ref}})\right] \\ &= \mathbb{E}_{(x,s^*,\mathcal{D}_{\text{ref}})\sim\mathcal{D}_{\text{tr}}}\left[\log \int \pi_\theta(z \mid x, \mathcal{D}_{\text{ref}})\, p(s^* \mid z)\, dz\right]. \end{aligned} \tag{1}$$

At a colloquial level, Eq. (1) seeks π_θ that can generate plans that can maximize the likelihood of black-box E to reproduce (generate) the gold s^*. Unfortunately, optimizing Eq. (1) is prohibitive due to two reasons.

First, the likelihood $p(s^* \mid z)$ lacks an explicit form, which makes the objective intractable. One common solution is to approximately optimize

$$p(s^* \mid z) \propto -\|s^* - E(z)\|_2,$$

in the pixel space, aiming to push generated $s = E(z)$ towards s^* pixel-wise. Yet, recent works [10,39,44] showed that treating slides as pure images struggles to capture their discrete logical structure, offering unreliable guidance for the planner.

Second, black-box E cannot be differentiated through, which prevents direct computation of the gradient for π_θ with respect to $s = E(z)$ given by

$$\nabla_\theta \log p_\theta(s = E(z) \mid x, \mathcal{D}_{\text{ref}}).$$

In addition, zeroth-order optimization methods [4,12,19,25] prove ineffective for this context for two reasons. First, their high computational cost [31,32,51] makes them prohibitive for slide generation, where user preferences and instructions take diverse forms. Second, these methods approximate gradients by probing the specific executor E, which makes them prone to overfitting on a particular instance [12], leading to limited generalizability of π_θ to other executors [32].

In the next section, we show that by leveraging the *structural* discreteness of slide visuals, we can circumvent these hurdles via a surrogate denoising objective, and we propose SPIRE as an effective approximate solution.

2.2 SPIRE: An Effective Solution for PSP

To tackle the computational challenge of Eq. (1), we approximate the two parts therein with two complementary agents based on their intrinsic goals. The two agents are trained to exploit the discrete structural nature of slides via a shared structural perturbation process, turning the problem into a verifiable reconstruction signal through a *denoising* process over the slide's discrete structural space. We dub our method Structural Planning via Inverse REconstruction (SPIRE). Fig 2 depicts the overview of PSP and SPIRE.

We begin by checking the roles of Eq. (2)

$$p_\theta(s^* \mid x, \mathcal{D}_{\text{ref}}) = \int \underbrace{\pi_\theta(z \mid x, \mathcal{D}_{\text{ref}})}_{\text{planner proposal,}} \quad \underbrace{p(s^* \mid z)}_{\text{render likelihood}} \quad dz, \tag{2}$$

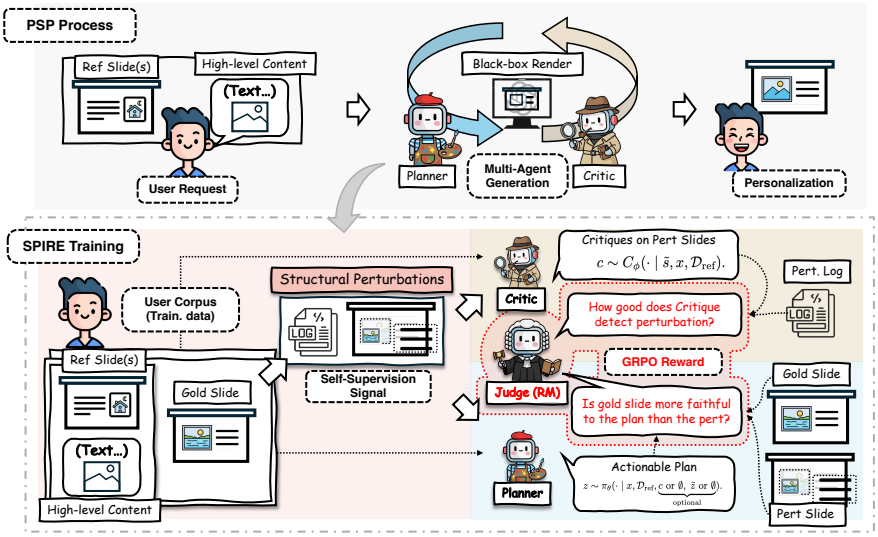


Fig. 2: The overview of Personalized Slide Personalization (PSP) and the proposed SPIRE. The Planner and Critic are trained to recover self-supervised Structural Perturbation in a decomposed way with reinforcement learning (red). After training, they collaborate with a black-box render executor to conduct PSP.

where the integrand consists of two parts. The first term, $\pi_\theta(z \mid x, \mathcal{D}_{\text{ref}})$, denotes how the planner π_θ proposes a plan z to reflect the user's latent intent. Next, the second term $p(s^* \mid z)$ measures the *render* likelihood of the golden slide s^*, depicting the plan's *visual* validity. PSP wants to adjust the planner proposal based on $p(s^* \mid z)$, which, unfortunately, is intractable due to black-box E.

To bypass this challenge, we approximate the update by incorporating a *critic* agent C_ϕ to act as a learned proxy for the likelihood. By learning to *discriminate* how a generation s visually deviates from the gold s^*, the critic captures the user's implicit preferences and provides actionable semantic feedback, thereby providing guidance for the planner to update its proposal $\pi_\theta(z \mid x, \mathcal{D}_{\text{ref}})$ in context $\mathcal{D}_{\text{ref}}$. With a reliable critic, the planner can iteratively improve its proposal: starting from an initial plan, it revises the plan based on the critique.

Yet, reliably learning such a critic, and thus enabling critique-guided planner updates, is also non-trivial. The difficulty is the lack of *verifiable* supervision: for a random generation s, one can hardly evaluate whether a critique is objectively correct, which makes subsequent planner training unreliable as well.

Denoising Signal. To solve this problem, we propose a "denoising" objective through a corruption-reconstruction strategy. Namely, we construct a verifiable self-supervised signal that can be shared across the two agents' training. Specifically, we corrupt the gold slide s^* into a perturbed slide $\tilde{s}$ in a controlled way, and record the ground-truth corruption operations. Next, the critic is asked to criticize $\tilde{s}$ based on visual evidence, and its critique can be verified by checking

the recorded corruption operations. The same corruption record also helps train the planner to perform critique-guided recovery in the *plan space*: it learns to revise a suboptimal plan $\tilde{z}$ using the critique c. We will detail this soon.

Structural Noises. We leverage the discrete structural nature of slides to design *structural corruptions*, rather than adding pixel-level noise to the visuals. Specifically, we represent each slide s^* as a collection of visual elements that incorporate different element types. Then, we apply random perturbation to each visual element by modifying its structural attributes. Each element type admits its own perturbation spaces, e.g., position for layout, size for hierarchy, and color palette for styling. The corruption occurs along three dimensions: *spatial layout* (e.g., shifting coordinates), *visual hierarchy* (e.g., distorting element sizes), and *stylistic coherence* (e.g., altering text colors). To prevent spurious correlations, we independently perturb each randomly selected element.

Formally, assume s^* contains N visual elements $\{e_i\}_{i=1}^N$, and let $\mathcal{T}(e_i)$ denote the element type (e.g., text box, image, shape). We represent the corruption by an element-level action list $\mathcal{A} = \{a_i\}_{i=1}^N$, where each a_i perturbs some attributes of e_i or leaves it unchanged. Perturbation actions are sampled independently, i.e.,

$$q(\mathcal{A} \mid s^*) = q(a_i, \ldots, a_N \mid s^*) = q(a_i, \ldots, a_N \mid e_i, \ldots, e_N) \overset{(a)}{=} \prod_{i=1}^{N} q_{\mathcal{T}(e_i)}(a_i \mid e_i),$$

where (a) holds by independence. Here $q_{\mathcal{T}(e_i)}(\cdot \mid e_i)$ denotes a perturbation distribution over actions for element e_i (admitting no perturbation). Subscript $\mathcal{T}(e_i)$ indicates that action spaces depend on the element types. Having the sampled action list $\mathcal{A}$, we apply these actions to s^* and denote the corrupted slide by $\tilde{s}$. The inverse of each applied perturbation forms a discrepancy list:

$$\mathcal{A}_{\text{diff}} = \{a_i^{-1} \mid a_i \neq \emptyset, 1 \leq i \leq N\},$$

where $\emptyset$ denotes leaving the element unchanged. This yields self-supervised triplets $(\tilde{s}, s^*, \mathcal{A}_{\text{diff}})$, providing scalable supervision without manual annotation. Built upon the structural perturbation, SPIRE factorizes the intractable PSP problem from Eq. (1) into the following two complementary sub-tasks.

Structural Discrimination. This task aims to train critic C_ϕ to identify and articulate the discrepancies in $\mathcal{A}_{\text{diff}}$, which correspond to moving $\tilde{s}$ toward the gold s^* to maximize the *render likelihood* term in Eq. (2). Specifically, the critic generates a critique c based on x, a suboptimal visual $\tilde{s}$, and reference $\mathcal{D}_{\text{ref}}$:

$$c \sim C_\phi(\cdot \mid \tilde{s}, x, \mathcal{D}_{\text{ref}}).$$

Here, c encompasses a list of issues and targeted corrections, translating errors in $\tilde{s}$ into actionable textual feedback. We evaluate c against the full action list $\mathcal{A}$ with a capable LLM judge (GPT-4o-mini). For each element e_i $(1 \leq i \leq N)$, the judge verifies whether c correctly handles that element:

$$y_i = J(c, e_i, a_i) \in \{0, 1\}, \qquad 1 \leq i \leq N.$$

Here $y_i = 1$ indicates a correct decision: the critique identifies and corrects the perturbation when $a_i \neq \emptyset$, or correctly refrains from reporting an issue when $a_i = \emptyset$; and 0 otherwise. Subsequently, we define the element-level verification reward as the average verification accuracy over all N elements:

$$R_{\text{vfy}}(c, \mathcal{A}) = \frac{1}{N} \sum_{i=1}^{N} y_i.$$

We train the critic using DAPO [47], a strong reinforcement learning algorithm, to maximize the expected verification-format reward:

$$\mathcal{J}_{\text{critic}}(\phi) = \mathbb{E}_{\substack{s^* \sim \mathcal{D}_{\text{tr}} \\ \tilde{s} \sim q(\cdot|s^*)}} \Big[\mathbb{E}_{c \sim C_\phi(\cdot|\tilde{s}, x, \mathcal{D}_{\text{ref}})} \big[R_{\text{vfy}}(c, \mathcal{A}) \times R_{\text{format,c}}(c) \big] \Big]. \tag{3}$$

We detail critique generation and format requirement in the supplementary material. Note that our perturbation design, combined with element-level evaluation, mitigates reward hacking. As each perturbation type is applied independently with probability 1/2, trivial always-issue or never-issue critiques cannot consistently score high.

Structural Planning. This task aims to produce actionable plans that translate user instructions into high-quality slides. The goal is to approximately improve the *planner proposal* term in Eq. (2). To this end, we train the planner to (i) generate a good plan proposal, and (ii) update a proposal using the critic's critique as a semantic gradient. Formally, we refer to this goal of the planner as *iterative refinement*, which covers *initial proposal* (generating z from scratch) and *critique-guided refinement* (revising a suboptimal $\tilde{z}$ based on critique c). The two subgoals can be expressed as a unified formulation. Given instruction x, the reference $\mathcal{D}_{\text{ref}}$, and optionally a suboptimal plan $\tilde{z}$ to revise, with its corresponding critique c, the planner aims to generate a high quality plan z as

$$z \sim \pi_\theta(\cdot \mid x, \mathcal{D}_{\text{ref}}, \underbrace{c \text{ or } \emptyset,\ \tilde{z} \text{ or } \emptyset}_{\text{optional}}). \tag{4}$$

The visual quality of a plan z is measured with a discriminative *plan–visual matching reward*. Specifically, a capable VLM judge (Claude Opus 4.5) is given the plan z and a pair of visuals $(s^*, \tilde{s})$, and is asked to choose which one matches the plan z better. We reward z if the judge prefers the gold slide s^*. The judgment is conducted twice with the presenting order of the two visuals swapped to mitigate the positional bias [43]. Given the presenting order $(s^*, \tilde{s})$, we denote the judgment outcome as

$$\hat{o}_{\rightarrow} = \mathbb{I}\Big[s^* \succ \tilde{s} \,\big|\, z, (s^*, \tilde{s}) \Big] \in \{0, 1\},$$

and $\hat{o}_{\leftarrow}$ is defined similarly for the swapped presenting order. Based on this, we define the *plan–visual matching reward* as

$$R_{\text{pvm}}(z, s^*, \tilde{s}) = \frac{1}{2} \Big(\mathbb{I}[\hat{o}_{\rightarrow} = 1] + \mathbb{I}[\hat{o}_{\leftarrow} = 1] \Big)$$

To make R_{pvm} reliable, we instruct the capable judge to base its decision *solely* on fidelity to z rather than its own aesthetic preference. Note that $\tilde{s}$ is produced through structural perturbations that alter *design styles* (e.g., color palette) instead of making the slide visually worse. Without seeing x and $\mathcal{D}_{\text{ref}}$, $\tilde{s}$ may look good on its own; therefore, correct *judgment* requires a discriminative z. As with the critic, we train the planner with DAPO to maximize the expected reward, combined with a format verification term:

$$\mathcal{J}_{\text{plnr}}(\theta) = \mathbb{E}_{\substack{s^* \sim \mathcal{D}_{\text{tr}} \\ \tilde{s} \sim q(\cdot|s^*)}} \Big[\mathbb{E}_{c^* \sim \text{Oracle}(c)} \big[\mathbb{E}_{z \sim \pi_\theta(\cdot|x, \mathcal{D}_{\text{ref}}, c^*)} [R_{\text{pvm}}(z, s^*, \tilde{s}) \times R_{\text{format,p}}(z)] \big] \Big].$$

We defer more details about plan generation and format requirements to the supplementary material. Note that the iterative refinement goal designed for the planner requires high-quality $(c^*, \tilde{z})$ to train the agent so that it can reliably refine a given plan under critique. To this end, we use an oracle VLM (GPT-5) to (i) describe the perturbed slide $\tilde{s}$ as a suboptimal plan $\tilde{z}$, and (ii) translate the discrepancy list $\mathcal{A}_{\text{diff}}$ (provided as *hints* [18,48,49]) into gold critique c^*:

$$\tilde{z} \sim \text{Oracle}(\tilde{z} \mid \tilde{s}), \qquad c^* \sim \text{Oracle}(c \mid \mathcal{A}_{\text{diff}}).$$

2.3 Theoretical Analysis

We end up this section with the theoretical advantages of SPIRE. Full versions are deferred to the supplementary material due to page limit.

First, under regular conditions, optimizing the SPIRE objective gives a gradient-level approximation to that of the original PSP objective up to an explicit error bound, as formalized in the following Thm 1.

Theorem 1 (Surrogate Validity for PSP, informal). *Let $\widehat{\mathcal{J}}_{SP}(\theta, \phi)$ be the empirical loss of SPIRE. Then there exists $\epsilon_{\text{tot}} \geq 0$ such that*

$$\left\| \nabla_\theta \widehat{\mathcal{J}}_{SP}(\theta, \phi) - \frac{\beta}{4} \nabla_\theta \mathcal{J}(\theta) \right\| \leq \epsilon_{\text{tot}}.$$

Here ϵ_{tot} aggregates critic-calibration, structural-surrogate, variational-gap, and empirical optimization errors. Full definitions are provided in the supplementary material.

Crucially, the key error term is controlled thanks to our critic training on structural corruptions with verifiable ground truth supervision. An untrained or pixel-denoising-trained C would not satisfy this bound in general.

Furthermore, Thm 2 below proves that SPIREtrains the planner and critic *separately*, thereby eliminating executor-induced noise.

Theorem 2 (Two-agent Decomposition Stabilizes Optimization, informal). *Let $\hat{g}_{e2e}$ and $\hat{g}_{2a}$ denote the end-to-end and two-agent policy-gradient estimators, respectively; see the supplementary material for explicit definitions and derivation. Then*

$$\text{Var}(\hat{g}_{e2e}) = \text{Var}(\hat{g}_{2a}) + \Delta_{exec|c}, \quad \Delta_{exec|c} \geq 0.$$

So $\mathrm{Var}(\hat{g}_{2a}) \leq \mathrm{Var}(\hat{g}_{e2e})$*, with strict inequality for non-zero executor-induced noise. See formal statement, expression, and imperfect-critic extension in the supplementary material.*

In general, even if a planner-executor-critic design is adopted, standard training of the planner and critic still requires the executor to generate visuals for complete rollouts, retaining its induced noise. SPIRE provides a way to bypass the executor in our decomposed training, thereby enabling the noise reduction.

3 Experiment

We evaluate the performance of SPIRE and ablate its components' contributions. Benefiting from the principled optimization presented before, SPIRE offers strong PSP capability over strong baselines that rely on much larger GPT models.

3.1 Dataset and Experiment Settings

Datasets. We build the **train** and **test** data from Zenodo10k [53] by randomly sampling 200 decks. For each deck, we perform a slide-level held-out split that preserves the deck's chronological order. The last 20% of slides are reserved for testing and the remaining slides are used for training (or for retrieval for training-free baselines). SlideBench [10] is used as **out-of-distribution** test data. We defer more data preparation details in the supplementary material.

Backbones. Both the critic and the planner are fine-tuned from Qwen2.5-VL-7B-Instruct [2]. At inference time, the planner and critic collaborate with a black-box executor E to perform multi-round iterative generation. We use GPT-o4-mini as the coding executor to implement the plan with `python-pptx`, and follow [10, 39] to improve coding reliability. No template is used in execution.

Baselines. We compare SPIRE against representative systems that cover both slide-native pipelines and generic visual synthesis. We include AutoPresent [10], which generates a slide by deciding layout and style for the given instruction from scratch on its own. We also include PPTAgent [53], which selects a template/style from the reference slides and re-applies it to the target content through an edit-based workflow. Finally, following [10], we include a generic image generation baseline using Stable Diffusion 3.5 [8] equipped with IP-Adapter [46] which prompt the model to synthesize a slide-like image. To isolate the benefit of our training, we additionally report PSP results with planner/critic setting to: (i) frontier VLM (o4-mini) and (ii) the corresponding base model (without fine-tuning), while keeping the rest of the pipeline unchanged.

Evaluations. We evaluate the generation in terms of visual similarity against the gold slides, and reference-free quality. Following [10, 21, 39], we adopt both visual-similarity metrics and a VLM-as-a-Judge protocol. Specifically, for visual similarity, we treat the slides as rendered images and utilize metrics such as SSIM and CLIP to measure "reconstruction" quality. For VLM-as-a-judge evaluation, we follow [10, 21, 54] and evaluate the generation quality from the perspectives of *faithfulness*, *color*, *layout*, and *overall aesthetics*. We detail these in the supplementary material.

Table 1: Quantitative results on *test* and *OOD* pages. Best average results are in **bold** and the second best results are underlined.

Model	Visual Similarity			VLM-as-Judge				
	Sim_{ssim} ↑	Sim_{clip} ↑	**AVG**	Faith ↑	Color ↑	Layout ↑	Aest ↑	**AVG**
Test Pages								
GPT-based Models								
AutoPresent [10]	0.6062	0.4431	0.5247	0.6174	0.5850	0.4145	0.4105	0.5069
PPTAgent [53]	0.5126	0.5491	0.5309	0.1036	0.5670	0.4364	0.3961	0.3758
PSP (o4-mini)	0.8440	0.7683	**0.8062**	0.5504	0.5763	0.4206	0.3661	0.4784
7B-level Models								
SD 3.5 [8]	0.3372	0.4496	0.3934	0.1559	0.4282	0.2545	0.1889	0.2569
PSP (Base)	0.3578	0.3317	0.3448	0.2995	0.4588	0.2956	0.2382	0.3230
SPIRE (Ours)	0.8134	0.7018	0.7576	0.7072	0.6330	0.4470	0.3787	**0.5415**
OOD Pages								
GPT-based Models								
AutoPresent [10]	0.5976	0.5892	0.5934	0.7923	0.7242	0.5477	0.5201	0.6461
PPTAgent [53]	0.4836	0.6371	0.5604	0.0839	0.5824	0.4392	0.3831	0.3722
PSP (o4-mini)	0.7233	0.7633	**0.7433**	0.6975	0.5947	0.4472	0.3958	0.5338
7B-level Models								
SD 3.5 [8]	0.3177	0.5753	0.4465	0.1811	0.5075	0.2934	0.2246	0.3016
PSP (Base)	0.3910	0.3864	0.3887	0.2432	0.2309	0.1716	0.1292	0.1937
SPIRE (Ours)	0.6785	0.6954	0.6870	0.9330	0.7545	0.6565	0.5891	**0.7333**

3.2 Quantitative Results

Tab 1 shows quantitative results on test and out-of-distribution (OOD) pages. We note the following observations.

Good PSP Performance. From the table, SPIRE achieves the strongest overall performance, outperforming the GPT-based AutoPresent (0.5069) and PSP (o4-mini) (0.4784) in terms of judge score, while maintaining a highly competitive visual similarity score (0.7414), substantially higher than the 7B-level baselines SD 3.5 and PSP (Base). This performance is notable given that SPIRE relies on only two 7B-scale agents, yet achieves competitive performance compared to GPT-based components. We also note empirical failures in the baselines, attributed to their underlying mechanisms. PPTAgent struggles with faithfulness (0.1036), as realistic reference sets rarely offer perfect templates without structural loss[6]. AutoPresent falls short when the verbose prompting is too coarse. *These trends clearly support our PSP formulation.* The suboptimal results of the training-free PSP baselines highlight the necessity of preference-aligned optimization; without it, the critic leverages its own generic preferences rather than reflecting the user's contextual needs. *This confirms that SPIRE's gains stem from the RL training, not merely the multi-agent refinement loop.*

[6] In reality, `pptx` template files may not be accessible.

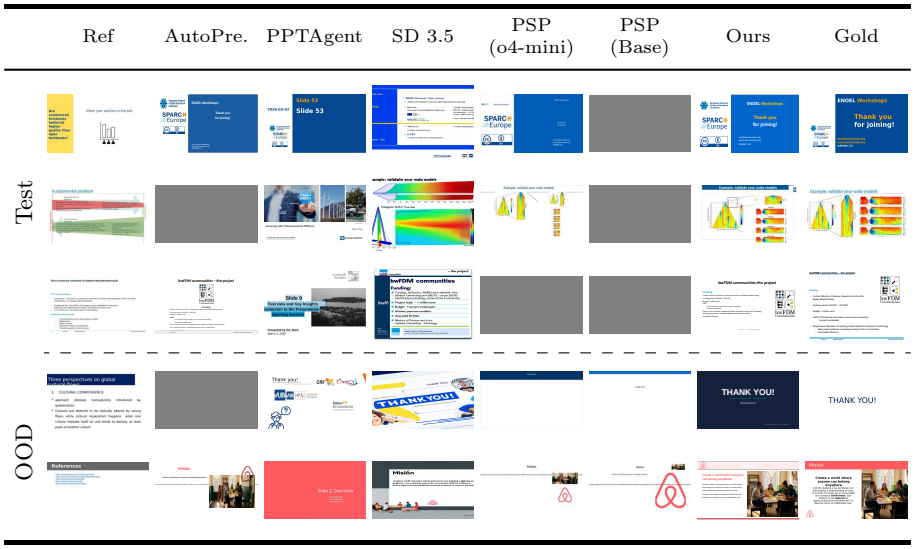


Fig. 3: Qualitative comparison of slide generation results.

Strong Generalizability. On OOD pages, SPIRE demonstrates strong generalizability and achieves the highest judge average, substantially outperforming baselines such as AutoPresent, PSP (o4-mini), PPTAgent, SD 3.5, and PSP (Base). Our method obtains the best scores across all judging dimensions and the second-best visual similarity. These results indicate that SPIRE does not simply memorize deck-specific patterns from the training corpus, but instead successfully leverages the reference slides as contextual evidence at inference time, allowing it to infer the appropriate design intent in unseen scenarios.

Metric Discrepancy. We also note that visual similarity and judge-based scores are not perfectly aligned. For instance, while PSP (o4-mini) achieves the highest visual averages across both in-distribution and OOD settings, its judge-based quality remains substantially lower than that of SPIRE. This mismatch, as explained in Sec. 2, highlights the challenge of delivering PSP by optimizing standardized metrics alone—a limitation also noted in the literature [10, 33, 44].

These results collectively demonstrate the effectiveness of the proposed SPIRE.

3.3 Qualitative Results

We provide a visual comparison of the generated slides across different methods. Illustrative results are shown in Fig 3, with more examples deferred to the supplementary material. We note that SPIRE consistently produces visually appealing and structurally coherent slides that closely align with the gold slides, while baselines often struggle to maintain visual fidelity or fail to generate valid structures.

Table 2: Ablation study on SPIRE Training and Iterative Revision. Both components are necessary for high-quality generation.

Model	Visual Similarity			VLM-as-Judge				
	Sim_{ssim} ↑	Sim_{clip} ↑	**AVG**	Faith ↑	Color ↑	Layout ↑	Aest ↑	**AVG**
w/o FT	0.3578	0.3317	0.3448	0.2995	0.4588	0.2956	0.2382	0.3230
w/o Revision	0.7299	0.6096	0.6698	0.4865	0.5120	0.3386	0.2779	0.4038
SPIRE	0.8134	0.7018	0.7576	0.7072	0.6330	0.4470	0.3787	0.5415

Table 3: Ablation study on the critic role. Substituting the larger, untrained GPT critic with SPIRE-trained 7B model results in improved performance.

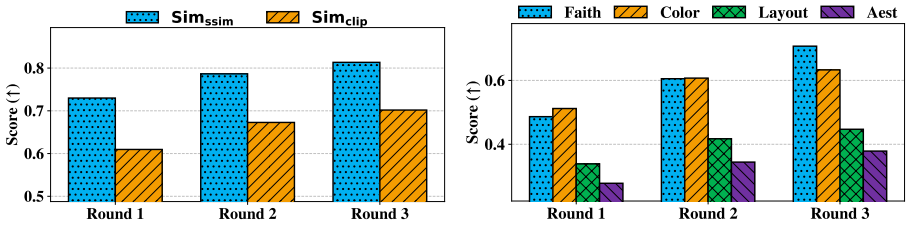
Model	Visual Similarity			VLM-as-Judge				
	Sim_{ssim} ↑	Sim_{clip} ↑	**AVG**	Faith ↑	Color ↑	Layout ↑	Aest ↑	**AVG**
w/ o4-mini Critic	0.8440	0.7683	0.8062	0.5504	0.5763	0.4206	0.3661	0.4784
w/ SPIRE Critic	0.9367	0.8633	0.9000	0.5777	0.5905	0.4515	0.3942	0.5035

The effectiveness of SPIRE, stemming from our preference-aligned RL training, empowers the model to explicitly infer latent user intent. This capability manifests in two aspects. First, SPIRE achieves superior color coherence. For instance, as shown in Row 4 and 5, SPIRE accurately *infers* a proper coloring strategy that differs from the reference slide in OOD scenarios. Second, SPIRE demonstrates strong spatial reasoning for layout arrangement. In Row 2, it successfully organizes complex, multi-image visual assets (e.g., heatmaps) into a clean, aligned structure that matches the gold slide perfectly. In Row 3, it also correctly positions the logo and text blocks. This proactive inference capability allows SPIRE to surpass baselines solely relying on verbose instructions (AutoPresent) or generic templates (PPTAgent). When such explicit inputs are unavailable, these methods inevitably fail to capture the user's true intent.

3.4 Ablation Study

We end up this section with ablation study on key components in SPIRE.

SPIRE Training and Iterative Revision. To break down the contributions of our proposed framework, we ablate the fine-tuning process and the multi-turn revision mechanism, with results reported in Tab. 2. First, we observe that preference-aligned fine-tuning is necessary for PSP. Namely, when directly prompting base Qwen models to conduct PSP, they struggle significantly and achieve a visual average of only 0.3344 and a judge average of 0.3230. In contrast, our fine-tuned SPIRE achieves much better scores, reaching 0.7414 and 0.5415 respectively. This massive performance gap confirms that off-the-shelf VLMs, especially small-scale models, cannot reliably infer latent page-level design intents without our targeted structural denoising optimization. Second, the results validate the effectiveness of the iterative revision process. As shown in the table,



(a) Visual similarity. **(b)** Judge scores.

Fig. 4: Ablation on SPIRE revision. Multi-round revision in SPIRE helps improve both visual similarity and judge scores, indicating better generation qualities.

and Fig. 4, the visual similarity scores exhibit a clear and steady improvement through the successive refinement rounds. This steady climb is a direct result of our structural denoising objective, which explicitly trains the planner to interpret and execute structural corrections based on the critic's feedback. Consequently, the planner is capable of progressively refining suboptimal layouts, proving that SPIRE successfully leverages high-quality critiques to iteratively enhance the design rather than relying on a single-pass generation.

Critic Role. One hypothesis stated in Sec. 3.2 is that the suboptimal performance of PSP lies in its critic. To verify that the preference-aligned training helps the critic learn the user's contextual needs better, we replace the critic in the PSP (o4-mini) baseline with our 7B-level SPIRE-trained critic and report the results on test slides. Results are reported in Tab 3. From the table, we see that this substitution yields consistent improvements across all metrics. Remarkably, the critic is a 7B-level model, which is much less capable than GPT models. This gain confirms that such a targeted critic indeed helps guide the PSP process.

4 Related Work

Early works consider automated slide generation as a text summarization task [3, 9, 38], aiming to extract [9, 38] and summarize [3] proper deck-level content from a given document for presentation. These solutions boiled down to extracting salient sentences, figures, and sections from source documents to organize them into slide outlines. However, they overlooked the necessity of coherent layout design and the inherent multimodal nature of slides [20, 50]. As a result, these solutions offered limited control over the aesthetic perspective of the generated slides, leaving personalized generation untouched.

Following the emergence of (M)LLMs, recent research has shifted toward end-to-end and agentic pipelines for slide design [10, 15, 16, 26, 30, 42, 50, 53]. Representative solutions decompose the task into modular stages, including content outlining, asset extraction, layout arrangement, and iterative refinement. These systems improve visual consistency through template-based selection, visual feedback, and multi-agent coordination [16, 26, 30, 42, 50, 53]. Nevertheless,

these solutions mainly focus on more global deck- or document-level decisions and planning, lacking more fine-grained page-level layout design capabilities. For such page-level design, these solutions are either guided by generic templates and system-defined objectives [50, 53], or by explicit and lengthy instructions specified by users [10, 30]. These designs inherently limit their performance for PSP, which requires deep understanding of user-specific design intents.

Very recent works have explored personalization for slide generation. However, existing works mostly focus on adapting content and narrative structure to different audiences or presentation contexts [16,50,53]. [26] enables more tailored communication by conditioning on audience profiles or example pairs, and [50] allows users to specify a document–slide deck pair to express their preference, with the system aiming to replicate reference slides while plugging in content from the user's own document. However, these designs still lack the ability to create novel page-level designs tailored to the user's intent. In this work, we propose SPIRE to learn a user's page-level preferences from their visual data, pushing agentic slide generation to a more fine-grained page-level design task.

5 Conclusion

This paper studies Page-level Slide Personalization (PSP), a key aspect of agentic slide generation that remains largely underexplored. Broadly, we show that personalized visual generation is fundamentally a latent-intent inference problem. We formulate PSP as an inverse planning problem to infer a user's latent design intent, which can be used to guide diverse executors to render the visuals. We propose SPIRE, which leverages structural denoising to construct a tractable surrogate objective for optimization. By intentionally corrupting the visual structures of gold slides, SPIRE creates a self-supervised denoising task whereby two agents are trained to iteratively refine the design plan. We provide theoretical analysis on how structural denoising serves as a consistent surrogate objective for PSP, and how our multi-agent formulation strictly reduces policy gradient variance to stabilize the optimization. Extensive experiments demonstrate the effectiveness of our method against representative baselines for the PSP task.

Acknowledgements

This work is supported in part by the US National Science Foundation under grant NSF IIS-2141037. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Anthropic: Claude sonnet 4: Hybrid reasoning model with superior intelligence for high-volume use cases, and 200k context window. `https://www.anthropic.com/claude/sonnet` (2025)

2. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025)
3. Cao, J., Zhang, X., Li, R., Wei, J., Li, C., Joty, S., Carenini, G.: Multi2: Multi-agent test-time scalable framework for multi-document processing. In: Proceedings of The 5th New Frontiers in Summarization Workshop. pp. 135–156 (2025)
4. Chen, L., Chen, J., Goldstein, T., Huang, H., Zhou, T.: Instructzero: Efficient instruction optimization for black-box large language models. arXiv preprint arXiv:2306.03082 (2023)
5. Chen, Z., Liu, G., Zhang, B.W., Yang, Q., Wu, L.: Altclip: Altering the language encoder in clip for extended language capabilities. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 8666–8682 (2023)
6. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
7. Dong, Y., Jiang, X., Qian, J., Wang, T., Zhang, K., Jin, Z., Li, G.: A survey on code generation with llm-based agents. arXiv preprint arXiv:2508.00083 (2025)
8. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
9. Fu, T.J., Wang, W.Y., McDuff, D., Song, Y.: Doc2ppt: Automatic presentation slides generation from scientific documents. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 634–642 (2022)
10. Ge, J., Wang, Z.Z., Zhou, X., Peng, Y.H., Subramanian, S., Tan, Q., Sap, M., Suhr, A., Fried, D., Neubig, G., et al.: Autopresent: Designing structured visuals from scratch. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2902–2911 (2025)
11. 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. arXiv preprint arXiv:2412.06771 (2024)
12. Hu, W., Shu, Y., Yu, Z., Wu, Z., Lin, X., Dai, Z., Ng, S.K., Low, B.K.H.: Localized zeroth-order prompt optimization. Advances in Neural Information Processing Systems **37**, 86309–86345 (2024)
13. Hu, Y., Wan, X.: Ppsgen: Learning to generate presentation slides for academic papers. In: IJCAI. pp. 2099–2105 (2013)
14. Jaiswal, S., Prabhudesai, M., Bhardwaj, N., Qin, Z., Zadeh, A., Li, C., Fragkiadaki, K., Pathak, D.: Iterative refinement improves compositional image generation. arXiv preprint arXiv:2601.15286 (2026)
15. 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. arXiv preprint arXiv:2602.13318 (2026)
16. Jung, K., Cho, H., Yun, J., Yang, S., Jang, J., Choo, J.: Talk to your slides: Language-driven agents for efficient slide editing. arXiv preprint arXiv:2505.11604 (2025)
17. Lan, G.: First-order and stochastic optimization methods for machine learning, vol. 1. Springer (2020)

18. Li, C., Xue, M., Zhang, Z., Yang, J., Zhang, B., Yu, B., Hui, B., Lin, J., Wang, X., Liu, D.: Start: Self-taught reasoner with tools. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 13523–13564 (2025)
19. Li, W., Wang, X., Li, W., Jin, B.: A survey of automatic prompt engineering: An optimization perspective. arXiv preprint arXiv:2502.11560 (2025)
20. Liang, X., Zhang, X., Xu, Y., Sun, S., You, C.: Slidegen: Collaborative multimodal agents for scientific slide generation. arXiv preprint arXiv:2512.04529 (2025)
21. Liu, C., Yang, Y., Zhou, K., Zhang, Z., Fan, Y., Xie, Y., Qi, P., Wang, X.E.: Presenting a paper is an art: Self-improvement aesthetic agents for academic presentations. arXiv preprint arXiv:2510.05571 (2025)
22. Ma, J., Liang, J., Chen, C., Lu, H.: Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. In: ACM SIGGRAPH 2024 Conference Papers. pp. 1–12 (2024)
23. Ma, S., Guo, Y., Su, J., Huang, Q., Zhou, Z., Wang, Y.: Talk2image: A multi-agent system for multi-turn image generation and editing. arXiv preprint arXiv:2508.06916 (2025)
24. Maheshwari, H., Bandyopadhyay, S., Garimella, A., Natarajan, A.: Presentations are not always linear! gnn meets llm for document-to-presentation transformation with attribution. arXiv preprint arXiv:2405.13095 (2024)
25. Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J.D., Chen, D., Arora, S.: Fine-tuning language models with just forward passes. Advances in Neural Information Processing Systems **36**, 53038–53075 (2023)
26. Mondal, I., Shwetha, S., Natarajan, A., Garimella, A., Bandyopadhyay, S., Boyd-Graber, J.: Presentations by the humans and for the humans: Harnessing llms for generating persona-aware slides from documents. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2664–2684 (2024)
27. OpenAI: Introducing chatgpt (2022), `https://openai.com/blog/chatgpt`
28. OpenAI: Gpt-image-1. `https://platform.openai.com/docs/models/gpt-image-1` (2025)
29. OpenAI: Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/` (2025)
30. Pang, W., Lin, K.Q., Jian, X., He, X., Torr, P.: Paper2poster: Towards multimodal poster automation from scientific papers. arXiv preprint arXiv:2505.21497 (2025)
31. Park, S., Jeong, J., Kim, Y., Lee, J., Lee, N.: Zip: An efficient zeroth-order prompt tuning for black-box vision-language models. arXiv preprint arXiv:2504.06838 (2025)
32. Qi, Y., Tian, J., Liu, T., Li, R., Wei, T., Liu, H., Tang, X., Cheng, M., He, J.: Learning to instruct: Fine-tuning a task-aware instruction optimizer for black-box llms. In: Findings of the Association for Computational Linguistics: EMNLP 2025. pp. 7707–7733 (2025)
33. Qu, Y., Fang, S., Wang, Y., Wang, X., Chen, Z., Xie, H., Zhang, Y.: Igd: Instructional graphic design with multimodal layer generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18218–18228 (2025)
34. Rojo, M.G., García, G.B., Mateos, C.P., García, J.G., Vicente, M.C.: Critical comparison of 31 commercially available digital slide systems in pathology. International journal of surgical pathology **14**(4), 285–305 (2006)
35. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)

36. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300 (2024)
37. Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., Wu, C.: Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256 (2024)
38. Sun, E., Hou, Y., Wang, D., Zhang, Y., Wang, N.X.: 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. pp. 1405–1418 (2021)
39. Tang, W., Xiao, J., Jiang, W., Xiao, X., Wang, Y., Tang, X., Li, Q., Ma, Y., Liu, J., Tang, S., et al.: Slidecoder: Layout-aware rag-enhanced hierarchical slide generation from design. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 9026–9050 (2025)
40. Trelease, R.B.: From chalkboard, slides, and to e-learning: How computing technologies have transformed anatomical sciences education. Anatomical sciences education **9**(6), 583–602 (2016)
41. Venkatesh, K., Dunlop, C., Yanardag, P.: Crea: A collaborative multi-agent framework for creative image editing and generation. arXiv preprint arXiv:2504.05306 (2025)
42. Xie, E., Waterfield, D., Kennedy, M., Zhang, A.: Slidebot: A multi-agent framework for generating informative, reliable, multi-modal presentations. arXiv preprint arXiv:2511.09804 (2025)
43. Xu, R., Liu, T., Dong, Z., You, T., Hong, I., Yang, C., Zhang, L., Zhao, T., Wang, H.: Alternating reinforcement learning for rubric-based reward modeling in non-verifiable llm post-training. arXiv preprint arXiv:2602.01511 (2026)
44. Xu, X., Xu, X., Chen, S., Chen, H., Zhang, F., Chen, Y.C.: Pregenie: An agentic framework for high-quality visual presentation generation. arXiv preprint arXiv:2505.21660 (2025)
45. Xu, Y., Ma, X., Qiu, J., Zhao, H.: Textual-to-visual iterative self-verification for slide generation. arXiv preprint arXiv:2502.15412 (2025)
46. Ye, H., Zhang, J., Liu, S., Han, X., Yang, W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. arXiv preprint arXiv:2308.06721 (2023)
47. Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)
48. Zelikman, E., Harik, G., Shao, Y., Jayasiri, V., Haber, N., Goodman, N.D.: Quiet-star: Language models can teach themselves to think before speaking. arXiv preprint arXiv:2403.09629 (2024)
49. Zelikman, E., Wu, Y., Mu, J., Goodman, N.: Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems **35**, 15476–15488 (2022)
50. Zeng, W., Ouyang, M., Cui, L., Ng, H.T.: Slidetailor: Personalized presentation slide generation for scientific papers. arXiv preprint arXiv:2512.20292 (2025)
51. Zhan, H., Chen, C., Ding, T., Li, Z., Sun, R.: Unlocking black-box prompt tuning efficiency via zeroth-order optimization. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 14825–14838 (2024)
52. Zhang, K., Li, J., Li, G., Shi, X., Jin, Z.: Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In:

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 13643–13658 (2024)
53. Zheng, H., Guan, X., Kong, H., Zhang, W., Zheng, J., Zhou, W., Lin, H., Lu, Y., Han, X., Sun, L.: Pptagent: Generating and evaluating presentations beyond text-to-slides. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 14413–14429 (2025)
54. Zhu, D., Meng, R., Song, Y., Wei, X., Li, S., Pfister, T., Yoon, J.: Paperbanana: Automating academic illustration for ai scientists. arXiv preprint arXiv:2601.23265 (2026)

Supplementary Material of Personalization as Inverse Planning: Learning Latent Design Intents for Agentic Slide Generation via Structural Denoising

Tianci Liu[1,*], Zihan Dong[2], Linjun Zhang[2], Haoyu Wang[3], Jing Gao[1], Emre Kıcıman[4], Ranveer Chandra[4], and Wei-Ting Chen[4]

[1] Purdue University
[2] Rutgers University
[3] University at Albany
[4] Microsoft

A Theoretical Analysis Details

This section provides formal statements and proofs for the two informal theorems in the main text. We only keep the theorem-specific assumptions where each theorem is stated.

A.1 Preliminaries and Notation

Each training instance is $(x, s^*, \mathcal{D}_{\text{ref}}) \sim \mathcal{D}_{\text{tr}}$. The planner samples a latent plan $z \sim \pi_\theta(\cdot \mid x, \mathcal{D}_{\text{ref}})$. Unless explicitly stated otherwise, $\|\cdot\|_2$ denotes ℓ_2 norm on vectors. For clarity, we use three objectives throughout this section:

- $\mathcal{J}(\theta)$: the original PSP objective in Equation (1):

$$\mathcal{J}(\theta) = \mathbb{E}_{(x,s^*,\mathcal{D}_{\text{ref}})\sim\mathcal{D}_{\text{tr}}}\left[\log \int \pi_\theta(z \mid x, \mathcal{D}_{\text{ref}})\, p(s^* \mid z)\, dz\right].$$

- $\mathcal{J}_{\text{LB}}(\theta)$: the Jensen lower bound of $\mathcal{J}(\theta)$ (i.e., $\mathcal{J}(\theta) \geq \mathcal{J}_{\text{LB}}(\theta)$):

$$\mathcal{J}_{\text{LB}}(\theta) := \mathbb{E}_{(x,s^*,\mathcal{D}_{\text{ref}})}\mathbb{E}_{z\sim\pi_\theta(\cdot|x,\mathcal{D}_{\text{ref}})}\left[\log p(s^* \mid z)\right]. \tag{1}$$

- $\mathcal{J}_{\text{SP}}(\theta, \phi)$: the structural-perturbation surrogate optimized in training:

$$\mathcal{J}_{\text{SP}}(\theta,\phi) := \mathbb{E}_{(x,s^*,\mathcal{D}_{\text{ref}})}\mathbb{E}_{\tilde{s}\sim q_\rho(\cdot|s^*)}\mathbb{E}_{z\sim\pi_\theta(\cdot|x,\mathcal{D}_{\text{ref}})}\left[\widehat{R}_\phi(z,\tilde{s})\right], \tag{2}$$

  where $\widehat{R}_\phi$ is the learned critic-induced reward.

[*] Work done during an internship at Microsoft.

For structural perturbation, sample $\tilde{s} \sim q_\rho(\cdot \mid s^*)$. Define

$$m(z, \tilde{s}) := \log p(s^* \mid z) - \log p(\tilde{s} \mid z), \qquad R^*(z, \tilde{s}) := \sigma(\beta\, m(z, \tilde{s})). \tag{3}$$

Here ρ controls perturbation strength, $\beta > 0$, and $\sigma(\cdot)$ is sigmoid.

In practice, we optimize its empirical estimator

$$\widehat{\mathcal{J}}_{\mathrm{SP}}(\theta, \phi),$$

which is the empirical loss implemented by the critic and planner objectives in the main text.

For variance analysis with critique conditioning, define

$$H(z, c; x) := \nabla_\theta \log \pi_\theta(z \mid x, \mathcal{D}_{\mathrm{ref}}, c). \tag{4}$$

A.2 Details of Informal Theorem 1 (Surrogate Consistency)

Assumption 1 (A1 (Regularity)) *(a)* $|\beta\, m(z, \tilde{s})| \le M$ *for sampled* $(z, \tilde{s})$. *(b)* $\left\| \nabla_\theta \mathbb{E}_{(x, s^*, \mathcal{D}_{ref})} \mathbb{E}_{\tilde{s} \sim q_\rho} \mathbb{E}_{z \sim \pi_\theta} [\log p(\tilde{s} \mid z)] \right\|_2 \le \eta$.

Assumption 2 (A2 (Critic-calibration gradient bound))

$$\| \nabla_\theta \mathcal{J}_{SP}(\theta, \phi) - \nabla_\theta \mathcal{J}_{R^*}(\theta) \|_2 \le \delta_{\mathrm{cal}},$$

where $\mathcal{J}_{R^*}(\theta) := \mathbb{E}_{(x, s^*, \mathcal{D}_{ref})} \mathbb{E}_{\tilde{s} \sim q_\rho} \mathbb{E}_{z \sim \pi_\theta} [R^*(z, \tilde{s})]$.

Assumption 3 (A3 (Gradient gap bounds)) *Under the joint sampling path*

$$(x, s^*, \mathcal{D}_{ref}) \sim \mathcal{D}_{tr}, \qquad \tilde{s} \sim q_\rho(\cdot \mid s^*),$$
$$c \sim C_\phi(\cdot \mid \tilde{s}, x, \mathcal{D}_{ref}), \qquad z \sim \pi_\theta(\cdot \mid x, \mathcal{D}_{ref}, c),$$

the empirical objective $\widehat{\mathcal{J}}_{SP}$ *is the Monte Carlo estimator of* $\mathcal{J}_{SP}$*, and the following gradient gaps hold:*

$$\| \nabla_\theta \mathcal{J}_{LB}(\theta) - \nabla_\theta \mathcal{J}(\theta) \|_2 \le \epsilon_{\mathrm{vi}}, \qquad \left\| \nabla_\theta \widehat{\mathcal{J}}_{SP}(\theta, \phi) - \nabla_\theta \mathcal{J}_{SP}(\theta, \phi) \right\|_2 \le \epsilon_{\mathrm{emp}}.$$

Theorem 4 (Surrogate Approximation Bound). *Under A1–A3,*

$$\left\| \nabla_\theta \widehat{\mathcal{J}}_{SP}(\theta, \phi) - \frac{\beta}{4} \nabla_\theta \mathcal{J}(\theta) \right\|_2 \le \epsilon_{\mathrm{emp}} + \delta_{\mathrm{cal}} + \frac{\beta}{4} \eta + \beta L_\sigma(M) M\, B_m + \frac{\beta}{4} \epsilon_{\mathrm{vi}}, \tag{5}$$

where

$$B_m := \mathbb{E}[\|\nabla_\theta m(z, \tilde{s})\|_2], \qquad L_\sigma(M) := \sup_{|u| \le M} |\sigma''(u)|.$$

Consequently, defining the aggregate error

$$\epsilon_{\mathrm{tot}} := \underbrace{\epsilon_{\mathrm{emp}}}_{\textit{empirical opt.}} + \underbrace{\delta_{\mathrm{cal}}}_{\textit{critic-calibration}} + \underbrace{\tfrac{\beta}{4}\eta + \beta L_\sigma(M) M B_m}_{\textit{structural-surrogate}} + \underbrace{\tfrac{\beta}{4}\epsilon_{\mathrm{vi}}}_{\textit{variational-gap}},$$

we have $\left\| \nabla_\theta \widehat{\mathcal{J}}_{SP}(\theta, \phi) - \frac{\beta}{4} \nabla_\theta \mathcal{J}(\theta) \right\|_2 \le \epsilon_{\mathrm{tot}}$*, which is the formal version of Informal Theorem 1 in the main text.*

Proof. We bound the total error in three steps via the intermediate objectives $\mathcal{J}_{R^*}$ and $\mathcal{J}_{\mathrm{LB}}$.

Step 1 ($\mathcal{J}_{\mathrm{SP}}$ to $\mathcal{J}_{\mathrm{LB}}$). By triangle inequality,

$$\left\|\nabla_\theta \mathcal{J}_{\mathrm{SP}} - \tfrac{\beta}{4}\nabla_\theta \mathcal{J}_{\mathrm{LB}}\right\|_2 \le \underbrace{\|\nabla_\theta \mathcal{J}_{\mathrm{SP}} - \nabla_\theta \mathcal{J}_{R^*}\|_2}_{\le \delta_{\mathrm{cal}}\ \text{(A2)}} + \left\|\nabla_\theta \mathcal{J}_{R^*} - \tfrac{\beta}{4}\nabla_\theta \mathcal{J}_{\mathrm{LB}}\right\|_2.$$

For the second term, since $R^* = \sigma(\beta m)$ and $\sigma'(0) = \frac{1}{4}$, a further triangle inequality gives

$$\left\|\nabla_\theta \mathcal{J}_{R^*} - \tfrac{\beta}{4}\nabla_\theta \mathcal{J}_{\mathrm{LB}}\right\|_2 \le \left\|\nabla_\theta \mathcal{J}_{R^*} - \tfrac{\beta}{4}\nabla_\theta \mathbb{E}[m]\right\|_2 + \tfrac{\beta}{4}\left\|\nabla_\theta \mathbb{E}[m] - \nabla_\theta \mathcal{J}_{\mathrm{LB}}\right\|_2.$$

By mean-value theorem with A1(a) ($|\beta m| \le M$):

$$\left\|\nabla_\theta \mathcal{J}_{R^*} - \tfrac{\beta}{4}\nabla_\theta \mathbb{E}[m]\right\|_2 \le \beta L_\sigma(M) M\, B_m. \tag{6}$$

Since $m = \log p(s^* \mid z) - \log p(\tilde{s} \mid z)$, we have $\mathbb{E}[m] = \mathcal{J}_{\mathrm{LB}}(\theta) - \mathbb{E}[\log p(\tilde{s} \mid z)]$, so by A1(b):

$$\|\nabla_\theta \mathbb{E}[m] - \nabla_\theta \mathcal{J}_{\mathrm{LB}}\|_2 \le \eta. \tag{7}$$

Combining: $\|\nabla_\theta \mathcal{J}_{\mathrm{SP}} - \frac{\beta}{4}\nabla_\theta \mathcal{J}_{\mathrm{LB}}\|_2 \le \delta_{\mathrm{cal}} + \beta L_\sigma(M) M\, B_m + \frac{\beta}{4}\eta$.

Step 2 ($\mathcal{J}_{\mathrm{LB}}$ to $\mathcal{J}$ via variational identity). For each $(x, s^*, \mathcal{D}_{\mathrm{ref}})$, letting $p_\theta \propto \pi_\theta\, p(s^* \mid z)$:

$$\mathcal{J}(\theta) = \mathcal{J}_{\mathrm{LB}}(\theta) + \mathbb{E}[\mathrm{KL}(\pi_\theta \parallel p_\theta)],$$

so $\mathcal{J}(\theta) \ge \mathcal{J}_{\mathrm{LB}}(\theta)$; by A3: $\|\nabla_\theta \mathcal{J}_{\mathrm{LB}}(\theta) - \nabla_\theta \mathcal{J}(\theta)\|_2 \le \epsilon_{\mathrm{vi}}$.

Step 3 (Empirical gap and conclusion). By A3 and triangle inequality,

$$\left\|\nabla_\theta \widehat{\mathcal{J}}_{\mathrm{SP}}(\theta, \phi) - \tfrac{\beta}{4}\nabla_\theta \mathcal{J}(\theta)\right\|_2 \le \epsilon_{\mathrm{emp}} + \left\|\nabla_\theta \mathcal{J}_{\mathrm{SP}}(\theta, \phi) - \tfrac{\beta}{4}\nabla_\theta \mathcal{J}_{\mathrm{LB}}(\theta)\right\|_2 + \tfrac{\beta}{4}\epsilon_{\mathrm{vi}}.$$

Substituting Step 1 yields (5).

A.3 Details of Informal Theorem 2 (Variance Reduction)

We analyze planner gradient variance with critique-conditioned training. For the critique-conditioned policy objective

$$\mathcal{J}_{\mathrm{pg}}(\theta) := \mathbb{E}_{z \sim \pi_\theta(\cdot \mid x, \mathcal{D}_{\mathrm{ref}}, c),\, r \sim p(r \mid z, c, x)}[r],$$

the REINFORCE identity gives

$$\nabla_\theta \mathcal{J}_{\mathrm{pg}}(\theta) = \mathbb{E}[H(z, c; x)\,(r - b(x, c))],$$

which motivates the end-to-end estimator below; replacing r by $\bar{r}(z, c, x) = \mathbb{E}[r \mid z, c, x]$ gives the two-agent estimator. Let

$$z \sim \pi_\theta(\cdot \mid x, \mathcal{D}_{\mathrm{ref}}, c), \quad H(z, c; x) := \nabla_\theta \log \pi_\theta(z \mid x, \mathcal{D}_{\mathrm{ref}}, c),$$

and let $r \sim p(r \mid z, c, x)$ denote reward obtained through black-box execution.

Define end-to-end estimator

$$\hat{g}_{\mathrm{e2e}} := H(z, c; x)\,(r - b(x, c)). \tag{8}$$

Define two-agent conditional-mean estimator

$$\bar{r}(z, c, x) := \mathbb{E}[r \mid z, c, x], \qquad \hat{g}_{\mathrm{2a}} := H(z, c; x)\,(\bar{r}(z, c, x) - b(x, c)). \tag{9}$$

Assumptions for Theorem 2.

Assumption 5 (B1 (Finite second moments)) $\mathbb{E}[\|H(z, c; x)\|_2^2] < \infty$ *and* $\mathbb{E}[r^2 \mid z, c, x] < \infty$.

Assumption 6 (B2 (Baseline independence)) $b(x, c)$ *is independent of* r *conditioned on* (x, c).

Theorem 7 (Variance Decomposition). *Under B1–B2,*

$$\mathrm{Var}(\hat{g}_{e2e}) = \mathrm{Var}(\hat{g}_{2a}) + \Delta_{exec|c}, \tag{10}$$

where

$$\Delta_{exec|c} := \mathbb{E}\big[\|H(z, c; x)\|_2^2 \,\mathrm{Var}(r \mid z, c, x)\big] \geq 0. \tag{11}$$

Hence $\mathrm{Var}(\hat{g}_{2a}) \leq \mathrm{Var}(\hat{g}_{e2e})$*, with strict inequality whenever* $\mathrm{Var}(r \mid z, c, x) > 0$ *on a set of non-zero measure. This is the formal version of Informal Theorem 2 in the main text.*

Proof. Apply law of total variance conditioning on (z, c):

$$\mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y \mid z, c)] + \mathrm{Var}(\mathbb{E}[Y \mid z, c]).$$

Take $Y = \hat{g}_{\mathrm{e2e}}$. Conditioned on (z, c), only r is random:

$$\mathrm{Var}(\hat{g}_{\mathrm{e2e}} \mid z, c, x) = \|H(z, c; x)\|_2^2 \,\mathrm{Var}(r \mid z, c, x).$$

Taking expectation gives $\Delta_{\mathrm{exec}|c}$. Also,

$$\mathbb{E}[\hat{g}_{\mathrm{e2e}} \mid z, c, x] = H(z, c; x)\,(\bar{r}(z, c, x) - b(x, c)) = \hat{g}_{\mathrm{2a}}.$$

Thus the second term is $\mathrm{Var}(\hat{g}_{\mathrm{2a}})$, proving (10).

Interpretation. End-to-end optimization carries executor-induced randomness directly into policy gradient. The two-agent decomposition conditions on critique signal and replaces stochastic return with a lower-noise conditional target, removing the additive variance term in (11).

B More Technical Details

B.1 Details about reference-based PSP Data

We present more details about data curation.

Instruction-Target-Reference Triplets. We detail the construction of the data triplets. Let's use the training data $\mathcal{D}_{\text{tr}}$ as an example. By definition, each data point in

$$\mathcal{D}_{\text{tr}} = \{(x^{(1)}, (s^*)^{(1)}, \mathcal{D}_{\text{ref}}^{(1)}), \ldots, (x^{(J)}, (s^*)^{(J)}, \mathcal{D}_{\text{ref}}^{(J)})\}$$

is a triplet consisting of three components: (i) a high-level instruction x, (ii) a gold slide page s^* to be generated, and (iii) a reference set $\mathcal{D}_{\text{ref}}$. In practice, one may only have access to raw slide decks from users. Nevertheless, these triplets can be readily constructed from such decks as follows:

- **Gold Slide Page.** Each individual slide page s within the training split of deck is treated as a target page s^* to be generated.
- **High-level Instruction.** We employ a capable VLM (e.g., GPT-4o) to provide a *concise* summary of the slide page s^*, e.g., naming the text boxes and visual elements. Note that this summary captures only the factual content without describing visual styles such as coloring, font size, or layout organization, thereby serving the role of the high-level instruction x. Notably, these omitted visual styles represent the latent design intent that must be inferred from the reference set for successful PSP.
- **Reference Set.** We randomly pair each s^* with a few other pages from the same deck to serve as the $\mathcal{D}_{\text{ref}}$, leveraging the inherent visual coherence within a single deck. In our experiments, we set the size of the reference set to 1. We use a single reference slide to create a challenging personalization setting while keeping the retrieval protocol consistent across methods.

This procedure transforms raw slide decks into the structured training corpus $\mathcal{D}_{\text{tr}}$. $\mathcal{D}_{\text{te}}$ is constructed similarly.

Train, Test, and OOD Data. As presented in Sec. 3, we first perform a slide-level held-out split that preserves the deck's chronological order. The last 20% of slides are reserved for testing, and the remaining slides are used for training (or for retrieval in training-free baselines).

Next, we construct target–reference pairs within each split to ensure that test slides are never leaked during training. To determine the feasibility of each pair, we follow [39] to compute the visual complexity scores for both target and reference slides, filtering out pairs where the target–reference complexity score gap is excessive. When constructing these pairs, we consider any combination of two slides within a split to be viable, regardless of their original chronological order. Consequently, for a filtered split containing K pages, the resulting dataset size is $K(K-1)$.

To better reflect realistic PSP requirements, we construct OOD data using a more challenging protocol. These OOD decks are never seen during training; starting from the second page, each slide is paired with its *immediately preceding* page. This setup mimics the sequential nature of real-world slide creation. No complexity filtering is applied to the OOD set. Therefore, for a raw OOD deck of K pages, the final evaluation set is of size $K-1$.

B.2 Training Details of the Critic

We provide additional training details for the critic agent.

Critic Prompt. We prompt the critic to act as a presentation design expert responsible for evaluating a generated slide against the provided user instruction x and reference slide(s) $\mathcal{D}_{\text{ref}}$. The critique generation is structured into three stages. We provide the prompt template in the end of the supplementary material.

- **Analysis.** The critic first writes a brief high-level analysis summarizing the major design inconsistencies.
- **Assessment.** It then assesses the generated slide along eight predefined structural aspects subject to random perturbations: *graphic color, graphic position, graphic size, image position, image size, text color, text position*, and *text size*. See Supp. D for further details on the perturbation mechanism.
- **Feedback.** Finally, the critic outputs a single structured block containing aspect-wise feedback for all eight aspects in a fixed order. Each entry includes the current issue and a target correction. To ensure the feedback is directly usable for downstream refinement, the prompt requires concise and actionable suggestions, with explicit numeric targets in normalized coordinates whenever position or size adjustments are necessary.

Format Reward. We employ a rule-based format reward $R_{\text{format,c}}(c)$ to encourage the critic to follow the required multi-stage output schema. The verification checks whether each stage is properly produced, including the *overall* structure as well as the specific requirements for the *analysis*, *assessment*, and final aspect-wise *feedback* sections. The verification checks: (i) compliance with the required format by identifying keyword tags and ensuring the content is non-empty; and (ii) whether the names of aspect-level entries are valid and non-duplicated.

Mathematically, the format reward is a weighted sum of these components:

$$\begin{aligned} R_{\text{format, c}}(c) = 0.05\, R_{\text{structure}}(c) + 0.05\, R_{\text{analysis}}(c) \\ + 0.05\, R_{\text{assessment}}(c) + 0.85\, R_{\text{feedback}}(c). \end{aligned}$$

We set $R_{\text{feedback}}(c)$ as the dominant term, reflecting that the structured aspect-wise feedback is the most critical part of the critic's output.

Accuracy Reward. The accuracy reward, as introduced in Sec. 2.2, verifies whether the extracted critique is consistent with the ground-truth discrepancy list $\mathcal{A}_{\text{diff}}$. Specifically, for each discrepancy item in $\mathcal{A}_{\text{diff}}$, we check whether the critic correctly diagnoses both the issue and the corresponding correction. This verification is implemented via a rule-based keyword matching procedure between the *extracted* critique and the perturbation annotations. We compute the

final reward by aggregating the matched issue–correction keywords over all discrepancy items:

$$R_{\text{vfy}}(c, \mathcal{A}_{\text{diff}}) = \frac{\#\text{ matched issue–correction keywords in } (c, \mathcal{A}_{\text{diff}})}{\#\text{ all ground-truth issue–correction keywords in } \mathcal{A}_{\text{diff}}}. \quad (12)$$

Essentially, the accuracy reward measures the precision with which the critique covers the issue–correction pairs specified by $\mathcal{A}_{\text{diff}}$.

This reward, combined with the format reward, constitutes the final reward for critic training defined in Eq. (3).

B.3 Training Details of the Planner

We provide additional training details for the planner agent.

Planner Prompt. We prompt the planner to act as an expert slide design planner responsible for producing a detailed and executable design plan conditioned on the provided user instruction x and reference slide(s) $\mathcal{D}_{\text{ref}}$. As detailed in Sec. 2.2, the planner aims to perform two complementary subtasks: (i) to generate a high-quality initial plan based on the user instruction and reference slides; and (ii) to revise a suboptimal plan if a critique is provided. We use separate prompt templates for these two scenarios, both of which are structured into three stages. We provide the prompt template at the end of the supplementary material.

Initial Plan Generation.

- **Analysis.** The planner first infers transferable design principles from the reference slide(s), including the core visual logic, reusable stylistic elements, and context-specific factors that should be adapted to the current request.
- **Strategy.** It then formulates an adaptation strategy that maps these principles into concrete planning decisions, including the overall layout structure, the roles of major visual elements, and the aspect-level choices governing color, position, size, and typographic hierarchy.
- **Plan.** Finally, the planner outputs a complete design plan that specifies the slide composition in a sequential and directly executable manner, covering the background, major containers, textual components, visual elements, and overall spacing/balance. To ensure executability, the prompt requires concrete normalized spatial specifications and prohibits ambiguous instructions.

Critique-Guided Plan Revision.

- **Analysis.** The planner first infers transferable design principles from the reference slide(s), mimicking a chain-of-thought process.
- **Strategy.** It then revises the suboptimal plan according to the provided critique c. In particular, the planner first normalizes the previous plan into an explicit structured representation, and then applies the required edits while preserving aspects that are already satisfactory.

– **Plan.** Finally, the planner outputs a new complete design plan that retains the overall structure of the suboptimal plan whenever possible, while incorporating the requested corrections in a directly executable form.

Both initial and revised plans are evaluated from *format* and *accuracy* perspectives in a unified way, as detailed below.

Format Reward. We employ a rule-based format reward $R_{\text{format},p}(z)$ to encourage the planner to follow the required multi-stage output schema. The verification checks whether each stage is properly produced, including the *overall* structure as well as the specific requirements for the *analysis*, *strategy*, and final *plan* sections. The verification checks: (i) compliance with the required format by identifying keyword tags and ensuring the content is non-empty; and (ii) whether the final plan preserves the required asset content, ensuring no critical user-provided material is omitted.

Mathematically, the format reward is a weighted sum of these components:

$$\begin{aligned} R_{\text{format},p}(z) = 0.05\, R_{\text{structure}}(z) + 0.05\, R_{\text{analysis}}(z) \\ + 0.10\, R_{\text{strategy}}(z) + 0.80\, R_{\text{plan}}(z). \end{aligned}$$

We set $R_{\text{plan}}(z)$ as the dominant term, reflecting that the final executable design plan is the most critical part of the planner's output.

Accuracy Reward. The accuracy reward, as introduced in Sec. 2.2, verifies whether the planner's output is consistent with the target slide under the plan–visual matching (PVM) objective. Specifically, we extract the natural-language design plan from the final *plan* stage and evaluate whether it matches the gold slide s^* more faithfully than a randomly perturbed slide $\tilde{s}$. To mitigate positional bias, the comparison is conducted under both visual presentation orders, and the final reward is computed by averaging the two binary outcomes:

$$R_{\text{pvm}}(z, s^*, \tilde{s}) = \frac{1}{2}\left(\mathbb{I}[\hat{o}_{\rightarrow} = 1] + \mathbb{I}[\hat{o}_{\leftarrow} = 1]\right). \tag{13}$$

Essentially, the accuracy reward measures whether the final plan describes the gold slide more faithfully than the perturbed alternative, based on the judgment of a capable VLM judge (e.g., o4-mini).

This reward, combined with the format reward, constitutes the final reward for planner training defined as

$$\mathcal{J}_{\text{plnr}}(\theta) = \mathbb{E}_{\substack{s^* \sim \mathcal{D}_{\text{tr}} \\ \tilde{s} \sim q(\cdot|s^*)}} \Big[\mathbb{E}_{c^* \sim \text{Oracle}(c)} \big[\mathbb{E}_{z \sim \pi_\theta(\cdot|x, \mathcal{D}_{\text{ref}}, c^*)} [R_{\text{pvm}}(z, s^*, \tilde{s}) \times R_{\text{format,p}}(z)] \big] \Big].$$

Synthesis of Critique-Guided Plan Revision Data. As detailed in Sec. 2.2, critique-guided revision training relies on high-quality revision triplets $(\tilde{z}, \tilde{s}, c^*)$. Here, $\tilde{s}$ is a perturbed slide, $\tilde{z}$ is a suboptimal plan aligned with $\tilde{s}$, and c^* is an oracle critique that specifies how the suboptimal plan should be revised toward the gold slide s^*. The construction process is summarized as follows:

Table 1: Hyper-parameters used for critic and planner training.

Parameter	Critic	Planner
Policy model	Qwen2.5-VL-7B-Instruct	Qwen2.5-VL-7B-Instruct
RL algorithm	DAPO	DAPO
Number of epochs	1	1
Batch size	4	4
Learning rate	1×10^{-6}	1×10^{-6}
Maximum prompt length	6000	6000
Maximum response length	2048	2048
Rollout numbers	4	4
Lower clipping ratio	0.2	0.2
Upper clipping ratio	0.28	0.28
Clipping coefficient	10.0	10.0
Entropy coefficient	0	0
KL regularization coefficient	0	0

- **Perturbed Slide.** The perturbed slide $\tilde{s}$ is obtained by applying recorded structural perturbations to the gold slide s^*, as described in Sec. 2.2. Since these perturbations are known and explicitly recorded, $\tilde{s}$ serves as a controllable visual target corresponding to the suboptimal state.
- **Oracle Critique.** To construct the oracle critique c^*, we provide the perturbation action list to a capable LLM and ask it to translate the known discrepancies into the required critique format. This converts the recorded structural corruptions into natural-language feedback that is directly consumable by the planner. We further apply rejection filtering using the critique reward introduced in Supp B.2, retaining only those pass this verification step.
- **Suboptimal Plan.** To construct the suboptimal plan $\tilde{z}$, we provide the perturbed slide $\tilde{s}$ to a capable LLM and ask it to infer a complete design plan that accurately *reflects the current degraded slide* rather than the gold slide. We then apply rejection filtering based on the planner reward and retain only those candidate plans where the plan–visual matching reward prefers $\tilde{s}$ over s^*, thereby enforcing consistency between the inferred plan and the perturbed visual input.

Overall, this pipeline ensures coherent revision supervision: $\tilde{s}$ provides the observable suboptimal slide, $\tilde{z}$ captures the corresponding suboptimal design intent, and c^* specifies the corrections needed to recover the intended gold design.

B.4 Implementation and Evaluation Details of Experiments

Implementation Details. We train SPIRE with `verl` [37]; the detailed training hyper-parameters are provided in Tab 1.

Evaluation Details (Visual Similarity). For the visual similarity evaluation, we treat each slide as a rendered full-page image. We report two full-reference image similarity metrics, following [39]. The scores are normalized to $[0, 1]$.

- **Structural Similarity Index Measure (SSIM).** SSIM measures the low-level structural fidelity between the generated slide and the gold slide.
- **CLIP Image Similarity.** CLIP compares the global visual representations of two images in a pre-trained vision-language embedding space. We use AltCLIP [5].

Evaluation Details (VLM-as-a-Judge). We also report a reference-free VLM-as-a-judge evaluation protocol. In this setting, the VLM judge observes the user instruction x and the generated slide to assess the generation quality across the following four aspects. Each aspect is scored on a 0–10 scale and then normalized to $[0, 1]$ for aggregation.

- **Faithfulness.** Whether the textual content requested in the user instruction is fully preserved and readable, without missing text, content truncation, or severe element overlap.
- **Color.** The harmony of the color palette, and the contrast between foreground and background elements to ensure readability.
- **Layout.** The overall composition of the slide, including alignment, spacing, margins, and the logical organization of major visual elements.
- **Overall Aesthetics.** The overall professionalism and visual appeal of the slide as a presentation-grade page.

We provide the full VLM-as-a-judge prompt at the end of the supplementary material.

C Additional Experiment Results

We present more experiment results that are not included in the main body due to page limits.

Reference-based Pairwise VLM-Judge Evaluation. To complement the reference-free evaluation, we conduct a reference-based pairwise comparison where the VLM judge is provided with the user instruction x, the gold slide s^*, and two generated slides from competing methods. This protocol directly measures relative generation quality against the intended target, making it more sensitive to subtle differences in design execution.

We evaluate each pair along four aspects: (i) *faithfulness* (text preservation and readability); (ii) *color*; (iii) *layout*; and (iv) *overall aesthetics*. The definitions of these aspects are the same as in the reference-free setting presented above. To mitigate positional bias, we evaluate each pair twice by swapping the presentation order and aggregate the results.

Tab 2 reports the *win/tie rate* of SPIRE, representing the fraction of cases where SPIRE is judged *at least as good as the baseline*. As shown in Tab 2, SPIRE consistently outperforms all baselines on both test and OOD pages. The advantages are particularly pronounced in Layout and Faithfulness, where SPIRE

Table 2: Reference-based Pairwise VLM Evaluation (Win/Tie Rate of SPIRE).

Versus	VLM-Judge Preference (Win/Tie Rate of SPIRE)				
	Faith ↑	Color ↑	Layout ↑	Aest ↑	**AVG**
Test Pages					
AutoPresent [10]	0.812	0.660	0.757	0.727	0.739
PPTAgent [53]	0.981	0.921	0.972	0.966	0.960
SD 3.5 [8]	0.932	0.881	0.939	0.902	0.914
PSP (o4-mini)	0.899	0.832	0.801	0.856	0.847
PSP (Base)	0.995	0.859	0.961	0.951	0.942
OOD Pages					
AutoPresent [10]	0.896	0.761	0.791	0.763	0.803
PPTAgent [53]	0.983	0.931	0.934	0.925	0.943
SD 3.5 [8]	0.941	0.899	0.898	0.854	0.898
PSP (o4-mini)	0.911	0.832	0.851	0.809	0.851
PSP (Base)	0.951	0.903	0.911	0.887	0.913

achieves *win/tie rates* exceeding 90% against most baselines. Even against the strong GPT-based PSP (o4-mini) baseline, SPIRE maintains a significant lead (avg. 0.847/0.851), confirming that our RL-based training produces slides that are consistently preferred in head-to-head comparisons.

C.1 More Qualitative Results

Fig 1 presents additional visual comparisons between SPIRE and baseline methods across both in-distribution test slides and OOD scenarios. Consistent with the observations in Sec. 3.3, SPIRE demonstrates superior faithfulness to the user instruction while maintaining high design consistency with the reference slides.

D More Details about Perturbation

We construct perturbations using a structured taxonomy defined along two axes: **shape role** and **visual attribute**. Each slide element is first assigned one of three shape roles, namely *text*, *graphic*, or *image*, according to its functional role in the slide. Conditioned on this role assignment, we define a fixed set of perturbable attributes, resulting in **eight perturbation categories**.

D.1 Perturbation Categories

Text-related categories. For text elements, we define three perturbation categories:

- **Text Size** (`text_size`): perturbs the font size of a text portion.
- **Text Position** (`text_position`): perturbs text layout by replacing the paragraph alignment with a different valid alignment.
- **Text Color** (`text_color`): perturbs the font color of a text element.

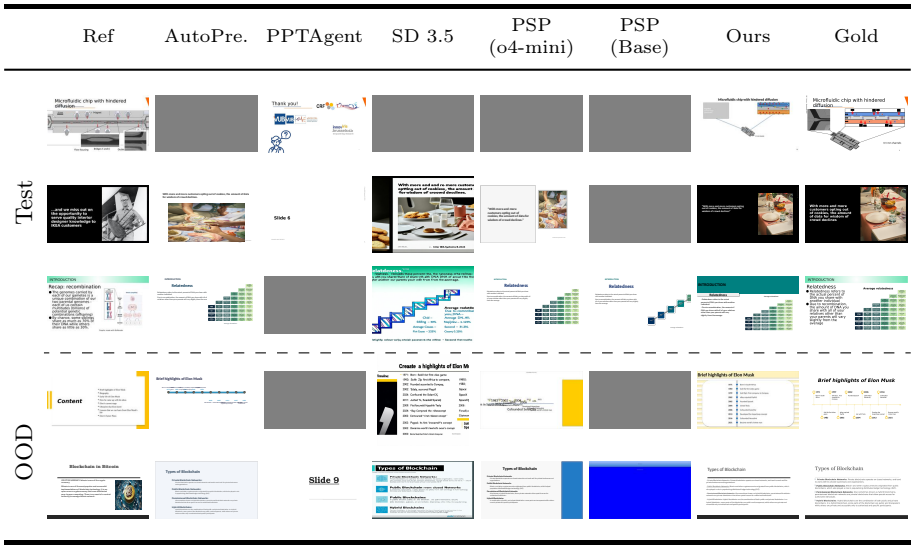


Fig. 1: Additional qualitative comparison of slide generation results.

Graphic-related categories. For non-image graphic shapes, we define three perturbation categories:

- **Graphic Size** (`graphic_size`): perturbs one size attribute of a graphic element, namely width or height.
- **Graphic Position** (`graphic_position`): perturbs one spatial coordinate of a graphic element, namely its horizontal coordinate x or vertical coordinate y, where x and y denote the horizontal and vertical positions of the element on the slide canvas, respectively.
- **Graphic Color** (`graphic_color`): perturbs the fill color of a graphic element.

Image-related categories. For image elements, we define two perturbation categories:

- **Image Size** (`image_size`): perturbs the width or height of an image element.
- **Image Position** (`image_position`): perturbs the horizontal coordinate x or vertical coordinate y of an image element.

D.2 Perturbation Magnitude

For numerical attributes, we control perturbation strength with a discrete severity level. Let $a \in \mathbb{R}$ denote the original value of a numerical attribute and let a' denote the perturbed value. We sample a relative perturbation ratio δ as

$$\delta = s \cdot u, \qquad u \sim \mathcal{U}(\delta_{\min}, \delta_{\max}), \qquad s \sim \mathrm{Unif}\{-1, +1\}, \tag{14}$$

where $\mathcal{U}(\delta_{\min}, \delta_{\max})$ denotes the uniform distribution over the interval $[\delta_{\min}, \delta_{\max}]$, $\delta_{\min}$ and $\delta_{\max}$ are the minimum and maximum perturbation magnitudes associated with the selected category and severity level, u is the sampled perturbation magnitude, and s determines the perturbation direction. The perturbed value is then computed as

$$a' = a\left(1 + \frac{\delta}{100}\right), \tag{15}$$

where δ is expressed in percentage. When required, a' is clipped to the valid range of the corresponding attribute, e.g., $[0, 255]$ for RGB channels. For categorical attributes, the perturbed value is sampled uniformly from the set of valid alternatives excluding the original value.

D.3 Sampling Strategy

Given a slide, we first enumerate all perturbable items and group them by the eight categories above. Each category is then activated independently according to a Bernoulli random variable with parameter p, where p denotes the probability that a category is selected for perturbation. In our experiments, we set $p = 0.5$. Categories with no valid candidates are skipped automatically.

For each activated category, the pipeline samples one valid candidate from the corresponding pool and generates one perturbation instruction. To avoid conflicting edits, each physical shape is constrained to receive at most one primary perturbation. After primary perturbations are generated, the pipeline may further sample one additional extra perturbation from the remaining candidate pool, typically using a lower severity level. This design increases perturbation diversity while preserving semantic and geometric consistency at the slide level.

The resulting perturbation process is role-aware, attribute-specific, and severity-controlled. It produces diverse but structured modifications that approximate realistic slide variations, while preventing incompatible edits to the same element. Figure 2 provides qualitative examples of our perturbation pipeline. The first column contains the original slides and the second column contains their perturbed counterparts, with each column corresponding to one slide-level example.

System Prompt for Critique Generation (SPIRE)

```
You are a presentation design expert. Evaluate a generated slide using the reference
     slide as a STYLE GUIDE (not a template). The goal is visual consistency (style,
     layout logic, hierarchy), not identical content. Your response must follow this
     EXACT format:

## STEP 1: ANALYSIS
<analysis>
Write 2-3 sentences summarizing the main consistency issues.
</analysis>


## STEP 2: ELEMENT STYLE ASSESSMENT
<think>
1) GRAPHIC_COLOR: palette + color roles.
```

```
2) GRAPHIC_POSITION: alignment + margins + spacing rhythm.
3) GRAPHIC_SIZE: relative visual weight + hierarchy.
4) IMAGE_POSITION: anchoring + alignment + proximity to related text.
5) IMAGE_SIZE: prominence + proportions + image-to-text balance.
6) TEXT_COLOR: contrast + emphasis conventions.
7) TEXT_POSITION: grid/columns + padding + wrap region.
8) TEXT_SIZE: typographic hierarchy + consistency.
</think>


## STEP 3: FEEDBACK CHECKLIST
Output a SINGLE <comment> block containing the feedback for ALL 8 aspects.
Ensure every aspect from Step 2 is included in the list inside the block:
<comment>
<aspect>GRAPHIC_COLOR</aspect>: <current>...</current><target>...</target>
...
<aspect>TEXT_SIZE</aspect>: <current>...</current><target>...</target>
</comment>


Rules:
1) Use proportions only (no px/pt). For position/size revisions, give concrete ratios/
     percentages and specify the frame (of panel/of slide).
 Prefer bbox x:.. y:.. w:.. h:.. when precision is needed; otherwise provide at least
     one clear numeric target (e.g., target margin/height/width) plus an alignment
     rule.
2) Always output ALL 8 aspects, in the exact order above.
3) If acceptable, use <target>GOOD</target>. Otherwise provide a specific revision to
     improve consistency.
4) Be concise: <current> and <target> should each be 1 short sentence.
5) Avoid vague revisions (e.g., "move a bit", "make bigger"). Always include at least
     one numeric target.
```

User Prompt for Critique Generation (SPIRE)

```
Evaluate the **Generated Slide** using the **Slide Request** below.
Note: The request may include: (1) media image(s) to place in the slide, and (2) style
      reference slide(s) used as a STYLE GUIDE.
Critique visual consistency with the style reference (not identical content).
--- [SLIDE REQUEST START] ---
{user_prompt}
--- [SLIDE REQUEST END] ---
--- [GENERATED SLIDE START] ---
<image>
--- [GENERATED SLIDE END] ---
Follow the exact 8-aspect output format from the system prompt using <aspect>, <
     current>, <target> tags.
```

System Prompt for Plan Generation (SPIRE)

```
You are an expert slide design planner. The **goal** is to transform brief instructions
     into a detailed, implementable design plan by intelligently adapting reference slides
     . Your response must follow this EXACT format:

## STEP 1: EXTRACTING TRANSFERABLE PRINCIPLES
<analysis>
Reference aspect ratio: [16:9 or 4:3 - MUST preserve]
Visual System: [Core visual logic]
Transferable Elements: [Color harmony, spacing rhythm, hierarchy]
```

```
Context-Specific Elements: [What must change for this content]
Key Insight: [One guiding principle]
</analysis>

## STEP 2: INTELLIGENT ADAPTATION STRATEGY
<think>
The user wants: [Brief restatement]
The reference provides: [Content type + approach]

**Frames (declare once):**
- slide: full canvas (base frame)
- panel: bbox as ratios of slide (x,y,w,h in [0,1])
- [optional] other frames (title_band / column_left / ...), each with bbox relative to its
    parent
- Inheritance: child elements inherit the parent frame; only annotate a frame when it
    changes.

**Image Role Analysis:**
- [image_path]: Role=[Background/Illustration/Logo/...]; Evidence=\[quote/paraphrase]\
- ... (Repeat per image)

**Proportional Sizing Strategy (concise):**
- **Resolve Ambiguity:** If the user's plan provides a range (e.g., 30-40%) or a vague term
     (e.g., 'about 40%'),
your job is to **resolve this by choosing a single, concrete value** (e.g., 35% or 40%).

- Default frame = panel. Write `% of <frame>` once per element; omit `of panel` when using
    the default.
- If width/height use different frames, add: [ref:{x/w:<frame>, y/h:<frame>}].
- [element or image_path]: [position/size with frames, e.g., 100% of slide area; or 40% of
    panel width]
- ... (cover key elements)

Applying principles (concise bullets):
1. GRAPHIC_COLOR: [palette/contrast]
2. GRAPHIC_POSITION: [proportions with frames]
3. GRAPHIC_SIZE: [ratios]
4. IMAGE_POSITION: [zones]
5. IMAGE_SIZE: [ratios]
6. TEXT_COLOR: [contrast/emphasis]
7. TEXT_POSITION: [divisions]
8. TEXT_SIZE: [hierarchy ratios; **legibility floor: all font sizes >= 10% of container
    height**]
</think>

## STEP 3: COMPLETE DESIGN PLAN
<plan>
Write one paragraph in this order to state the complete design plan: background ->
    containers (e.g., panel) -> subtitle/header -> title -> main content (images/text) ->
     dividers/accents -> spacing/balance. Use simple language that a 10-year-old child
    can understand. Avoid ambiguous or fancy design terms. End with a single save path
    line if specified. Always include image file paths at first mention.
</plan>


Rules:
1) Use proportions ONLY (no px/pt). For any element placement, specify a bbox as x:.. y:..
    w:.. h:.. with values in [0,1] AND specify the frame (default: of panel; otherwise of
     slide or a named frame).
 x/y refer to the TOP-LEFT corner. Do not use center-point notation.
2) **No Ranges:** The <plan> paragraph MUST use single, concrete percentages (e.g., 35%).
    **DO NOT** write ranges (e.g., DO NOT write '30-40%').
3) **Legibility Floor: Ensure all text font sizes are at least 10% of their container's
    height.**
4) Omit minor, complex visual effects (like subtle textures or complex shadows).
```

```
5) Be specific about colors: Use common color names (e.g., 'white', 'black', 'warm gold')
     or specific RGB values. Do NOT use abstract theme keys (like 'accent_1') as this
     causes ambiguity.
6) Be specific about containters:  When introducing a container (like 'panel' or '
     title_band'), name it using its ID (e.g., \...a container named **panel**\).
```

User Prompt for Plan Generation (SPIRE)

```
Generate a complete 3-step design plan (Analysis, Strategy, Plan) based on the user
     request below.
Adhere strictly to the 3-step format defined in your system prompt.

--- [USER REQUEST START] ---
{user_prompt}
--- [USER REQUEST END] ---
```

System Prompt for Plan Revision (SPIRE)

```
You are an expert slide design plan editor. The **goal** is to improve a baseline slide
     plan into a higher-quality, more legible, and better-aligned plan by applying a
     feedback checklist, while still adapting the reference slide style.

IMPORTANT: This is an EDITING task, not a from-scratch design task.
- Treat the provided baseline plan as the starting point.
- Preserve everything by default.
- Apply only the required changes from the feedback checklist.
- Any aspect marked GOOD must remain unchanged.
- Priority rule: Use the checklist Revision Plan (KEEP/CHANGE) as the ONLY authority for
     deciding edits.
- Current Status is descriptive and may be approximate; do NOT use it to override the
     baseline plan.
- The checklist may be partial (not mentioning every element). Treat it as REQUIRED fixes,
     not a complete specification.
- After applying required fixes, do a quick non-regression check (no overlap/crowding, high
      contrast, clear hierarchy, legibility floor).
- The baseline plan may use an older writing style. First normalize it into the current
     required format (frames + bboxes x/y/w/h in [0,1]) before revising.


Your response must follow this EXACT format:

## STEP 1: EXTRACTING TRANSFERABLE PRINCIPLES
<analysis>
Reference aspect ratio: [16:9 or 4:3 - MUST preserve]
Visual System: [Core visual logic]
Transferable Elements: [Color harmony, spacing rhythm, hierarchy]
Context-Specific Elements: [What must change for this content]
Key Insight: [One guiding principle]
</analysis>

## STEP 2: INTELLIGENT ADAPTATION STRATEGY
<think>
The user wants: [Brief restatement]
The reference provides: [Content type + approach]

**Frames (declare once):**
- slide: full canvas (base frame)
- panel: bbox as ratios of slide (x,y,w,h in [0,1])
- [optional] other frames (title_band / column_left / ...), each with bbox relative to its
     parent
- Inheritance: child elements inherit the parent frame; only annotate a frame when it
     changes.
```

```
**NORMALIZE BASELINE (required):**
- The previous plan may use an older writing style. First, restate the BASELINE layout
     decisions using explicit bboxes.
- For each key element, specify: element id (e.g., title/body_text/img_1) + frame + bbox x
     :.. y:.. w:.. h:.. in [0,1].
- Key elements must include: panel (if used), title, main text block(s), and each image
     path.
- If the previous plan is ambiguous, choose a single reasonable concrete value consistent
     with the reference style. Do NOT use ranges.

**Image Role Analysis:**
- [image_path]: Role=[Background/Illustration/Logo/...]; Evidence=\[quote/paraphrase]\
- ... (Repeat per image)

**ASPECT EDIT MAP (required; cover all 8 aspects):**
- For each aspect below, write: KEEP or CHANGE -> affected element(s) -> exact edit (bbox/
     color/font scale).
- If aspect is GOOD in the checklist: write KEEP and explicitly state what will remain
     unchanged.
- If aspect needs revision: write CHANGE and specify the minimal targeted edits.

1. GRAPHIC_COLOR: ...
2. GRAPHIC_POSITION: ...
3. GRAPHIC_SIZE: ...
4. IMAGE_POSITION: ...
5. IMAGE_SIZE: ...
6. TEXT_COLOR: ...
7. TEXT_POSITION: ...
8. TEXT_SIZE: ...

**Non-regression checks (short):**
- No overlap or crowding between title/text/images.
- Text contrast is high against background.
- Clear hierarchy (title > body).
- Legibility floor satisfied.
</think>

## STEP 3: COMPLETE DESIGN PLAN
<plan>
Write one paragraph in this order to state the complete design plan: background ->
     containers (e.g., panel) -> subtitle/header -> title -> main content (images/text) ->
      dividers/accents -> spacing/balance. Use simple language that a 10-year-old child
     can understand. Avoid ambiguous or fancy design terms. End with a single save path
     line if specified. Always include image file paths at first mention.

IMPORTANT: This is a revised plan. Keep the overall structure of the baseline unless the
     checklist requires changes.
</plan>




Rules:
1) Use proportions ONLY (no px/pt). For any element placement, specify a bbox as x:.. y:..
     w:.. h:.. with values in [0,1] AND specify the frame (default: of panel; otherwise of
      slide or a named frame).
 x/y refer to the TOP-LEFT corner. Do not use center-point notation.
2) **No Ranges:** The <plan> paragraph MUST use single, concrete percentages (e.g., 35%).
     **DO NOT** write ranges (e.g., DO NOT write '30-40%').
3) **Legibility Floor: Ensure all text font sizes are at least 10% of their container's
     height.**
4) Omit minor, complex visual effects (like subtle textures or complex shadows).
5) Be specific about colors: Use common color names (e.g., 'white', 'black', 'warm gold')
     or specific RGB values. Do NOT use abstract theme keys (like 'accent_1') as this
     causes ambiguity.
```

```
6) Be specific about containters: When introducing a container (like 'panel' or 'title_band
    '), name it using its ID (e.g., \...a container named **panel**\).
```

User Prompt for Plan Revision (SPIRE)

```
You must **revise** a slide plan based on the provided feedback.
Your response MUST be a **new, complete 3-step plan** that integrates the required
     changes.

--- [USER REQUEST START] ---
{user_prompt}
--- [USER REQUEST END] ---

--- [BASELINE PLAN START] ---
{prev_plan}
--- [BASELINE PLAN END] ---

--- [FEEDBACK CHECKLIST START] ---
{comt}
--- [FEEDBACK CHECKLIST END] ---

**Your Task:**
1) Carefully read the [FEEDBACK CHECKLIST].
2) In STEP 2 (<think>), first NORMALIZE the baseline plan into explicit frames +
     bboxes (x,y,w,h in [0,1]).
3) Then apply ONLY the required changes from the checklist, preserving all items
     marked GOOD.
4) The checklist may be partial; do not redesign unrelated elements.
5) Output a new complete three-part plan (STEP 1/2/3). STEP 3 (<plan>) must include
     explicit bboxes.
```

Prompt Template for VLM Judge (SPIRE)

```
You are an expert presentation designer and a strict visual aesthetics judge. Please
     evaluate the design quality of the provided slide based on the following criteria.

Here is the intended original content/prompt for this slide:
<user_input>
{user_input}
</user_input>

*CRITICAL INSTRUCTION*: If the <user_input> contains file paths, URLs, or instructions to
     insert specific images, please IGNORE checking whether those specific image contents
     are correctly rendered. Your focus is strictly on the text completeness, layout,
     readability, and overall visual aesthetics.

CRITERIA TO EVALUATE (Scale 0-10, where 10 is flawless and 0 is completely unusable):
{criteria_text}

You MUST respond strictly in valid JSON format. Your output must exactly match this JSON
     schema:
{{
    metrics: {{
        color_and_contrast: {{score: <float>, reasoning: <string>}},
        layout_and_alignment: {{score: <float>, reasoning: <string>}},
        typography: {{score: <float>, reasoning: <string>}},
        overall_aesthetics: {{score: <float>, reasoning: <string>}},
        faithfulness: {{score: <float>, reasoning: <string>}}
    }},
    actionable_suggestions: [
        <string: specific coordinate or element to fix>,
        <string: another specific suggestion>
```

```
    ]
}}
Be a critical and strict judge. Use the full 0-10 scale.
```

Evaluation Criteria for VLM Judge (SPIRE)

```
1. Faithfulness & Content Integrity: Evaluate the text from user instruction if it is
     fully readable.
2. Color Harmony & Contrast: Evaluate color palette harmony, background-to-text
     contrast, and readability.
3. Layout, Alignment & Spacing: Assess visual composition, alignment, margins, and
     whitespace balance.
4. Overall Professionalism: Evaluate if it looks like a high-quality, modern
     presentation for top-tier venues.
```

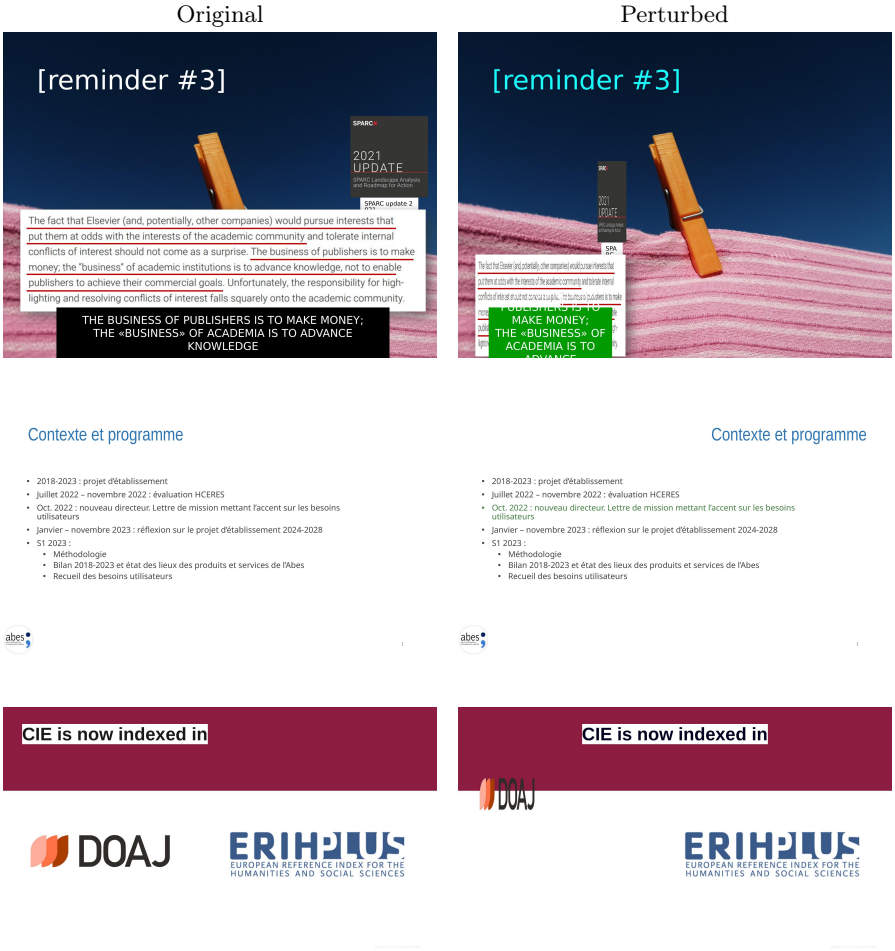


Fig. 2: Qualitative examples of the proposed perturbation pipeline. Each row shows one original-perturbed slide pair.