DIFFSCHED: CLOSED-LOOP SEMANTIC COMPUTATION SCHEDULING FOR EFFICIENT COMPOSITIONAL TEXT-TO-IMAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image diffusion models still struggle with prompts that combine multiple objects, attributes, counts, and relations. Regional control can improve compositional accuracy, but fixed schedules may waste computation on resolved requirements while providing insufficient intervention for difficult ones. We propose **DIFFSCHED**, a closed-loop scheduling framework that adapts regional computation and semantic review during generation. Before sampling, role-specialized agents decompose the prompt into objects, relations, spatial regions, and an initial generation plan. During sampling, a lightweight trigger tracks the magnitude and recent trend of disagreement between regional and global denoising predictions. At eligible checkpoints, a low trigger score stops the current regional interventions without invoking semantic review; otherwise, a Review Agent diagnoses unresolved requirements and the Scheduler updates the next intervention interval. The residual guides execution rather than certifying semantic correctness, while high-risk requirements retain mandatory checks. On GenEval 2 and T2I-CompBench++, **DIFFSCHED** achieves a better quality–computation trade-off than fixed scheduling heuristics, using approximately 42–56% fewer denoiser evaluations while improving compositional scores. These results show that selectively allocating regional computation and semantic feedback can improve compositional generation without uniformly increasing inference cost.

1 INTRODUCTION

Text-to-image diffusion models produce convincing images, yet complex prompts still expose failures in object presence, attribute binding, counting, and relations. A realistic image can omit a requested object or reverse its position relative to another. TIFA and T2I-CompBench++ document these compositional errors (Hu et al., 2023; Huang et al., 2025), while GenEval 2 expands primitive coverage and compositional difficulty and highlights the need to audit evaluation as generators improve (Kamath et al., 2025). Stronger global conditioning alone does not determine which requirement needs further attention during sampling. Compositional generation also requires deciding where additional computation is useful and when it can stop.

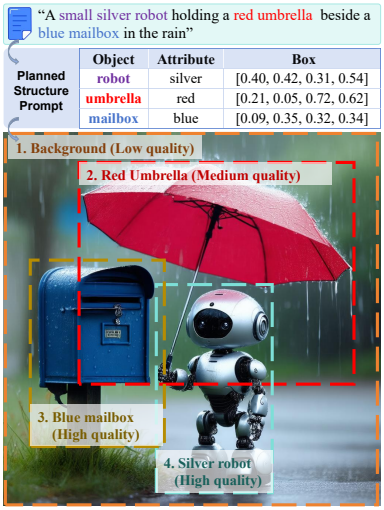


Figure 1: Compositional requirements require different amounts of computation.

Existing methods introduce structure through attention control (Feng et al., 2023; Chefer et al., 2023), spatial constraints (Li et al., 2023; Xie et al., 2023), and regional conditioning (Bar-Tal et al., 2023; Yang et al., 2024). MCCD uses role-specialized agents to parse scenes before hierarchical compositional diffusion (Li et al., 2025); more recently, coDrawAgents combines priority-based object grouping with visually grounded planning and layout checking (Li et al., 2026b). These approaches improve *what* and *where* to generate, but additional control has a cost. As

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

illustrated in Fig. 1, not every pixel needs equally detailed rendering to satisfy a prompt: a smooth background may need little refinement, while a small object carrying a requested attribute or relation may need sustained intervention. Under a fixed regional schedule, resolved requirements can continue consuming computation while difficult relations receive insufficient intervention; allocation therefore needs to follow generation progress, not only the initial layout. The objective is not to lower overall image quality, but to avoid spending equal effort on requirements with different refinement needs.

Recent acceleration methods support this non-uniform view. DeepCache reuses intermediate features (Ma et al., 2024); RAS selectively updates regions (Liu et al., 2026b); and RALU reserves higher-resolution processing for artifact-prone regions (Jeong et al., 2026). SDiT schedules regional updates from spatial complexity (Lin et al., 2026), while HSA assigns different denoising durations to video tokens (Chu & Patel, 2026). However, spatial complexity or model activity does not establish whether a particular prompt requirement is satisfied. Conversely, semantic feedback can guide correction: PromptLoop learns step-wise prompt refinement from latent feedback (Lee & Ye, 2026), and Inference Intervention revisits earlier diffusion states after detecting structural errors (Guo et al., 2025). These directions suggest complementary roles for inexpensive execution signals and more costly semantic diagnosis, rather than treating either as a complete scheduling criterion. Computation scheduling also offers a promising basis for future multi-device collaboration. Patch-level, heterogeneous, and topology-aware parallel inference already distribute diffusion workloads across GPUs (Fang et al., 2025; Liang et al., 2025; Yang et al., 2026b). Requirement-level schedules could additionally expose which regional actions and reviews deserve resources, allowing future systems to match semantic needs to device capacity while accounting for dependencies and communication. This prospect motivates making the allocation explicit, without assuming that regional branches are independent. The immediate question is therefore: *which regional interventions should continue, and when is semantic review worth invoking?*

We address this question with **DIFFSCHED**, a closed-loop semantic computation scheduler. Before sampling, role-specialized agents decompose the prompt into objects, relations, spatial supports, and an initial regional plan. During sampling, a global branch and selected regional branches share the current latent state. A lightweight trigger measures normalized disagreement between their predictions inside each regional support. At eligible checkpoints, a sufficiently low score stops the current regional interventions and bypasses review. Otherwise, the Review Agent inspects an intermediate preview and returns requirement-level diagnoses. The Scheduler uses these diagnoses to continue, strengthen, change, or roll back an intervention for the next execution interval. A low residual is not a semantic correctness judgment; high-risk requirements retain mandatory checks. This separates the value of continuing an action from the need to diagnose what remains wrong. The distinction from complexity-based acceleration or prompt refinement is the joint scheduling of requirement-level intervention and the semantic observations used to revise it. Both regional computation and review frequency respond to the evolving generation rather than a fixed schedule.

We evaluate **DIFFSCHED** on GenEval 2 and T2I-CompBench++, measuring compositional faithfulness, denoiser evaluations, diffusion GPU time, review workload, and complete prompt-to-image latency. Component and scheduling ablations examine the value of planning, feedback, and selective triggering under their actual execution costs; sensitivity studies vary the trigger threshold and semantic-agent backbone. The related source is in `https://anonymous.4open.science/r/DIFFSCHED-E1BB/`. Our contributions are threefold:

- We formulate compositional generation as *semantic computation scheduling*, assigning regional actions and computation to explicit objects and relations according to evolving intervention needs.
- We introduce a lightweight regional-control residual trigger for stopping low-value continuation and avoiding unnecessary semantic review, while retaining mandatory checks. Our analysis characterizes its local decisions under diminishing returns and valid upper calibration.
- We develop a closed-loop scheduler that translates agent diagnoses into subsequent regional actions and evaluate its quality–cost trade-off through end-to-end accounting, component and scheduling ablations, trigger sensitivity, and controller robustness.

2 Background and Related Work

Compositional Control in Diffusion Models. Compositional diffusion methods use attention control, spatial grounding, and regional conditioning. Structured Diffusion and Attend-and-Excite

improve concept coverage and binding (Feng et al., 2023; Chefer et al., 2023), while ISAC stabilizes instance layouts before binding their semantics (Jo et al., 2025). GLIGEN and BoxDiff impose spatial constraints (Li et al., 2023; Xie et al., 2023); MultiDiffusion fuses diffusion paths under shared spatial constraints (Bar-Tal et al., 2023). RPG decomposes prompts into regional tasks (Yang et al., 2024), and MCCD uses specialized agents to extract scene elements before hierarchical compositional diffusion (Li et al., 2025). These methods differ in how they obtain conditions and regions, but additional regional predictions incur computation. A relation may require joint control of several objects, so counting object boxes alone does not characterize the amount of regional work. For semantic groups $\mathcal{U} = \{u_1, \ldots, u_M\}$ and active iteration sets W_m, a simple accounting proxy is

$$C_{\text{diff}} \approx T\bar{c}_D + \bar{c}_D \sum_{m=1}^{M} |W_m|, \tag{1}$$

where T is the number of global steps and $\bar{c}_D$ the average evaluation cost. The expression separates global work from the additional evaluations assigned to each group. It is not a latency model: batching, resolution, and implementation affect execution time. **DIFFSCHED** schedules these additional evaluations rather than replacing the regional controller.

Adaptive Computation in Diffusion Inference. DeepCache reuses features across denoising steps (Ma et al., 2024); RAS updates selected regions while reusing cached predictions elsewhere (Liu et al., 2026b); and RALU selectively upsamples artifact-prone regions (Jeong et al., 2026). SDiT combines region clustering, complexity-driven scheduling, and boundary refinement (Lin et al., 2026). For video generation, HSA assigns token-specific step counts from velocity dynamics while maintaining global context through cached attention states (Chu & Patel, 2026). These approaches exploit unequal refinement needs without assigning equal refinement effort everywhere. Model activity and regional complexity, however, are not explicit tests of whether an attribute, count, or relation requested by the prompt has been satisfied. Distributed inference addresses a complementary constraint: PipeFusion pipelines patches and model layers (Fang et al., 2025), STADI balances step–patch workloads across heterogeneous GPUs (Liang et al., 2025), and SwiftFusion optimizes topology-aware sequence parallelism (Yang et al., 2026b). Such methods manage execution and communication; requirement-level scheduling instead decides which semantic interventions remain useful. Combining these decisions with device placement is a potential extension, not a distributed capability evaluated here.

Feedback-Guided and Agentic Generation. Feedback can revise conditions, layouts, or generation actions. RPG supports iterative generation and editing (Yang et al., 2024); coDrawAgents checks layouts against the evolving canvas (Li et al., 2026b); and T2I-Copilot evaluates outputs to guide regeneration (Chen et al., 2025a). Within sampling, Inference Intervention detects structural errors and can resume from an earlier latent (Guo et al., 2025), while PromptLoop trains a multimodal prompt-refinement policy with reinforcement learning on intermediate feedback (Lee & Ye, 2026). A frozen image generator therefore does not imply a training-free controller: PromptLoop learns the refinement policy rather than generator weights. These methods establish the value of semantic feedback, but acquiring it adds decoding and reasoning cost beyond regional denoising. We retain the additive accounting

$$C_{\text{e2e}} = C_{\text{diff}} + C_{\text{review}} + C_{\text{aux}}, \tag{2}$$

where C_{review} covers semantic inspection and C_{aux} covers planning, preview decoding, scheduling, and related overhead. This is work accounting, not elapsed latency; end-to-end time is measured separately to account for overlapping operations. **DIFFSCHED** jointly schedules regional interventions and the semantic observations used to revise them, rather than invoking feedback at every candidate checkpoint. The trigger and receding-horizon update are introduced in Sec. 3; additional related work appears in Appendix A.

3 METHODOLOGY

Overview. Given a prompt p, **DIFFSCHED** augments a pretrained latent text-to-image diffusion model (Rombach et al., 2022) with requirement-level computation scheduling. Let z_n be the latent at denoising iteration n, t_n its noise level, and c_p the full-prompt condition. The backbone produces the global prediction $\epsilon_n^{\text{g}} = \epsilon_\theta(z_n, t_n, c_p)$ and otherwise follows its original sampler. DIFFSCHED leaves this global trajectory unchanged and schedules only the additional compositional controls around it. Stage I uses role-specialized agents to extract explicit requirements, semantic groups, spatial supports, and an initial plan. Stage II executes that plan over short intervals, using a cheap

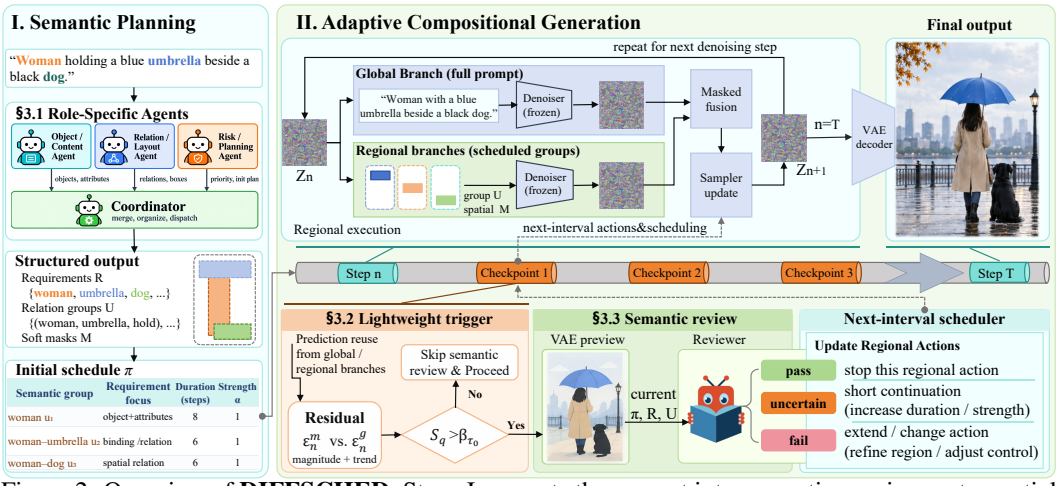


Figure 2: Overview of **DIFFSCHED**. Stage I converts the prompt into semantic requirements, spatial supports, and an initial regional schedule. Stage II runs one global diffusion trajectory with scheduled regional branches. At each checkpoint, a lightweight residual gate either stops low-value regional continuation or requests semantic review; the diagnosis updates only the next interval.

diffusion-side residual to decide whether the current regional intervention should continue. Semantic review is invoked only when the residual remains high or when a checkpoint is marked mandatory, after which the next interval is rescheduled.

For interval $\mathcal{I}_q$, we write the generation plan as

$$\Pi_q = \{a_\ell\}_{\ell=1}^{L_q}, \qquad a_\ell = (\tau_\ell, u_\ell, W_\ell, \lambda_\ell), \tag{3}$$

where τ_ℓ is the generation action, u_ℓ its semantic target, W_ℓ its execution interval, and $\lambda_\ell \geq 0$ its control strength when applicable. Regional denoising and spatial control are the main actions; late local refinement and rollback are optional corrective actions. Semantic review is not part of Π_q itself, but an observation used to decide the next plan. Let $\mathcal{C} = \{c_0, \ldots, c_Q\}$ be the ordered checkpoints, with $c_0 = 0$ and $c_Q = T$, and define $\mathcal{I}_q = [c_q, c_{q+1})$ for $q = 0, \ldots, Q-1$. The initial plan Π_0 is executed on $\mathcal{I}_0$; at each later checkpoint c_q, DIFFSCHED decides the regional allocation for $\mathcal{I}_q$. No regional allocation extends beyond the next checkpoint without being reconsidered.

3.1 Stage I: Semantic Structuring and Initial Scheduling

Following agent-assisted scene decomposition in compositional generation (Yang et al., 2024; Li et al., 2025), DIFFSCHED uses role-specialized agents for content, relation/layout, and risk/planning analysis. These roles may share the same underlying multimodal model; their specialization comes from different instructions and structured outputs rather than separately trained networks. A Coordinator merges the outputs into a scene graph $\mathcal{G} = (V, E)$, where each node $v_i = (o_i, \mathcal{A}_i, b_i)$ records an object, its requested attributes, and an approximate spatial region, and each edge $e_{ij} = (v_i, v_j, \rho_{ij})$ records a spatial, depth, or interaction relation. The corresponding statements form the requirement set $\mathcal{R} = \{r_1, \ldots, r_N\}$ used by the Review Agent later in sampling.

Requirements that should be handled together are collected into semantic groups $\mathcal{U} = \{u_1, \ldots, u_M\}$. Relational requirements remain joint rather than being reduced to independent object boxes. For example, "a woman holding a blue umbrella next to a black dog" yields a woman–umbrella group and a woman–dog group. Each group receives a textual condition c_m and a soft latent-space support $M_m \in [0,1]^{H \times W}$ obtained from the enclosing region of its member entities with a small context margin, while the full prompt remains active globally. The planning agent then assigns each group a coarse intervention level $\kappa_m \in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$ and an observability phase indicating when the requirement should become visually judgeable. Regional step allocation is drawn from

$$\mathcal{K} = \{0, K_{\mathrm{s}}, K_{\mathrm{m}}, K_{\mathrm{l}}\}. \tag{4}$$

Low-risk requirements may begin without a regional branch, whereas small objects, difficult bindings, and interaction relations can receive longer initial windows. Stage I also proposes candidate spatial controls and marks mandatory checkpoints for requirements whose correctness should not be inferred

from diffusion-side signals alone. These allocations form Π_0 and remain provisional: after sampling begins, every continuing regional action is reconsidered at the next checkpoint.

3.2 Stage II-A: Regional Execution and Lightweight Triggering

On each interval $\mathcal{I}_q$, the global branch remains active while only the semantic groups selected by Π_q receive additional regional computation. Let $\mathcal{A}_n$ denote the groups active at iteration n. For each $u_m \in \mathcal{A}_n$, the same frozen backbone produces $\epsilon_n^m = \epsilon_\theta(z_n, t_n, c_m)$. We fuse these regional predictions with the global prediction using soft masks, following regional compositional diffusion (Bar-Tal et al., 2023; Li et al., 2025):

$$\omega_{m,n} = \frac{\lambda_m(n) M_m}{1 + \sum_{j \in \mathcal{A}_n} \lambda_j(n) M_j}, \quad \widetilde{\epsilon}_n = \epsilon_n^{\mathrm{g}} + \sum_{m \in \mathcal{A}_n} \omega_{m,n} \odot \left(\epsilon_n^m - \epsilon_n^{\mathrm{g}}\right). \tag{5}$$

The original sampler advances z_n using $\widetilde{\epsilon}_n$. When needed, the planned boxes or masks can also drive a training-free spatial controller such as BoxDiff (Xie et al., 2023).

Regional-control residual. DIFFSCHED does not call the Review Agent at every checkpoint. While a regional branch is active, it measures the normalized disagreement between the regional and global predictions inside the target support,

$$r_{m,n} = \frac{\|M_m \odot (\epsilon_n^m - \epsilon_n^{\mathrm{g}})\|_2}{\|M_m \odot \epsilon_n^{\mathrm{g}}\|_2 + \varepsilon}. \tag{6}$$

Both predictions are already available during regional execution, so Eq. 6 adds no denoiser evaluation. We exclude the fusion weight $\omega_{m,n}$ because a smaller control strength would otherwise reduce the score mechanically even when the underlying regional/global disagreement remains large. The residual is therefore an execution signal, not a semantic correctness score: it indicates whether the regional branch is still producing a meaningfully different prediction, but not whether the prompt requirement has been satisfied. This use of internal diffusion dynamics is related to region-adaptive sampling (Liu et al., 2026b), but here the signal controls requirement-level regional continuation and the need for semantic review.

For each group whose current action reaches checkpoint c_q ($q \geq 1$), we summarize the h most recent active iterations before the checkpoint by the mean residual $\bar{r}_{m,q}$ and normalized recent increase $\delta r_{m,q}$. The group and checkpoint scores are

$$s_{m,q} = \bar{r}_{m,q} + \gamma[\delta r_{m,q}]_+, \qquad s_q = \max_{m \in \mathcal{A}_q} s_{m,q}, \tag{7}$$

where $\gamma \geq 0$ controls the trend term and $\mathcal{A}_q$ denotes the regional actions reaching c_q. We use $\tau_\beta = \beta\tau_0$, with τ_0 calibrated on held-out prompts and $\beta = 1$ by default. At a gate-eligible checkpoint, $s_q \leq \tau_\beta$ stops the current regional actions from continuing into $\mathcal{I}_q$ and skips semantic review; $s_q > \tau_\beta$ routes the state to the Review Agent. Mandatory checkpoints bypass the gate. The rule is deliberately one-sided: a low residual can support stopping the current intervention, but it does not by itself establish semantic correctness.

3.3 Stage II-B: Semantic Review and Receding-Horizon Rescheduling

When review is requested at checkpoint c_q, DIFFSCHED caches the noisy latent z_{c_q}, forms the sampler's current clean-latent estimate $\widehat{z}_0^{(q)}$, and decodes $\widehat{x}_q = \mathcal{D}(\widehat{z}_0^{(q)})$ with the pretrained VAE. The preview is used only for observation; unless rollback is selected, sampling continues from the cached diffusion latent rather than from a VAE re-encoding. Intermediate diffusion states have similarly been used to guide later generation decisions in prior work (Lee & Ye, 2026; Guo et al., 2025).

The **Review Agent (Reviewer)** compares $\widehat{x}_q$ with the subset of $\mathcal{R}$ that should be observable at the current phase and returns

$$y_j^{(q)} \in \{\texttt{pass}, \texttt{uncertain}, \texttt{fail}\}, \tag{8}$$

together with confidence, affected semantic group, and failure type. The Reviewer diagnoses the state but does not choose the generation action. The Scheduler combines the diagnosis Y_q, the previous plan Π_{q-1}, and residual statistics R_q to construct the plan for the next interval. A confident `pass` removes further regional allocation for that requirement; `uncertain` usually keeps a short continuation; and `fail` can extend the regional window, increase control strength, or switch to stronger spatial control or late refinement. If the useful correction phase has already passed, the

Scheduler may restore an earlier cached latent and resume the full denoising trajectory under revised controls.

For every regional group, the new allocation remains bounded by the next checkpoint,

$$W_{m,q} \subseteq \mathcal{I}_q, \qquad |W_{m,q}| = K_{m,q}, \qquad K_{m,q} \in \mathcal{K}, \tag{9}$$

and, when review is used,

$$\Pi_q = \mathrm{Schedule}(\mathcal{S}, \Pi_{q-1}, Y_q, R_q), \qquad q \geq 1. \tag{10}$$

The resulting plan is executed only on $\mathcal{I}_q$ and is reconsidered at c_{q+1}. This receding-horizon loop keeps the two forms of feedback separate: the residual handles routine continuation decisions at low cost, while the Reviewer is used only when choosing the next action requires semantic evidence.

3.4 Theoretical Analysis of Triggered Semantic Scheduling

At each checkpoint, the trigger decides whether to continue the current regional intervention and whether to acquire semantic feedback. We analyze these decisions through marginal utility and value of information. The results are conditional on diminishing returns and valid upper calibration; Appendix B gives the assumptions, proofs, and the additional conditions needed for a trajectory-level guarantee.

Regional stopping and semantic review. For an active group u_m at c_q, let $U_{m,q}(k)$ be its expected utility under a fixed continuation rule and fixed other actions, with $k \in \{0, \ldots, H_q\}$ and $H_q = |\mathcal{I}_q|$. We optimize

$$J^{\mathrm{D}}_{m,q}(k) = U_{m,q}(k) - \lambda_{\mathrm{D}} k, \tag{11}$$

where $\lambda_{\mathrm{D}} > 0$ prices an evaluation in utility units. Assume $\Delta U_{m,q}(k) = U_{m,q}(k) - U_{m,q}(k-1)$ is non-increasing.

Proposition 1 (Regional stopping rule). The largest optimal allocation in this integer relaxation is

$$K^*_{m,q} = \max(\{1 \leq k \leq H_q : \Delta U_{m,q}(k) \geq \lambda_{\mathrm{D}}\} \cup \{0\}). \tag{12}$$

If $\Delta U_{m,q}(1) \leq \lambda_{\mathrm{D}}$, zero is an optimum; it is unique under strict inequality. At equality, the largest optimizer may be positive. The zero-allocation conclusion also holds on the implemented grid $\mathcal{K} \cap \{0, \ldots, H_q\}$, which contains zero.

Let x_q denote information available before review, Y_q the Reviewer observation, and $\mathfrak{A}_q$ a finite, nonempty set of admissible schedules. For each a, $Q_q(a; x_q, Y_q)$ is its conditional expected downstream utility, net of action costs but excluding the current review cost. The gross value of review is

$$\mathcal{I}_q(x_q) = \mathbb{E}_{Y_q|x_q}\left[\max_{a \in \mathfrak{A}_q} Q_q(a; x_q, Y_q)\right] - \max_{a \in \mathfrak{A}_q} \mathbb{E}_{Y_q|x_q}[Q_q(a; x_q, Y_q)]. \tag{13}$$

Proposition 2 (Review acquisition rule). For integrable action values, $\mathcal{I}_q(x_q) \geq 0$. Optimal review-conditioned action selection is strictly preferable to optimal no-review selection exactly when $\mathcal{I}_q(x_q) > \lambda_{\mathrm{R}}$, where $\lambda_{\mathrm{R}} > 0$ is the acquisition cost in the same utility units. Equality permits either decision; the rule does not assert that an approximate Reviewer–Scheduler attains either optimum.

Dual calibration. At a gate-eligible checkpoint with valid scores for a nonempty active set, let $s_q = \max_{m \in \mathcal{A}_q} s_{m,q}$. Assume non-decreasing upper envelopes satisfy $\Delta U_{m,q}(1) \leq \phi_{\mathrm{D}}(s_{m,q}) + \varepsilon_{\mathrm{D}}$ and $\mathcal{I}_q(x_q) \leq \phi_{\mathrm{R}}(s_q) + \varepsilon_{\mathrm{R}}$. On a finite nonnegative calibration grid $\mathcal{T}$, define

$$\tau_j = \max\{s \in \mathcal{T} : \phi_j(s) + \varepsilon_j \leq \lambda_j\}, \quad j \in \{\mathrm{D}, \mathrm{R}\}, \quad \tau_0 = \min(\tau_{\mathrm{D}}, \tau_{\mathrm{R}}). \tag{14}$$

The feasible sets must be nonempty; otherwise no certified bypass is available. Selecting feasible grid points avoids an unattained supremum. Fitting envelopes on held-out data alone does not establish their validity at every future checkpoint.

Theorem 1 (Dual-certificate trigger). Under the preceding diminishing-return and calibration assumptions, $s_q \leq \tau_0$ implies that (i) zero additional evaluations are optimal for each modeled regional objective, and (ii) $\mathcal{I}_q(x_q) - \lambda_{\mathrm{R}} \leq 0$. The result certifies the two local comparisons, not semantic correctness or optimality of the entire scheduler. Joint stopping requires the action-value consistency condition in Appendix B; a high score has no converse guarantee and is routed to semantic inspection.

Corollary 1 (Expected review saving). For a fixed candidate-checkpoint protocol with $N_{\mathrm{mandatory}}$ mandatory reviews and gate-eligible set $\mathcal{C}_{\mathrm{gate}}$,

$$\mathbb{E}[N_{\mathrm{rev}}] = N_{\mathrm{mandatory}} + \sum_{q \in \mathcal{C}_{\mathrm{gate}}} \Pr(s_q > \tau_0). \tag{15}$$

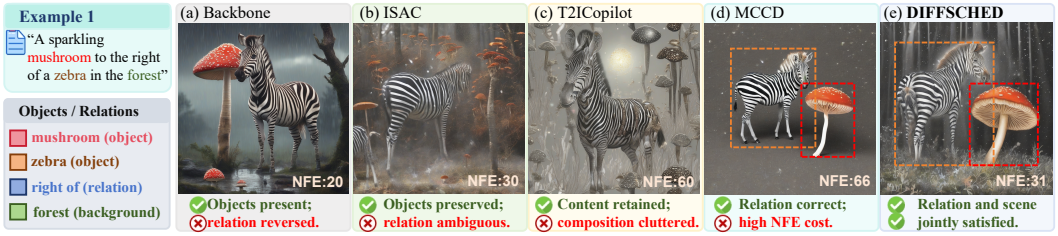


Figure 3: Qualitative comparison. **DIFFSCHED** better preserves the full prompt structure than the baselines while maintaining low NFE cost (More qualitative shown in Appendix D).

Relative to reviewing every checkpoint in the same protocol, the expected number skipped is $\sum_{q \in \mathcal{C}_{\text{gate}}} \Pr(s_q \le \tau_0)$. No independence between checkpoints is required.

Corollary 2 (Threshold scaling). For $\tau_\beta = \beta\tau_0$ and $0 < \beta \le 1$, every certified bypass remains within the default certified region. On a common replayed sequence of scores and mandatory flags, $\beta_1 \le \beta_2$ implies $N_{\text{rev}}^{\text{replay}}(\beta_2) \le N_{\text{rev}}^{\text{replay}}(\beta_1)$, also in expectation. Fresh closed-loop runs can visit different states, so their total review counts and latency need not be monotone. For $\beta > 1$, additional bypasses outside the default region are assessed empirically.
The local comparisons extend to a finite decision horizon under Bellman-consistent joint action values and optimal no-review action selection, as specified in Appendix B.

4 Experiments

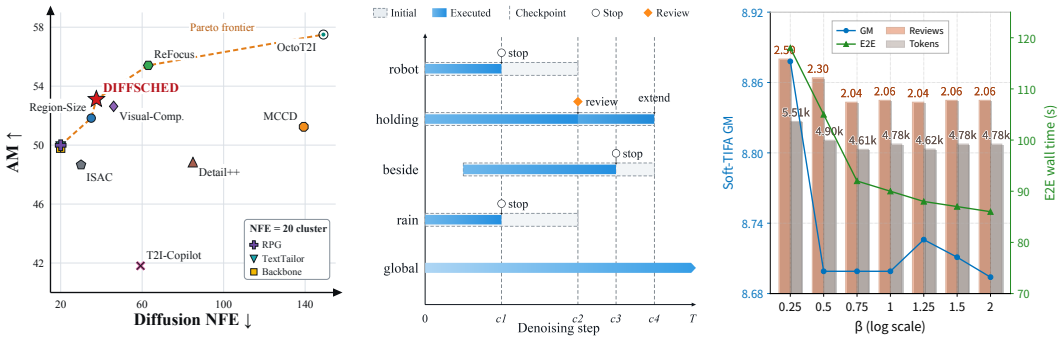
We evaluate **DIFFSCHED** on GenEval 2 (Kamath et al., 2025) and T2I-CompBench++ (Huang et al., 2025). GenEval 2 reports Soft-TIFA geometric mean (GM) at the prompt level and arithmetic mean (AM) over visual atoms, while T2I-CompBench++ evaluates 2D spatial, non-spatial, numeracy, and complex-composition capabilities. We address three questions: **RQ1**, whether adaptive semantic scheduling improves the quality–computation trade-off; **RQ2**, which components and scheduling decisions account for the improvement; and **RQ3**, how the Trigger threshold and semantic-agent backbone affect this trade-off. Controlled variants use the same SDXL diffusion backbone, sampler, resolution, global denoising configuration, prompt set, and random seeds, with 100 test samples per setting. Results on additional diffusion backbones are reported in Appendix D. We report diffusion-side computation separately from semantic-controller overhead, since fewer regional denoiser evaluations do not necessarily translate into lower end-to-end latency. All experiments are run on NVIDIA RTX PRO 6000 GPUs.

4.1 RQ1: Overall Quality–Efficiency Performance

Table 1 compares the backbone, three scheduling baselines, adaptation-required methods, training-free guidance and agent-assisted generation methods, and internal ablations. Runtime numbers are interpreted only when the execution setting is comparable; otherwise they provide quality–cost context. Fig. 3 illustrates the same progression qualitatively, from unconstrained global generation to fixed regional control and finally adaptive regional scheduling.
Quality–cost trade-off. **DIFFSCHED** reaches 10.71/53.11 GM/AM with 37.52 NFE and 6.07 GPU-s. In the AM–NFE plane of Fig. 4a, it provides a favorable quality–cost operating point. Most importantly, it strictly improves over the directly comparable scheduling baselines: Region-Size achieves 9.15/51.83 GM/AM with 64.44 NFE and 10.54 GPU-s, while Visual-Complexity reaches 9.29/52.63 with 85.38 NFE and 13.67 GPU-s. The Static-Semantic-Schedule reaches 9.76/51.25 GM/AM with 42.95 NFE and 7.55 GPU-s, showing that online rescheduling improves both quality and computation over a fixed semantic plan. Thus, **DIFFSCHED** obtains higher GM and AM while using about 42–56% fewer denoiser evaluations and 42–56% less diffusion GPU time than the two heuristic schedulers. The same pattern appears against MCCD: **DIFFSCHED** improves GM/AM from 7.82/51.24 to 10.71/53.11 while reducing NFE from 139.31 to 37.52 and GPU time from 20.93 to 6.07 seconds, corresponding to about 73% fewer denoiser evaluations and 71% lower diffusion-side GPU time. These results indicate that the gain comes from allocating regional computation more selectively rather than simply using more of it.

Method	GenEval 2		T2I-CompBench++				Diffusion Compute	
	GM ↑	AM ↑	2D ↑	Non-Spatial ↑	Num. ↑	Complex ↑	NFE ↓	GPU-s ↓
Backbone (Podell et al., 2024)	9.61	49.79	0.105	0.320	0.633	0.313	**20.00**	**2.93**
Heuristic Scheduling								
Region-Size	9.15	51.83	0.311	0.324	0.667	0.321	64.44	10.54
Visual-Complexity	9.29	52.63	0.311	0.322	0.700	0.331	85.38	13.67
Static-Semantic-Schedule	9.76	51.25	0.324	0.321	0.67	0.320	42.95	7.55
Adaptation-Required								
BiDPO (Liu et al., 2026a)	8.48	**56.70**	0.358	0.328	0.700	0.375	20.00	4.95
OctoT2I (Jiang et al., 2026)	13.14	57.50	0.052	0.318	0.846	0.371	148.98	41.67
Training-Free Guidance								
Detail++ (Chen et al., 2025b)	10.89	48.80	0.056	0.307	0.808	0.258	84.94	16.28
TexTailor (Li et al., 2026a)	7.59	49.94	0.051	0.327	0.692	0.343	20.00	2.77
ISAC (Jo et al., 2025)	7.47	48.66	0.062	0.316	0.577	0.318	30.00	17.80
ReFocus (Phung et al., 2024)	10.08	55.41	0.154	0.325	0.885	0.326	63.00	14.38
Training-Free Agentic Inference								
MCCD (Li et al., 2025)	7.82	51.24	0.311	0.323	0.700	0.317	139.31	20.93
RPG (Yang et al., 2024)	8.83	49.99	0.198	0.305	0.550	0.229	20.00	13.23
T2I-Copilot† (Chen et al., 2025a)	3.82	41.81	0.122	0.301	0.692	0.295	59.20	8.08
Component Ablation								
w/o Planning Agent	10.32	49.08	0.103	0.320	0.567	0.311	23.16	3.73
w/o Review Agent	10.03	51.11	0.303	0.321	0.633	0.327	33.66	5.46
w/o Trigger	10.78	52.47	0.296	0.321	0.633	0.320	40.66	6.79
DIFFSCHED	**10.71**	**53.11**	0.310	0.320	0.667	0.326	37.52	6.07

Table 1: Overall compositional quality and diffusion-side computation. NFE and GPU-s measure diffusion cost; all internal scheduling and ablation variants use the same generation settings.



(a) Scheduling quality–cost frontier. (b) Closed-loop schedule adaptation. (c) Trigger-threshold sensitivity.

Fig. 4a places the scheduling results in the broader quality–cost context: methods toward the upper left achieve higher compositional quality with fewer diffusion evaluations, while **DIFFSCHED** improves both axes relative to the directly comparable scheduling baselines. Fig. 4b uses Fig. 1 as a representative trajectory and shows how the executed schedule departs from the initial plan as requirements resolve or remain difficult. Fig. 4c studies how aggressively semantic review can be bypassed. Semantic-review and end-to-end costs are analyzed separately in RQ2 and Appendix C.

4.2 RQ2: Component and Scheduling Analysis

RQ2 separates the effects of semantic planning, intermediate review, and the allocation rule. Internal variants execute their natural schedules rather than being forced to a matched regional-compute budget; therefore both early termination and additional computation for unresolved requirements appear in the measured cost.

Method	Reviews ↓	Tokens ↓	E2E (s) ↓
T2I-Copilot	2.96	5308.5	67.86
ReFocus	2.00	5321.4	59.29
OctoT2I	3.48	5744.9	144.08
w/o Planning	1.52	5219.9	45.62
w/o Trigger	2.98	8810.5	60.23
DIFFSCHED	**1.40**	**4827.9**	**42.73**

Table 2: End-to-end overhead comparison.

Component ablation. The *w/o Planning Agent* variant removes the Stage-I planning role that constructs the initial requirement-level schedule. It reduces computation to 23.16 NFE and 3.73 GPU-s, but GM/AM fall to 10.32/49.08 from 10.71/53.11, with the larger loss appearing at the atom level. The *w/o Review Agent* variant retains the initial plan but

Configuration	GM ↑	AM ↑	NFE ↓	Diff. GPU-s ↓	Reviews / Img. ↓	Rev. Tokens ↓	E2E Wall ↓
MiMo-V2.5 (Xiaomi MiMo Team, 2026)	**10.71**	**53.11**	37.52	6.07	1.40	4827.90	42.73
Qwen3-VL-8B (Bai et al., 2025)	10.67	52.74	36.30	5.48	1.32	4210.40	38.56
DeepSeek-VL-7B (Lu et al., 2024)	10.64	52.31	35.68	5.21	1.25	3824.60	33.72
Qwen3-VL-2B (Bai et al., 2025)	10.62	51.86	**35.06**	**5.15**	**1.08**	**3012.80**	**25.84**

Table 3: Semantic-agent backbone robustness with the remaining framework fixed. E2E Wall reports measured prompt-to-image latency under each configuration.

removes intermediate semantic diagnosis; it reaches 10.03/51.11 GM/AM with 33.66 NFE and 5.46 GPU-s. Together, these ablations show that the initial semantic plan and online visual diagnosis contribute complementary information. The *w/o Trigger* comparison isolates selective review. Reviewing every candidate checkpoint uses 40.66 NFE and 6.79 GPU-s, versus 37.52 NFE and 6.07 GPU-s for the full method. Table 2 shows a larger controller-side difference: the Trigger reduces reviews from 2.98 to 1.40 per image, Reviewer Tokens from 8810.5 to 4827.9, and E2E latency from 60.23 to 42.73 seconds, reductions of approximately 53%, 45%, and 29%, respectively. GM remains essentially unchanged (10.78 versus 10.71), while AM increases from 52.47 to 53.11. Selective review therefore reduces both diffusion and controller cost without sacrificing the atom-level compositional gain.

Scheduling comparison. The heuristic rows in Table 1 provide a controlled comparison of allocation rules. Region-Size allocates computation from spatial extent, whereas Visual-Complexity uses appearance-based difficulty. Both spend more diffusion computation than **DIFFSCHED** yet attain lower GM and AM. This strict dominance in the quality–NFE plane is the main scheduling result summarized by Fig. 4a. Fig. 4b take fig.1 as an example an explains the mechanism qualitatively: **DIFFSCHED** can stop a resolved regional action, continue or strengthen an unresolved one, and reconsider the allocation at the next checkpoint rather than committing to a fixed heuristic schedule.

4.3 RQ3: Sensitivity and Robustness of **DIFFSCHED**

Trigger sensitivity. We vary the threshold as $\tau_\beta = \beta\tau_0$, with $\beta = 1$ the calibrated default. Fig. 4c shows a clear quality–cost trade-off. As β increases, the bypass region expands: Reviews/Img. drops from 2.50 at $\beta = 0.25$ to 2.06 at $\beta = 1$, while Reviewer Tokens decrease from 5.51k to 4.78k and then saturate. GM shows the opposite trend, dropping most noticeably between $\beta = 0.25$ and 0.5 and changing only slightly thereafter. E2E latency decreases overall but is not strictly monotonic across fresh closed-loop runs. Thus, smaller β preserves more semantic feedback, while larger β favors lower controller cost. For $0 < \beta \leq 1$, bypasses remain within the conservative calibrated region; $\beta > 1$ is treated only as an empirical operating regime.

Semantic-agent robustness. We next replace the semantic-agent backbone with MiMo-V2.5 (Xiaomi MiMo Team, 2026), Qwen3-VL-8B/2B (Bai et al., 2025), and DeepSeek-VL-7B (Lu et al., 2024), while keeping the semantic schema, candidate checkpoints, review format, admissible actions, diffusion backbone, and generation settings fixed. Table 3 shows a clear controller quality–cost trade-off. MiMo-V2.5 achieves the strongest GM/AM (10.71/53.11), but also incurs the largest review workload and end-to-end latency among the tested backbones. Smaller models progressively reduce review frequency, token use, and latency; Qwen3-VL-2B reduces E2E latency from 42.73 s to 25.84 s, with a moderate decrease in AM from 53.11 to 51.86. The scheduling interface therefore remains effective across semantic-agent capacities, while the controller backbone determines a practical quality–latency operating point. Additional fixed-frequency review experiments, Trigger calibration, checkpoint placement, per-agent runtime, and qualitative failure cases are reported in the appendix.

5 Conclusion

We presented **DIFFSCHED**, a closed-loop semantic computation scheduler for compositional text-to-image generation. It combines semantic planning, a lightweight diffusion-side Trigger, and selective semantic review to adapt regional computation during sampling rather than following a fixed schedule. Experiments on GenEval 2 and T2I-CompBench++ show improved quality–computation trade-offs over heuristic and static semantic scheduling, while reducing unnecessary regional denoising and review overhead. Our analysis characterizes when a low residual can support both stopping regional continuation and bypassing semantic review under conservative calibration. These results indicate that effective compositional generation depends not only on what control is applied, but also on when and where computation is allocated. Future work will focus on stronger calibration, cheaper semantic feedback, and transfer across diffusion backbones and generation settings.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101. URL https://doi.org/10.1561/2200000101. Earlier version: arXiv:2107.07511.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing diffusion paths for controlled image generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1737–1752. PMLR, 2023.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021. doi: 10.1093/imaiai/iaaa017. URL https://doi.org/10.1093/imaiai/iaaa017.

Hoyeon Chang, Seungjin Kim, and Yoonseok Choi. Dynamic VLM-guided negative prompting for diffusion models. *arXiv preprint arXiv:2510.26052*, 2025.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics*, 42(4):1–10, 2023. doi: 10.1145/3592116.

Chieh-Yun Chen, Min Shi, Gong Zhang, and Humphrey Shi. T2i-copilot: A training-free multi-agent text-to-image system for enhanced prompt interpretation and interactive generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19396–19405, 2025a.

Lifeng Chen, Jiner Wang, Zihao Pan, Beier Zhu, Xiaofeng Yang, and Chi Zhang. Detail++: Training-free detail enhancer for text-to-image diffusion models. *arXiv preprint arXiv:2507.17853*, 2025b. URL https://arxiv.org/abs/2507.17853.

Ernie Chu and Vishal M. Patel. Not all tokens need 40 steps: Heterogeneous step allocation in diffusion transformers for efficient video generation. *arXiv preprint arXiv:2605.06892*, 2026. URL https://arxiv.org/abs/2605.06892.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12606–12633. PMLR, 2024. URL https://proceedings.mlr.press/v235/esser24a.html.

Jiarui Fang, Jinzhe Pan, Aoyu Li, Xibo Sun, and Jiannan Wang. Pipefusion: Patch-level pipeline parallelism for diffusion transformers inference. In *Advances in Neural Information Processing Systems*, volume 38, pp. 109089–109110, 2025. doi: 10.52202/085713-3289. URL https://papers.nips.cc/paper_files/paper/2025/hash/8da04a60948be713dc766f0c7e3a5b1f-Abstract-Conference.html.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations*, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Danfeng Guo, Sanchit Agarwal, Yu-Hsiang Lin, Jiun-Yu Kao, Tagyoung Chung, Nanyun Peng, and Mohit Bansal. Improving faithfulness of text-to-image diffusion models through inference intervention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4077–4086, 2025.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966. doi: 10.1109/TSSC.1966.300074.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20406–20417, 2023.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025. doi: 10.1109/TPAMI.2025.3531907.

Shantanu Jaiswal, Mihir Prabhudesai, Nikash Bhardwaj, Zheyang Qin, Amir Zadeh, Chuan Li, Katerina Fragkiadaki, and Deepak Pathak. Iterative refinement improves compositional image generation. *arXiv preprint arXiv:2601.15286*, 2026.

Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Training-free mixed-resolution latent upsampling for spatially accelerated diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18673–18682, 2026.

Xu Jiang, Bin Chen, Gehui Li, Yule Duan, Ronggang Wang, and Jian Zhang. OctoT2I: A self-evolving agentic text-to-image router. *arXiv preprint arXiv:2606.01803*, 2026. URL https://arxiv.org/abs/2606.01803.

Sanghyun Jo, Wooyeol Lee, Ziseok Lee, Jonghyun Choi, Jaesik Park, and Kyungsu Kim. ISAC: Training-free instance-to-semantic attention control for multi-instance generation, 2025. URL https://arxiv.org/abs/2505.20935. Accepted to ECCV 2026.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Suhyeon Lee and Jong Chul Ye. PromptLoop: Plug-and-play prompt refinement via latent feedback for diffusion model alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41859–41869, 2026.

Binglei Li, Mengping Yang, Zhiyu Tan, Junping Zhang, and Hao Li. TexTailor: Inference-time textual guidance tailoring for multimodal diffusion transformers, 2026a. URL https://arxiv.org/abs/2601.02211. To appear in ECCV 2026.

Chunhan Li, Qifeng Wu, Jia-Hui Pan, Ka-Hei Hui, Jingyu Hu, Yuming Jiang, Bin Sheng, Xihui Liu, Wenjuan Gong, and Zhengzhe Liu. coDrawAgents: A multi-agent dialogue framework for compositional image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9802–9812, 2026b. URL https://openaccess.thecvf.com/content/CVPR2026F/papers/Li_coDrawAgents_A_Multi-Agent_Dialogue_Framework_for_Compositional_Image_Generation_CVPRF_2026_paper.pdf.

Mingcheng Li, Xiaolu Hou, Ziyang Liu, Dingkang Yang, Ziyun Qian, Jiawei Chen, Jinjie Wei, Yue Jiang, Qingyao Xu, and Lihua Zhang. MCCD: Multi-agent collaboration-based compositional diffusion for complex text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13263–13272, 2025.

Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Kai Li, and Song Han. DistriFusion: Distributed parallel inference for high-resolution diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7183–7193, 2024. URL https://openaccess.thecvf.com/content/

CVPR2024/html/Li_DistriFusion_Distributed_Parallel_Inference_for_High-Resolution_Diffusion_Models_CVPR_2024_paper.html.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22511–22521, 2023.

Long Lian, Boyi Li, Adam Yala, and Trevor Darrell. LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *Transactions on Machine Learning Research*, 2024.

Han Liang, Jiahui Zhou, Zicheng Zhou, Xiaoxi Zhang, and Xu Chen. STADI: Fine-grained step-patch diffusion parallelism for heterogeneous GPUs. *arXiv preprint arXiv:2509.04719*, 2025. URL https://arxiv.org/abs/2509.04719.

Bowen Lin, Fanjiang Ye, Yihua Liu, Zhenghui Guo, Boyuan Zhang, Weijian Zheng, Yufan Xu, Tiancheng Xing, Yuke Wang, and Chengming Zhang. SDiT: Semantic region-adaptive for diffusion transformers. *arXiv preprint arXiv:2601.12283*, 2026.

Zhuohan Liu, Wujian Peng, Yitong Chen, and Zuxuan Wu. Compositional text-to-image generation via region-aware bimodal direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36604–36614, 2026a.

Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2346–2356, 2026b.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. DeepSeek-VL: Towards real-world vision-language understanding, 2024. URL https://arxiv.org/abs/2403.05525.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Scaling inference time compute for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2523–2534, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ma_Scaling_Inference_Time_Compute_for_Diffusion_Models_CVPR_2025_paper.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15762–15772, 2024.

Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7932–7942, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Phung_Grounded_Text-to-Image_Synthesis_with_Attention_Refocusing_CVPR_2024_paper.html.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.01952.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. doi: 10.1109/CVPR52688.2022.01042. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Stability AI. Introducing stable diffusion 3.5. Stability AI, October 2024. URL https://stability.ai/news-updates/introducing-stable-diffusion-3-5. Accessed 2026-09-26.

Wenhao Sun, Ji Li, and Zhaoqiang Liu. Just-in-time: Training-free spatial acceleration for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40643–40652, 2026.

Paulo Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, September 2007. doi: 10.1109/TAC.2007.904277.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024. doi: 10.1109/CVPR52733.2024.00786. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wallace_Diffusion_Model_Alignment_Using_Direct_Preference_Optimization_CVPR_2024_paper.html.

Junjie Wang, Xinghua Lou, Jason Li, Ye Tian, Keyu Chen, Yulin Li, Bin Kang, Jacky Mai, Yanwei Li, Zhuotao Tian, and Liqiang Nie. Benchmarking and evolving reason-reflect-rectify for reflective visual generation, 2026. URL https://arxiv.org/abs/2605.19639.

Tsung-Han Wu, Long Lian, Joseph E. Gonzalez, Boyi Li, and Trevor Darrell. Self-correcting LLM-controlled diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6327–6336, 2024. doi: 10.1109/CVPR52733.2024.00605. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_Self-correcting_LLM-controlled_Diffusion_Models_CVPR_2024_paper.html.

Xiaomi MiMo Team. MiMo-V2.5. Xiaomi MiMo Model Release, 2026. URL https://mimo.mi.com/models/en-US/mimo-v2.5. Accessed 2026-09-26.

Jinheng Xie, Yuexiang Li, Yawen Huang, Haozhe Liu, Wentian Zhang, Yefeng Zheng, and Mike Zheng Shou. BoxDiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7452–7461, 2023.

Zewei Xin, Qinya Li, Chaoyue Niu, Fan Wu, and Guihai Chen. Adaptive routing of text-to-image generation requests between large cloud model and light-weight edge model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19482–19491, 2025.

Bangji Yang, Ruihan Guo, Jiajun Fan, Chaoran Cheng, and Ge Liu. M3: High-fidelity text-to-image generation via multi-modal, multi-agent and multi-round visual reasoning. *arXiv preprint arXiv:2602.06166*, 2026a.

Jiacheng Yang, Jun Wu, Yaoyao Ding, Zhiying Xu, Yida Wang, and Gennady Pekhimenko. Swiftfusion: Scalable sequence parallelism for distributed inference of diffusion transformers on GPUs. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pp. 1037–1050, 2026b. doi: 10.1145/3786335.3813174. URL https://arxiv.org/abs/2601.20273.

Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56704–56721. PMLR, 2024.

APPENDIX

The appendix provides supplementary materials that support the main paper. The contents are organized as follows:

- **Appendix A: Expanded Related Work.** Provides a more comprehensive discussion of related studies and positions our work within the broader literature.
- **Appendix B: Detailed Theoretical Analysis.** Presents additional theoretical formulations, derivations, proofs, and supporting analysis omitted from the main text for brevity.
- **Appendix C: Cost Analysis.** Provides a detailed analysis of the computational and operational costs of the proposed approach, together with relevant comparisons and implementation considerations.
- **Appendix D: Additional Experiments.** Reports supplementary experimental settings, implementation details, additional results, and analyses that complement the experiments presented in the main paper.
- **Appendix E: Pseudocode.** Summarizes the complete inference procedure.

A RELATED WORK

This appendix expands the discussion in Sec. 2. We first review compositional control, adaptive diffusion inference, and feedback-guided agents, then discuss model adaptation, distributed execution, the theoretical basis of selective review, and evaluation. These distinctions matter because methods can improve composition through different mechanisms: changing the generator, improving its conditions, revising a generated image, or changing which inference operations are executed. Our comparison focuses on the unit of control, the feedback used to update it, and the computation required to obtain that feedback, rather than treating all inference-time interventions as the same form of scheduling.

A.1 COMPOSITIONAL TEXT-TO-IMAGE GENERATION AND REGIONAL CONTROL

Attention-based control. Early training-free approaches improve prompt adherence by modifying how text conditions influence denoising. Structured Diffusion Guidance uses linguistic structure to separate the conditioning associated with individual concepts and reduce incorrect composition (Feng et al., 2023). Attend-and-Excite intervenes in cross-attention to strengthen neglected subjects and address subject omission (Chefer et al., 2023). These methods show that a pretrained generator can fail even when the prompt contains the necessary information: the problem can lie in how that information is used during sampling. Attention Refocusing further constrains self- and cross-attention with layouts to improve object placement and spatial organization (Phung et al., 2024). ISAC distinguishes early instance formation from subsequent semantic binding: it first stabilizes class-agnostic layouts using self-attention and then binds cross-attention semantics to those instances (Jo et al., 2025). This distinction is relevant to scheduling because different controls need not be equally useful at every denoising phase. However, attention strength or separation is not itself a verification that a requested relation or count has been satisfied.

Spatial grounding and regional generation. GLIGEN adds trainable grounding components to a frozen diffusion backbone, allowing generation to respond to grounded concepts and spatial inputs (Li et al., 2023). BoxDiff instead imposes box-based constraints during sampling without additional parameter training (Xie et al., 2023). The two methods therefore offer spatial control through different deployment paths: freezing the original generator does not imply that every added component is training-free. MultiDiffusion coordinates multiple diffusion predictions through shared spatial constraints and supports region-specific conditions without retraining (Bar-Tal et al., 2023). Such fusion methods provide a useful execution primitive for composition, but evaluating several conditions can add work beyond the global denoising path. The relevant cost depends on the active conditions, their spatial resolution, and their duration, not only on the number of global sampling steps. Moreover, a relation can require joint control of multiple entities; independent object boxes need not encode the intended interaction.

Method family	What is adapted	Decision signal	Relation to DIFFSCHED
Regional / compositional control BoxDiff (Xie et al., 2023) MultiDiffusion (Bar-Tal et al., 2023) MCCD (Li et al., 2025)	Spatial constraints, regional conditions, or regional diffusion branches	Prompt decomposition, layouts, masks, and scene structure	Provides regional-control primitives and structured semantic targets that **DIFFSCHED** schedules over time.
Conditioning adaptation Detail++ (Chen et al., 2025b) TexTailor (Li et al., 2026a)	Prompt detail or stage-specific textual conditioning	Conditioning structure and denoising stage	Adapts conditioning rather than jointly scheduling regional execution and semantic review.
Semantic / agentic feedback SLD (Wu et al., 2024) T2I-Copilot (Chen et al., 2025a) coDrawAgents (Li et al., 2026b) M3 (Yang et al., 2026a)	Layout correction, editing, regeneration, or iterative refinement	Language-model reasoning and visual feedback	Supports adaptive semantic reasoning, but generally acts at generation, editing, or refinement granularity rather than scheduling regional denoising intervals.
Within-trajectory feedback Inference Intervention (Guo et al., 2025) PromptLoop (Lee & Ye, 2026)	Latent restoration, corrective control, or prompt updates	Intermediate generation states and semantic feedback	Closest in feedback timing. **DIFFSCHED** additionally decides whether semantic feedback is worth acquiring and how much regional computation follows.
Adaptive diffusion computation RAS (Liu et al., 2026b) RALU (Jeong et al., 2026) SDiT (Lin et al., 2026) HSA (Chu & Patel, 2026)	Region, resolution, token, or denoising-step allocation	Model dynamics, spatial complexity, or artifact indicators	Motivates non-uniform computation, but does not directly diagnose explicit prompt requirements. HSA operates on video generation.
Distributed diffusion execution DistriFusion (Li et al., 2024) PipeFusion (Fang et al., 2025) STADI (Liang et al., 2025) SwiftFusion (Yang et al., 2026b)	Patch, layer, step, or sequence placement across devices	Hardware capacity, topology, workload, and communication	Complementary systems layer. Requirement-level schedules could inform future device allocation, but multi-device execution is not evaluated here.
DIFFSCHED	**Requirement-level regional actions, duration, strength, and review**	**Regional–global residuals and selective semantic diagnosis**	**Jointly revises regional computation and semantic-information acquisition over short execution intervals.**

Table 4: Taxonomy of related inference-time mechanisms. Methods are ordered from semantic generation control to computation and systems scheduling. Categories denote their primary mechanism and are not mutually exclusive.

Language-guided scene decomposition. LLM-grounded Diffusion translates text into captioned layouts before generating images with a pretrained diffusion model (Lian et al., 2024). RPG combines recaptioning, regional planning, and complementary regional diffusion, and also supports iterative generation and editing (Yang et al., 2024). MCCD assigns scene-understanding tasks to specialized agents, then uses their descriptions and spatial information in hierarchical compositional diffusion with regional enhancement (Li et al., 2025). These methods establish that language and multimodal models can supply actionable structure rather than merely rewrite a prompt. They also provide important antecedents for the structured requirement set and semantic groups used by **DIFFSCHED**. Our distinction is not the use of agents or decomposition itself, but the subsequent allocation of regional execution and the decision to acquire further semantic observations. It would be inaccurate to describe all language-guided methods as static, since RPG already incorporates a feedback loop.

Conditioning schedules and detail refinement. Some recent methods modify when or where conditioning enters the generator. Detail++ uses Progressive Detail Injection to introduce simplified-to-detailed prompts across denoising stages, together with attention-based alignment to preserve layout (Chen et al., 2025b). TexTailor examines the contribution of text guidance across transformer blocks and tailors that guidance at inference time (Li et al., 2026a). These are related forms of selective intervention, even though their controlled units differ from semantic-group denoiser branches. Accordingly, the meaningful distinction is not "scheduled" versus "unscheduled" generation in an unrestricted sense. It is whether the method jointly revises requirement-level regional work and the semantic inspection used to decide that work. Conditioning schedules and regional refinement can also supply actions for such a scheduler when their implementation is compatible with the backbone.

A.2 Efficient and Spatially Adaptive Diffusion Inference

Solvers, learned acceleration, and feature reuse. DPM-Solver reduces the evaluations required to integrate the diffusion ODE through a specialized high-order solver, without additional training (Lu et al., 2022). Consistency Models take a different route, learning a mapping that supports one- or few-step generation; they can be trained directly or distilled from a diffusion model (Song et al., 2023). DeepCache exploits temporal redundancy by reusing high-level U-Net features while updating other features during denoising (Ma et al., 2024). These approaches respectively change numerical integration, the learned generative mapping, or the amount of recomputed model work. They need not change the semantic decomposition of a prompt. In principle, some could be combined with regional scheduling, but compatibility is not automatic: altered solver steps, features, or prediction statistics can change the meaning and calibration of a residual-based trigger.

Spatially selective execution. RAS uses the spatial and temporal persistence of model focus to update selected regions while reusing predictions elsewhere (Liu et al., 2026b). RALU addresses mixed-resolution generation, including aliasing and noise–timestep mismatch, and selectively upsamples regions that need higher-resolution processing (Jeong et al., 2026). JiT uses sparse anchor-token computation and deterministic micro-flow updates to support training-free spatial acceleration of diffusion transformers (Sun et al., 2026). These methods differ in how they preserve information outside actively computed regions. Skipping a full model evaluation is not necessarily equivalent to freezing that region's latent state. Their shared implication for our problem is that useful inference work is spatially uneven, while global consistency still requires information to flow between regions.

SDiT is particularly close in its explicit regional scheduling. It combines semantic-aware clustering, complexity-driven regional updates, and boundary-aware refinement without model retraining (Lin et al., 2026). Therefore, neither regional adaptation nor the use of semantic structure alone is a sufficient novelty claim for **DIFFSCHED**. The distinction lies in the scheduling target and observation policy: SDiT allocates backbone work from regional complexity, whereas our framework attaches additional interventions to explicit prompt requirements and decides when a Review Agent should inspect those requirements. A visually simple region can still contain an unresolved count or relation, and a visually complex region need not require another semantic review. These are motivations for requirement-level control, not a claim that complexity-based methods are universally inferior.

Token-dependent denoising duration. Heterogeneous Step Allocation assigns different denoising durations to spatiotemporal tokens using velocity dynamics (Chu & Patel, 2026). Its key–value cache preserves access to full-sequence context, while a cached Euler update advances skipped-token

latents without additional model evaluations. HSA is a video-generation method, but it provides a relevant example of varying computation across both space and time. Its allocation signal concerns token dynamics rather than an explicit diagnosis of unsatisfied image-prompt requirements. Together with regional acceleration, it supports the motivation that not every pixel or token needs the same refinement effort. This does not mean that backgrounds may violate the prompt: the desired outcome remains a coherent, faithful image, with computation concentrated where further work is useful.

Inference-time search. Additional test-time computation can also be spent on searching for better samples rather than extending a single denoising path. Ma et al. study inference-time scaling through noise search and verifier-guided candidate selection (Ma et al., 2025). This changes the comparison from "how many steps?" to "which inference operation is worth its cost?" Candidate search, model routing, regional control, and semantic review consume different resources. A scheduler that reduces regional evaluations therefore should not be credited with reducing total inference cost unless search, verification, and other policy-dependent work are included in the accounting.

A.3 Multi-Agent Reasoning and Feedback-Guided Adaptive Generation

Planning, checking, and editing. SLD uses language-model feedback to detect and correct compositional errors in generated images, forming a self-correction loop without additional training (Wu et al., 2024). T2I-Copilot separates prompt interpretation, generation, and output evaluation, allowing quality feedback to guide iterative regeneration (Chen et al., 2025a). These systems show why semantic reasoning is useful beyond initial prompt parsing. They also illustrate a different feedback granularity from an intermediate diffusion checkpoint: evaluating and repairing an output can involve another generation or editing operation rather than a short continuation of the existing regional schedule.

More recent agent frameworks make the representation of requirements and their dependencies increasingly explicit. coDrawAgents combines task routing, priority-based object grouping, planning against an evolving canvas, and layout checking before painting (Li et al., 2026b). M3 organizes multimodal reasoning into planning, checking, refinement, editing, and verification across multiple rounds (Yang et al., 2026a). These approaches already incorporate prioritization and adaptation; they should not be described merely as one-time scene parsers. Their relation to **DIFFSCHED** is instead that structured diagnosis can inform a sequence of actions, while our scheduler operates on regional computation intervals and explicitly gates acquisition of that diagnosis. Agent-assessed improvement should also be distinguished from a guarantee of improvement under an independent evaluation metric.

Iterative refinement has been studied directly as an alternative to allocating all additional computation to parallel samples. Jaiswal et al. use a multimodal critic to guide sequential correction of compositional images and compare that process with parallel generation and selection (Jaiswal et al., 2026). R3-Bench distinguishes identifying an error from proposing an effective repair, and R3-Refiner uses reinforcement learning to improve reflective correction (Wang et al., 2026). These results motivate separating the Review Agent from the Scheduler: an accurate failure description is useful only if it leads to an appropriate, affordable action. They do not imply that any particular review frequency or scheduling policy is optimal.

Feedback within an ongoing trajectory. Inference Intervention detects structural problems during denoising and can resume from an earlier latent with revised layout information (Guo et al., 2025). Dynamic VLM-Guided Negative Prompting uses intermediate image predictions to update negative conditions at selected sampling stages (Chang et al., 2025). PromptLoop trains a multimodal prompt-refinement policy with reinforcement learning so that prompts can be updated from latent feedback during sampling, while leaving the image generator's weights unchanged (Lee & Ye, 2026). These methods are closer to our feedback setting than output-only regeneration because their observations affect an unfinished sample. They differ in the action being revised: layout and rollback, negative conditioning, or prompt text, rather than a common requirement-level allocation of regional actions and durations.

Observation cost and execution cost. Feedback-guided methods raise two distinct decisions: whether the current action should continue, and whether another semantic observation would change

the next action. The residual trigger in **DIFFSCHED** addresses the first using predictions already available from active branches, and reserves semantic inspection for states where the cheap signal is insufficient or a mandatory check is due. This is narrower than treating the residual as a semantic score. It also differs from assuming every review is valuable: its usefulness depends on observability, diagnosis quality, available corrective actions, and remaining sampling time. Review overhead is deployment-dependent; it should be measured rather than presumed to dominate every system.

A.4 Learning, Self-Evolution, and Deployment Requirements

What requires adaptation? Diffusion-DPO adapts a generator to preference data using a diffusion-specific preference objective (Wallace et al., 2024). BiDPO extends preference-based compositional alignment through bimodal preferences and region-aware optimization (Liu et al., 2026a). These methods modify the trained generator, whereas GLIGEN trains added grounding components and PromptLoop trains a feedback policy while retaining a frozen image backbone (Li et al., 2023; Lee & Ye, 2026). Thus, "frozen backbone," "training-free controller," and "plug-and-play deployment" describe different properties. An inference-time interface can be convenient even when its checkpoint or controller was produced by substantial method-specific training.

OctoT2I introduces another preparation mechanism: a self-evolving Propose–Solve–Evaluate–Learn process builds a capability knowledge base for stateful, multi-round tool routing (Jiang et al., 2026). This is not synonymous with parameter fine-tuning, but it is additional preparation that can affect deployment comparisons. RouteT2I instead considers routing an entire text-to-image request between a lightweight edge generator and a larger cloud model using predicted quality and cost (Xin et al., 2025). Both are relevant to allocation, but their decision unit is a model or tool invocation rather than an additional regional prediction within one diffusion trajectory.

Comparison with plug-and-play scheduling. For this paper, it is useful to distinguish methods requiring task-specific training or offline preparation, inference-time generation/control methods, and policies that explicitly allocate regional diffusion work. These are operational comparison categories, not mutually exclusive claims about all capabilities of each system. A prepared knowledge base can be distributed as a plug-and-play artifact, and a training-free method can still require prompt design, profiling, or calibration. **DIFFSCHED** itself uses residual calibration; it should not be described as requiring no preparation whatsoever. The most direct scheduling comparison therefore holds the generator, semantic representation, regional primitives, and evaluation protocol fixed, then varies the allocation and review policy. Comparisons with broader generation systems remain informative, but do not isolate that mechanism on their own.

A.5 Distributed Inference and Prospective Multi-Device Collaboration

Execution across devices. DistriFusion distributes high-resolution diffusion computation over image patches and reuses context from previous steps to overlap communication with computation (Li et al., 2024). PipeFusion partitions patches and model layers to construct a pipeline for diffusion-transformer inference, exploiting inter-step similarity to reduce pipeline overhead (Fang et al., 2025). STADI jointly allocates step and patch workloads to accommodate heterogeneous GPU capabilities (Liang et al., 2025). SwiftFusion addresses distributed sequence parallelism through topology-aware partitioning and communication optimizations, including Torus Attention and one-sided communication (Yang et al., 2026b). These works expose hardware and communication constraints that are absent from a simple count of denoiser evaluations. Their objective is to execute a workload efficiently, not necessarily to decide whether a particular semantic requirement warrants additional intervention.

Semantic allocation and device placement. Requirement-level scheduling could complement these execution systems by exposing which regional actions are active, which can terminate, and when a semantic review is needed. A future runtime could use that information when placing work across devices or deciding whether communication is justified by the expected remaining computation. This is a prospective extension of **DIFFSCHED**, not a multi-device capability evaluated in this paper. There are important dependencies: regional branches share the current latent, fusion can couple overlapping supports, and a review can change subsequent work. Treating each region as

an independent job would therefore be incorrect. Hardware heterogeneity, batching opportunities, context exchange, and synchronization must be considered together with semantic priority.

The distinction also clarifies why regional NFE savings cannot establish a multi-device speedup. Shortening one branch may help only if it removes work from the critical path or releases resources that other branches can use. Conversely, dispatching a small intervention to another device can cost more in communication than it saves in computation. Distributed diffusion research therefore motivates explicit schedules, but the benefit of coupling those schedules to device placement requires a separate systems evaluation.

A.6 Event-Triggered Computation, Information Value, and Calibration

Event-triggered control. Tabuada studies event-triggered execution of stabilizing control tasks as an alternative to uniformly periodic updates (Tabuada, 2007). The useful connection is the separation between inexpensive monitoring and a more costly control update. However, a stability guarantee for a specified dynamical system does not transfer directly to semantic image correctness. In **DIFFSCHED**, the monitored residual measures regional/global prediction disagreement, and semantic utility must be related to that signal through additional assumptions and calibration. The control analogy motivates the architecture; it is not a substitute for validating those assumptions.

Value of semantic observations. Howard's information-value formulation evaluates an observation by the improvement it enables in subsequent decisions (Howard, 1966). This differs from measuring uncertainty or information content alone: a Reviewer can be uncertain about a detail even when no available action would profitably correct it. Conversely, a small but actionable observation can be valuable. The distinction motivates our separate treatment of continuation utility and review value. The theoretical statements concern modeled checkpoint-level decisions; they do not assert that an approximate Reviewer–Scheduler implements an optimal policy or that every high residual indicates a semantic failure. Joint stopping and trajectory-level statements need the additional action-value and continuation conditions specified in the proof appendix.

Calibration and adaptive trajectories. Conformal prediction provides a related framework for finite-sample marginal uncertainty guarantees under appropriate exchangeability conditions (Angelopoulos & Bates, 2023). Distribution-free conditional coverage, however, is substantially more restrictive and cannot generally be obtained in its strongest form without further assumptions (Barber et al., 2021). These distinctions matter when fitting upper envelopes for a residual-based decision: coverage over a held-out population is not a pointwise guarantee at every future adaptive checkpoint. We discuss these tools as relevant uncertainty methodology, not as a claim that **DIFFSCHED** implements a conformal guarantee. Likewise, raising a threshold skips at least as many reviews on a common replayed score sequence, but new closed-loop runs can visit different states. An observed online review-count trend must remain an empirical result unless stronger coupling conditions are established.

A.7 Compositional Evaluation and Efficiency Accounting

Requirement-level evaluation. TIFA converts prompt content into visual questions and evaluates faithfulness through question answering, providing more localized diagnostics than a single image–text similarity score (Hu et al., 2023). GenEval evaluates object-centric properties such as co-occurrence, position, count, and color with detection-based checks (Ghosh et al., 2023). T2I-CompBench++ broadens compositional testing across attribute binding, object relationships, numeracy, and complex compositions (Huang et al., 2025). GenEval 2 expands primitive coverage and compositional difficulty while examining benchmark drift and introducing Soft-TIFA-based evaluation (Kamath et al., 2025). These benchmarks support requirement-level analysis, but their evaluators have their own error modes; no single score establishes that every part of a prompt is correct.

For a feedback-driven generator, the runtime Reviewer and the final evaluator also have different roles. The Reviewer supplies a diagnosis that can change execution, whereas evaluation measures the resulting image against the original request. Reporting only the controller's own confidence would conflate those roles. Independent evaluation, category-level breakdowns, and prompt-matched visual

examples help distinguish successful generation from a controller merely accepting its own output. Case studies should include the original prompt, relevant requirements, and the actual schedule or review decisions when those decisions are used to explain an outcome. Attractive images without recorded execution do not establish why the scheduler helped.

Which cost is reduced? A quality–cost comparison should identify the measured resource. Denoiser evaluations describe algorithmic diffusion work; elapsed diffusion GPU time also reflects batch shape, resolution, caching, and implementation. End-to-end latency includes semantic planning, preview decoding, review, rescheduling, communication, and all repeated computation on the generation path. The acceleration mechanisms above change different parts of that path, so equal NFE does not imply equal runtime (Ma et al., 2024; Li et al., 2024). Similarly, token counts describe semantic-model workload but are not a hardware-independent measure of latency. For direct scheduling comparisons, we use each policy's actual execution cost rather than forcing all policies to consume a hard regional budget. Offline training, knowledge acquisition, and calibration are separate preparation costs and should not be confused with prompt-to-image latency or silently attributed to the same measurement.

A.8 Positioning of DIFFSCHED

Table 4 summarizes the distinction between changing conditioning, allocating backbone computation, obtaining semantic feedback, and placing computation on devices. Several methods combine more than one mechanism; the table identifies their principal intervention rather than assigning exclusive capabilities. In particular, existing work already includes closed-loop editing, semantic grouping, regional scheduling, and adaptive token execution. Our contribution is not any one of these concepts in isolation. **DIFFSCHED** couples explicit requirement-level regional actions with a two-level observation policy: inexpensive prediction residuals screen continuation decisions, and semantic review is acquired when a more informative diagnosis is needed or a mandatory check applies.

This positioning also defines the scope of the evidence. Comparisons with a static semantic schedule test online revision while retaining the initial semantic representation; comparisons with regional heuristics test the choice of allocation signal; and trigger ablations test selective observation. Broader baselines establish context across generation strategies, not universal superiority over every adaptation or execution mechanism. The residual remains an execution signal rather than a semantic correctness score, and the multi-device extension remains prospective. Keeping these distinctions explicit allows the empirical and theoretical results to support the joint scheduling claim without implying that all forms of adaptive generation are new.

B Proofs for the Dual-Certificate Trigger

This appendix gives the statements and proofs supporting Sec. 3.4. We retain the distinction between regional stopping, the value of semantic information, and the stronger conditions required for joint policy improvement. A calibrated residual supplies a certificate only at states where the assumed upper bounds hold. Neither the algebra below nor fitting an envelope to a finite sample establishes those bounds everywhere.

B.1 Notation and Checkpoint-Level Objective

Let c_q be a checkpoint and $\mathcal{I}_q = [c_q, c_{q+1})$ the following interval. The active semantic groups are indexed by $\mathcal{A}_q$. For $u_m \in \mathcal{A}_q$, let $k \in \{0, \dots, H_q\}$, with $H_q = |\mathcal{I}_q| \geq 1$, be the number of additional regional evaluations. This integer range is an analytical relaxation of the implementation's allocation grid $\mathcal{K}_q = \mathcal{K} \cap \{0, \dots, H_q\}$, with $0 \in \mathcal{K}_q$. Any coordinate-wise comparison fixes the positions of those evaluations, the other actions, and the continuation rule; changing these choices changes the utility function being compared.

Let x_q represent information available to the decision rule without a new semantic review, including the execution summaries needed for the decision. The semantic observation Y_q is additional information supplied by the Reviewer. Conditional utilities below use a common model of outcomes before and after observing Y_q. They are not values assigned to a review merely because the Reviewer reports high confidence.

The coefficients $\lambda_{\mathrm{D}} > 0$ and $\lambda_{\mathrm{R}} > 0$ price, respectively, one additional regional evaluation and the current semantic observation in the same units as utility. They are analytical cost weights, not raw seconds or token counts. For group u_m, let $U_{m,q}(k)$ be its expected utility, including the chosen continuation value. The local objective is

$$J^{\mathrm{D}}_{m,q}(k) = U_{m,q}(k) - \lambda_{\mathrm{D}} k. \tag{A1}$$

Assumption A1 (Diminishing regional utility). For each modeled active group,

$$\Delta U_{m,q}(k+1) \le \Delta U_{m,q}(k), \qquad \Delta U_{m,q}(k) = U_{m,q}(k) - U_{m,q}(k-1), \tag{A2}$$

for $1 \le k < H_q$. Marginal utilities are defined for $1 \le k \le H_q$ and may be negative. This assumption concerns each specified one-dimensional continuation problem; it does not assume that overlapping regional controls are independent or that their utilities add.

B.2 Regional Stopping

Proposition A1 (Marginal-gain stopping rule). Under Assumption A1,

$$K^*_{m,q} = \max(\{1 \le k \le H_q : \Delta U_{m,q}(k) \ge \lambda_{\mathrm{D}}\} \cup \{0\}) \tag{A3}$$

is the largest optimizer of Eq. A1 on $\{0, \dots, H_q\}$. If $\Delta U_{m,q}(1) < \lambda_{\mathrm{D}}$, zero is uniquely optimal. If $\Delta U_{m,q}(1) = \lambda_{\mathrm{D}}$, zero is still optimal, but a positive allocation can tie and the largest optimizer need not be zero.

Proof. For $1 \le k \le H_q$,

$$\begin{aligned} J^{\mathrm{D}}_{m,q}(k) - J^{\mathrm{D}}_{m,q}(k-1) &= U_{m,q}(k) - U_{m,q}(k-1) - \lambda_{\mathrm{D}} \\ &= \Delta U_{m,q}(k) - \lambda_{\mathrm{D}}. \end{aligned} \tag{A4}$$

These increments are non-increasing. If no increment is nonnegative, every positive allocation has lower objective than zero. Otherwise, let k^* be the last index with a nonnegative increment. The objective is non-decreasing up to k^* and strictly decreasing afterward, so k^* is its largest optimizer. If the first increment is strictly negative, so are all subsequent increments, making zero the unique optimizer. If the first is zero, a zero-increment plateau can tie with zero; no later increment can be positive. This proves each assertion. □

Corollary A1. If $\Delta U_{m,q}(1) \le \lambda_{\mathrm{D}}$, then $0 \in \arg\max_{0 \le k \le H_q} J^{\mathrm{D}}_{m,q}(k)$. The same holds on any feasible allocation subset containing zero, including $\mathcal{K}_q$. Equation A3 need not itself select a point in $\mathcal{K}_q$; a positive implemented allocation must instead maximize the objective over that grid.

B.3 Semantic Review as Value of Information

Review supplies an observation rather than directly changing the latent, which gives a value-of-information comparison (Howard, 1966). Condition on pre-review information x_q, and let $\mathfrak{A}_q$ be a finite, nonempty action set available both with and without review. An action can encode an entire next-interval schedule. For $a \in \mathfrak{A}_q$, $Q_q(a; x_q, Y_q)$ is the conditional expected downstream utility, net of the cost of the generation action and its continuation but excluding the current review cost. Assume these values are integrable. The no-review value is

$$V^{\mathrm{N}}_q(x_q) = \max_{a \in \mathfrak{A}_q} \mathbb{E}_{Y_q | x_q}[Q_q(a; x_q, Y_q)]. \tag{A5}$$

After review, the action may depend on Y_q, giving

$$V^{\mathrm{R},gross}_q(x_q) = \mathbb{E}_{Y_q | x_q}\left[\max_{a \in \mathfrak{A}_q} Q_q(a; x_q, Y_q)\right]. \tag{A6}$$

The gross value of the observation is

$$\mathcal{I}_q(x_q) = V^{\mathrm{R},gross}_q(x_q) - V^{\mathrm{N}}_q(x_q). \tag{A7}$$

The same action set is essential: any action available without review remains available after review, so the information can be ignored. An implementation that imposes different action sets must include those restrictions explicitly in the comparison.

Proposition A2 (Non-negativity and review criterion). For every such x_q, $\mathcal{I}_q(x_q) \geq 0$. With review cost λ_{R},

$$V_q^{\mathrm{R}}(x_q) = V_q^{\mathrm{R},gross}(x_q) - \lambda_{\mathrm{R}}, \tag{A8}$$

and optimal review-conditioned selection is strictly preferable to optimal no-review selection if and only if

$$\mathcal{I}_q(x_q) > \lambda_{\mathrm{R}}. \tag{A9}$$

At equality the two decisions tie.

Proof. For every fixed a and observation Y_q, $\max_{a'} Q_q(a'; x_q, Y_q) \geq Q_q(a; x_q, Y_q)$. Taking conditional expectations and maximizing over a gives $V_q^{\mathrm{R},gross} \geq V_q^{\mathrm{N}}$ at x_q. Moreover,

$$V_q^{\mathrm{R}}(x_q) - V_q^{\mathrm{N}}(x_q) = \mathcal{I}_q(x_q) - \lambda_{\mathrm{R}},$$

which proves the criterion and the equality case. □

The proposition compares optimal decision rules under a specified observation model. A noisy Reviewer is permitted because its noise is included in the law of Y_q. However, a heuristic Scheduler need not attain either maximum: a rollout using its chosen action measures that policy's value, not necessarily the value of information in Eq. A7.

B.4 Dual Residual Calibration

At a gate-eligible checkpoint, assume $\mathcal{A}_q$ is nonempty and every active score is available. Define

$$s_q = \max_{m \in \mathcal{A}_q} s_{m,q}. \tag{A10}$$

The certificate does not assign a score of zero to an unobserved branch or an empty active set. Such cases require the implementation's explicit no-active-action or mandatory-review rule and are outside this certificate.

Assumption A2 (Monotone dual calibration). There exist non-decreasing functions $\phi_{\mathrm{D}}, \phi_{\mathrm{R}}$ and nonnegative errors $\varepsilon_{\mathrm{D}}, \varepsilon_{\mathrm{R}}$ such that, at the checkpoint states where the certificate is applied,

$$\Delta U_{m,q}(1) \leq \phi_{\mathrm{D}}(s_{m,q}) + \varepsilon_{\mathrm{D}} \quad \text{for every active } m, \tag{A11}$$

$$\mathcal{I}_q(x_q) \leq \phi_{\mathrm{R}}(s_q) + \varepsilon_{\mathrm{R}}. \tag{A12}$$

These are upper-bound assumptions on the target utilities, not consequences of a small diffusion residual alone. In particular, agreement of two incorrect predictions does not establish semantic correctness.

Let $\mathcal{T} \subset [0, \infty)$ be a finite calibration grid. Define

$$\tau_{\mathrm{D}} = \max\{s \in \mathcal{T} : \phi_{\mathrm{D}}(s) + \varepsilon_{\mathrm{D}} \leq \lambda_{\mathrm{D}}\}, \tag{A13}$$

$$\tau_{\mathrm{R}} = \max\{s \in \mathcal{T} : \phi_{\mathrm{R}}(s) + \varepsilon_{\mathrm{R}} \leq \lambda_{\mathrm{R}}\}, \tag{A14}$$

$$\tau_0 = \min(\tau_{\mathrm{D}}, \tau_{\mathrm{R}}). \tag{A15}$$

Both feasible sets are assumed nonempty. Otherwise the certified bypass is disabled rather than assigning an uncertified default threshold. Because the grid is finite, each selected maximum is feasible. The score of a test checkpoint need not belong to the grid: monotonicity extends feasibility to all smaller scores, conditional on Assumption A2 at that checkpoint.

Lemma A1 (Conservative max aggregation). If $s_q \leq \tau_0$, then $s_{m,q} \leq \tau_0$ for every active group and

$$\Delta U_{m,q}(1) \leq \lambda_{\mathrm{D}} \quad \text{for every } m \in \mathcal{A}_q. \tag{A16}$$

Proof. Since $s_{m,q} \leq s_q \leq \tau_0 \leq \tau_{\mathrm{D}}$, Assumption A2, monotonicity, and feasibility of the selected grid point give

$$\Delta U_{m,q}(1) \leq \phi_{\mathrm{D}}(s_{m,q}) + \varepsilon_{\mathrm{D}} \leq \phi_{\mathrm{D}}(\tau_{\mathrm{D}}) + \varepsilon_{\mathrm{D}} \leq \lambda_{\mathrm{D}}.$$

No continuity assumption or attainment of a supremum is needed. □

B.5 Proof of the Dual-Certificate Theorem

Theorem A1 (Dual-certificate trigger). Under Assumptions A1–A2, at a gate-eligible checkpoint with $s_q \le \tau_0$: (i) zero additional evaluations are optimal for each local regional objective $J^{\mathrm{D}}_{m,q}$; and (ii) the best review-conditioned decision after acquisition cost is no better than the best no-review decision, $V^{\mathrm{R}}_q(x_q) \le V^{\mathrm{N}}_q(x_q)$.

Proof. Lemma A1 gives $\Delta U_{m,q}(1) \le \lambda_{\mathrm{D}}$ for each active group. Corollary A1 therefore implies $0 \in \arg\max_k J^{\mathrm{D}}_{m,q}(k)$; it does not assert that the largest optimizer $K^*_{m,q}$ equals zero in a tie. For the second conclusion,

$$\mathcal{I}_q(x_q) \le \phi_{\mathrm{R}}(s_q) + \varepsilon_{\mathrm{R}} \le \phi_{\mathrm{R}}(\tau_{\mathrm{R}}) + \varepsilon_{\mathrm{R}} \le \lambda_{\mathrm{R}}.$$

Proposition A2 then gives $V^{\mathrm{R}}_q(x_q) - V^{\mathrm{N}}_q(x_q) = \mathcal{I}_q(x_q) - \lambda_{\mathrm{R}} \le 0$. □

The theorem makes two local comparisons. It does not identify semantic success, validate high-score corrective actions, or show that stopping all interacting branches maximizes the joint schedule value. The latter requires joint action-value consistency, stated below. The high-score region lacks the certificate and is inspected semantically; a high score is not itself a certificate that review will pay for its cost.

B.6 Review-Frequency Consequence

First fix a protocol in which each candidate checkpoint is visited once and its mandatory flag is fixed. Let $\mathcal{C}_{\mathrm{gate}}$ be the non-mandatory candidates and $N_{\mathrm{mandatory}}$ the mandatory count. For scores generated under a specified policy π_τ, set $Z_q(\tau) = \mathbf{1}\{s^{\pi_\tau}_q > \tau\}$. Then

$$N_{\mathrm{rev}}(\tau) = N_{\mathrm{mandatory}} + \sum_{q \in \mathcal{C}_{\mathrm{gate}}} Z_q(\tau),$$

$$\mathbb{E}_{\pi_\tau}[N_{\mathrm{rev}}(\tau)] = N_{\mathrm{mandatory}} + \sum_{q \in \mathcal{C}_{\mathrm{gate}}} \Pr_{\pi_\tau}(s^{\pi_\tau}_q > \tau). \tag{A17}$$

This identity uses only linearity of expectation. Residuals at different checkpoints need not be independent.

Corollary A2 (Expected review saving). Relative to the review-all count $N_{\mathrm{periodic}} = N_{\mathrm{mandatory}} + |\mathcal{C}_{\mathrm{gate}}|$ on the same fixed candidate protocol,

$$N_{\mathrm{periodic}} - \mathbb{E}_{\pi_{\tau_0}}[N_{\mathrm{rev}}(\tau_0)] = \sum_{q \in \mathcal{C}_{\mathrm{gate}}} \Pr_{\pi_{\tau_0}}(s^{\pi_{\tau_0}}_q \le \tau_0). \tag{A18}$$

Multiplication by a constant λ_{R} gives the saving in the review-count term of the objective, not necessarily the full runtime saving.

Proof. Subtract Eq. A17 from the fixed review-all count and use $1 - \Pr(s_q > \tau_0) = \Pr(s_q \le \tau_0)$ at each candidate. □

If stopping, mandatory flags, or rollback changes which checkpoints are visited, count visits rather than distinct diffusion indices. For a random finite visit set $\mathcal{V}_\pi$, the general identity is

$$\mathbb{E}_\pi[N_{\mathrm{rev}}] = \mathbb{E}_\pi\left[\sum_{v \in \mathcal{V}_\pi} \big(B_v + (1 - B_v)Z_v\big)\right],$$

where B_v marks a mandatory review and Z_v a triggered one. Comparing policies with different visit sets requires this expression, not a shared constant N_{periodic}.

B.7 Threshold Scaling and the Sensitivity Parameter

DIFFSCHED uses

$$\tau_\beta = \beta\tau_0, \qquad \beta > 0. \tag{A19}$$

Corollary A3 (Certified and aggressive regimes). At a state satisfying Assumptions A1–A2, every bypass under $0 < \beta \leq 1$ lies within the default certified region. On a common sequence of scores $\{s_q\}$, candidate visits, and mandatory flags, define $N_{\mathrm{rev}}^{\mathrm{replay}}(\beta) = N_{\mathrm{mandatory}} + \sum_{q \in \mathcal{C}_{\mathrm{gate}}} \mathbf{1}\{s_q > \beta\tau_0\}$. For $\beta_1 \leq \beta_2$,

$$N_{\mathrm{rev}}^{\mathrm{replay}}(\beta_2) \leq N_{\mathrm{rev}}^{\mathrm{replay}}(\beta_1), \tag{A20}$$

so the inequality also holds in expectation under any common score law. For $\beta > 1$, scores in $(\tau_0, \beta\tau_0]$ are not covered by the default certificate. This additional interval is empty when $\tau_0 = 0$.

Proof. For $0 < \beta \leq 1$, $s_q \leq \beta\tau_0$ implies $s_q \leq \tau_0$ because $\tau_0 \geq 0$. For common scores and $\beta_1 \leq \beta_2$, $\{s_q > \beta_2\tau_0\} \subseteq \{s_q > \beta_1\tau_0\}$ pointwise. Summing the corresponding indicators proves Eq. A20; expectations preserve the inequality. □

The common-score qualification is necessary. In fresh closed-loop runs, $\Pr_{\pi_\beta}(s_q^{\pi_\beta} > \beta\tau_0)$ depends on both the threshold and the trajectory induced by previous decisions. For example, let $\tau_0 = 1$ and consider three non-mandatory checkpoints. With $\beta_1 = 0.5$, a score 0.75 at the first checkpoint triggers a correction that makes the remaining scores zero, giving one review. With $\beta_2 = 1$, the first review is skipped and the remaining scores may both become 2, giving two reviews. Thus the gate's nested acceptance sets alone do not prove monotone online review counts. Online counts, latency, and final quality must be evaluated empirically; a non-monotone GM curve does not contradict the replay result.

B.8 Finite-Horizon Composition

A local coordinate certificate does not automatically extend to a joint schedule. For example, a two-branch net objective can satisfy $G(0,0) = 0$, $G(1,0) = G(0,1) = -1$, and $G(1,1) = 1$. Stopping each branch is optimal when the other is stopped, yet their joint continuation is better. This example satisfies one-step coordinate "diminishing returns" but contains a cross-branch complementarity. The following assumption makes explicit what is needed to exclude this gap.

Let the policy objective be

$$\mathcal{J}(\pi) = \mathbb{E}_\pi[F(x_{\mathrm{term}}) - \lambda_{\mathrm{D}} N_{\mathrm{D}} - \lambda_{\mathrm{R}} N_{\mathrm{R}} - C_{\mathrm{other}}]\,. \tag{A21}$$

Here F is bounded terminal semantic utility; N_{D} counts regional evaluations charged at λ_{D}; N_{R} counts semantic reviews; and C_{other} contains policy-dependent costs not already charged, including auxiliary controls, extra observation work, or replayed global steps. Common fixed costs can be omitted. Every cost is expressed in the same objective units and is counted exactly once.

Assumption A3 (Bellman-consistent calibration). The comparison has a finite decision horizon, finite feasible action sets, and integrable costs. Checkpoint information is sufficient for the conditional action values, and all compared actions use the same subsequent policy when their values are evaluated. Assumptions A1–A2 hold with these continuation values on each state at which a replacement is made.

Write a no-review action as $(b, \mathbf{k})$, where $\mathbf{k}$ counts continuations of the current active regional interventions and b contains the remaining choices. Its expected net value is $G_q(b, \mathbf{k}; x_q) = \mathbb{E}[Q_q((b, \mathbf{k}); x_q, Y_q) \mid x_q]$. Removing the current continuations is feasible, and the following sufficient joint condition holds for every feasible $(b, \mathbf{k})$:

$$G_q(b, \mathbf{k}; x_q) - G_q(b, \mathbf{0}; x_q) \leq \sum_{m \in \mathcal{A}_q} \left[J_{m,q}^{\mathrm{D}}(k_m) - J_{m,q}^{\mathrm{D}}(0)\right].$$

Exact separability is one sufficient case, but the displayed upper bound is all that is needed. At a certified state it implies that an optimal no-review action exists with $\mathbf{k} = \mathbf{0}$. The replacement selects such an optimizer. Costs of obtaining the common checkpoint information, including its score, are included identically in the comparison; any extra cost introduced solely by the replacement must be deducted from its gain.

The finite-horizon assumption is immediate for forward sampling without rollback. With rollback, it requires a finite unrolling of chronological checkpoint visits, for example under a finite rollback limit. The state must include the current diffusion index, cache, and remaining rollback allowance. A finite number of diffusion steps alone does not rule out infinitely many rollbacks. These are conditions on the theorem, not an imposed matched experimental computation budget.

Theorem A2 (Sequential composition of certified decisions). Let $\pi^{(0)}$ be a finite reference policy satisfying Assumptions A1–A3. At each replacement, keep later decisions fixed, recompute the corresponding conditional values, and replace a certified checkpoint decision by an optimal no-review action with zero continuation of the current regional branches. Making these replacements in backward decision order yields

$$\mathcal{J}(\pi^{(L)}) \geq \mathcal{J}(\pi^{(L-1)}) \geq \cdots \geq \mathcal{J}(\pi^{(0)}). \tag{A22}$$

This is a conditional policy-improvement statement, not global optimality of the practical Scheduler.

Proof. Fix one replacement and condition on the reached checkpoint information x_q. By Theorem A1, every term $J^{\mathrm{D}}_{m,q}(k_m) - J^{\mathrm{D}}_{m,q}(0)$ is nonpositive. The joint condition in A3 therefore gives $G_q(b, \mathbf{k}; x_q) \leq G_q(b, \mathbf{0}; x_q)$ for every feasible choice. Consequently,

$$\max_b G_q(b, \mathbf{0}; x_q) = \max_{b,\mathbf{k}} G_q(b, \mathbf{k}; x_q) = V^{\mathrm{N}}_q(x_q).$$

For a reference no-review action, its conditional value is at most $V^{\mathrm{N}}_q(x_q)$. For any reference review-conditioned action rule, its value after review cost is at most $V^{\mathrm{R}}_q(x_q)$, which is no greater than $V^{\mathrm{N}}_q(x_q)$ by Theorem A1. Thus the replacement weakly improves conditional value whether or not the reference would review.

The two policies agree before this decision and use the same continuation afterward. Their distribution of pre-decision histories is therefore the same. Applying the law of total expectation to the nonnegative conditional improvement gives $\mathcal{J}(\pi^{(i+1)}) \geq \mathcal{J}(\pi^{(i)})$. Repeating in backward order, with calibration and action values appropriate to the updated continuation, proves Eq. A22. □

Corollary A4. Strict improvement follows if, on a set of reached checkpoint histories of positive probability, either the reference acquires review with $\mathcal{I}_q(x_q) < \lambda_{\mathrm{R}}$, or it takes a no-review action continuing a branch with $\Delta U_{m,q}(1) < \lambda_{\mathrm{D}}$. All other replacements must weakly improve conditional value and satisfy A3. In the second case, A1 makes the corresponding nonzero continuation term strictly negative, and the joint bound makes its removal strictly beneficial. Taking expectations of a nonnegative improvement that is positive on such a set gives $\mathcal{J}(\pi^{(L)}) > \mathcal{J}(\pi^{(0)})$.

Approximate decisions and additional gating cost. If the implemented stop-and-bypass action is at most $\eta_q(x_q)$ below $V^{\mathrm{N}}_q(x_q)$, its comparison with the reference can lose up to $\eta_q(x_q)$. Any extra sensing/gating cost $g^{\mathrm{extra}}_q(x_q)$ not shared by the reference must also be subtracted. Applying the same one-decision argument gives the relaxed bound

$$\mathcal{J}(\pi^{(L)}) \geq \mathcal{J}(\pi^{(0)}) - \sum_{i=0}^{L-1} \mathbb{E}_{d_i}\left[\eta_{q_i}(x_{q_i}) + g^{\mathrm{extra}}_{q_i}(x_{q_i})\right],$$

where d_i is the common distribution of histories reaching the ith replacement under the two policies in that comparison; the summand is zero when no replacement occurs. This bound still assumes valid local calibration and joint consistency. It makes explicit why a small residual alone cannot prove an unconditional improvement of a heuristic implementation.

B.9 Calibration Procedure

Calibration must specify the utility whose upper bounds are required. Fix a bounded terminal score F, the requirement/group weights used to define $U_{m,q}$, the cost coefficients, the admissible action set, and the continuation policy before collecting calibration rollouts. Use the same objective in $U_{m,q}$ and Q_q; for finite-horizon claims, include every policy-dependent continuation cost. The supplied formulation does not fix a numerical choice of these quantities, so none is presumed here.

Regional calibration. For a cached pre-review state and group, run paired continuations with zero and one additional evaluation of the current regional action. Keep the other actions, evaluation position, and continuation policy fixed, and pair randomness where possible. Repeated rollouts estimate the difference in expected utility, $\Delta U_{m,q}(1)$, rather than treating one noisy outcome difference as its exact value. Retain the resulting score–utility pairs to fit a non-decreasing envelope. A joint claim additionally needs the uniform joint bound in A3; coordinate rollouts alone do not verify it.

Review calibration. Equation A7 contains two optimizations. Estimating it requires conditional action values across the admissible schedules and integration over possible review observations, with the same continuation and objective. The current review cost is excluded from the gross information value and is subtracted only in Eq. A8. Separate action-value estimation or action selection from the rollouts used to evaluate the selected actions, and account for Monte Carlo error in the maxima. A single Reviewer–Scheduler rollout compared with a fixed no-review policy instead estimates an implemented-policy difference; it is not automatically equal to, or an upper bound on, the optimal information value.

Suppose the conditional value estimates obey

$$|\widehat{Q}_q(a; x_q, y) - Q_q(a; x_q, y)| \le \eta \quad \text{for all relevant } a, y,$$

and observation expectations are evaluated exactly. Both the maximum and expectation operators preserve this error bound. Hence $|\widehat{\mathcal{I}}_q - \mathcal{I}_q| \le 2\eta$. Numerical estimation of the observation expectation adds its own error. Such uncertainty must be included when using estimated information values to construct an upper envelope.

Envelope fitting and validation. Use separate data for fitting the envelopes, selecting their margins and thresholds, and auditing the final certificate. Checkpoints from one prompt or one trajectory are dependent; splitting at the prompt/trajectory level avoids treating them as independent calibration examples. Select $\tau_{\mathrm{D}}, \tau_{\mathrm{R}}$ on the finite feasible grid of Eqs. A13–A14 and set τ_0 to their minimum. Report calibration sizes, rollout counts, $\lambda_{\mathrm{D}}, \lambda_{\mathrm{R}}$, margins, thresholds, and out-of-sample violations. Residual distributions can change with the backbone, threshold policy, and continuation rule; transfer of an envelope requires validation rather than automatic reuse.

Statistical meaning of a certificate. Let E_{D} be the event that all active-group compute bounds hold at a test checkpoint and E_{R} the event that its review bound holds. If, under a stated test-state distribution, $\Pr(E_{\mathrm{D}}^c) \le \delta_{\mathrm{D}}$ and $\Pr(E_{\mathrm{R}}^c) \le \delta_{\mathrm{R}}$, the union bound gives

$$\Pr(s_q \le \tau_0,\ E_{\mathrm{D}}^c \cup E_{\mathrm{R}}^c) \le \delta_{\mathrm{D}} + \delta_{\mathrm{R}}.$$

This is a marginal bound on an uncertified bypass, not a guarantee for every state or the same error rate conditional on bypass. If $p_{\mathrm{skip}} = \Pr(s_q \le \tau_0) > 0$, the corresponding conditional bound is at most $\min\{1, (\delta_{\mathrm{D}} + \delta_{\mathrm{R}})/p_{\mathrm{skip}}\}$. Individual per-group coverage must be combined across active groups before it can serve as E_{D}. Any statistical procedure used to claim these rates must state its sampling assumptions; empirical envelope fitting alone supplies no distribution-free pointwise or adaptive-trajectory coverage. The theorems above remain conditional statements until the required bounds are justified for the states on which they are used.

C Cost Decomposition and Controller Amortization

DIFFSCHED can reduce regional computation and semantic-review workload, but it also adds prompt structuring, gating, preview decoding, and rescheduling. We characterize when the savings exceed this overhead. Algorithmic diffusion work, additive resource cost, elapsed diffusion GPU time, and end-to-end latency are kept distinct. All statements concern the specified cost metric; none by itself establishes better generation quality.

General cost decomposition. Let π_{ref} be a reference execution policy and π_{ds} the proposed policy. For backbone scale s, let $D_\pi(s)$ and $H_\pi(s)$ be their expected per-image diffusion and controller costs under the same workload distribution and the same additive unit. Diffusion cost includes global and regional evaluations, enabled refinement, and replay caused by rollback. Controller cost includes the remaining operations without double counting. Define

$$C_\pi(s) = D_\pi(s) + H_\pi(s). \tag{16}$$

The identity can also be used for individual runs with all terms interpreted consistently as run-level costs. It is not obtained by adding overlapping wall-clock durations. Write

$$\begin{aligned} \Delta D(s) &= D_{\mathrm{ref}}(s) - D_{\mathrm{ds}}(s), \\ \Delta H(s) &= H_{\mathrm{ds}}(s) - H_{\mathrm{ref}}(s). \end{aligned} \tag{17}$$

The reference may have its own planning and control overhead; neither $H_{\mathrm{ref}} = 0$ nor $\Delta D \ge 0$ is assumed.

Theorem 2 (Controller amortization). Under Eq. 16, **DIFFSCHED** has lower expected additive inference cost exactly when

$$\boxed{\Delta D(s) > \Delta H(s).} \tag{18}$$

Equivalently, the amortization margin

$$M(s) = \Delta D(s) - \Delta H(s) \tag{19}$$

is positive. This is an accounting identity: equality gives equal cost, and no optimality of the scheduler or quality guarantee is implied.

Proof. Subtracting Eq. 16 for the two policies gives $C_{\mathrm{ref}}(s) - C_{\mathrm{ds}}(s) = \Delta D(s) - \Delta H(s) = M(s)$. The strict inequality follows exactly when this difference is positive. □

Corollary 2.1 (Denoiser-equivalent special case). Assume that the only difference in diffusion cost is $\Delta N = N_{\mathrm{ref}} - N_{\mathrm{ds}} > 0$ equivalent evaluations, each with the same additive cost $c_D(s) > 0$, so that $\Delta D(s) = \Delta N\, c_D(s)$. This is an additional cost model, not an identity for arbitrary regional batching, resolutions, or backward passes. If $\Delta H(s) = H(s)$, then

$$\boxed{\Delta N\, c_D(s) > H(s)} \tag{20}$$

is equivalent to lower additive cost. With $N_{\mathrm{ref}} > 0$, define

$$\eta = \frac{\Delta N}{N_{\mathrm{ref}}}, \qquad \rho(s) = \frac{H(s)}{N_{\mathrm{ref}} c_D(s)}. \tag{21}$$

The same condition becomes

$$\boxed{\rho(s) < \eta.} \tag{22}$$

When the per-evaluation cost varies, ΔD must instead be measured or computed from the corresponding weighted operations; a raw NFE reduction is not a wall-clock speedup.

Critical-path latency. Operations on different devices or streams may overlap. For each policy, partition its measured prompt-to-image critical path into diffusion execution L_π^{diff} and the remaining exposed delay X_π, including controller work, communication, and synchronization. These are disjoint critical-path contributions, not sums of all component runtimes. Then

$$L_\pi^{\mathrm{e2e}} = L_\pi^{\mathrm{diff}} + X_\pi. \tag{23}$$

This also holds after taking expectations over the same workload. The exact latency break-even condition is

$$\boxed{\Delta L_{\mathrm{diff}} > \Delta X,} \tag{24}$$

where $\Delta L_{\mathrm{diff}} = L_{\mathrm{ref}}^{\mathrm{diff}} - L_{\mathrm{ds}}^{\mathrm{diff}}$ and $\Delta X = X_{\mathrm{ds}} - X_{\mathrm{ref}}$. Elapsed diffusion GPU time can help diagnose this decomposition, but it need not equal the diffusion contribution to the critical path. The end-to-end measurement remains the source of the latency comparison.

Expected overhead under checkpoint gating. First consider a finite protocol with common, once-visited candidate sets $\mathcal{C}_{\mathrm{mand}}$ and $\mathcal{C}_{\mathrm{gate}}$. For the policy π_τ induced by threshold τ, let $Z_q^\tau = \mathbf{1}\{s_q^{\pi_\tau} > \tau\}$ at gate-eligible candidates. Let g_q^τ be always-incurred checkpoint bookkeeping, including the gate where evaluated, and let h_q^τ be the incremental observation cost when review is requested. The latter includes preview decoding, semantic review, and review-induced scheduling, excluding work already charged elsewhere. Mandatory checkpoints can have zero residual-gate cost if they bypass it. Define

$$\begin{aligned} p_q(\tau) &= \Pr_{\pi_\tau}(Z_q^\tau = 1), \\ \bar{h}_q(\tau) &= \mathbb{E}_{\pi_\tau}[h_q^\tau \mid Z_q^\tau = 1]. \end{aligned} \tag{25}$$

When $p_q(\tau) = 0$, set the product $p_q(\tau)\bar{h}_q(\tau)$ to zero. Let $H_0(\tau)$ be mean one-time planning cost, $G(\tau) = \sum_q \mathbb{E}_{\pi_\tau}[g_q^\tau]$, and let $\bar{h}_q^{\mathrm{mand}}(\tau)$ be the mean mandatory-review cost. The expected controller cost is

$$\begin{aligned} H_{\mathrm{ds}}(\tau) = H_0(\tau) + G(\tau) &+ \sum_{q \in \mathcal{C}_{\mathrm{mand}}} \bar{h}_q^{\mathrm{mand}}(\tau) \\ &+ \sum_{q \in \mathcal{C}_{\mathrm{gate}}} p_q(\tau)\bar{h}_q(\tau). \end{aligned} \tag{26}$$

The conditional mean is necessary because review duration and token use may depend on the state that triggered review. In general $\mathbb{E}[Z_q h_q] \neq \mathbb{E}[Z_q]\mathbb{E}[h_q]$.

Write $H_{\text{fixed}}(\tau) = H_0(\tau) + G(\tau) + \sum_{q \in \mathcal{C}_{\text{mand}}} \bar{h}_q^{\text{mand}}(\tau)$ and $\Delta D(\tau) = D_{\text{ref}} - D_{\text{ds}}(\tau)$. Combining Eqs. 18 and 26 gives

$$\sum_{q \in \mathcal{C}_{\text{gate}}} p_q(\tau) \bar{h}_q(\tau) < \Delta D(\tau) - H_{\text{fixed}}(\tau) + H_{\text{ref}}. \tag{27}$$

This is a break-even relation among expected measured costs, not a hard inference budget or a guarantee that increasing the threshold lowers total cost. The threshold can change both the visited states and later diffusion work.

If all gate-eligible conditional mean review costs equal a common $\bar{h}(\tau) > 0$ and $Q_g = |\mathcal{C}_{\text{gate}}| > 0$, define $\bar{p}(\tau) = Q_g^{-1} \sum_q p_q(\tau)$. The condition becomes

$$\boxed{\bar{p}(\tau) < \frac{\Delta D(\tau) - H_{\text{fixed}}(\tau) + H_{\text{ref}}}{Q_g \bar{h}(\tau)}.} \tag{28}$$

It is exact under equal conditional costs and approximate when those costs are only approximately equal. The right-hand side also depends on the policy. If it is nonpositive, no nonnegative review rate satisfies the strict condition. If $Q_g = 0$, use Eq. 18 directly.

For policies with random, skipped, or repeated checkpoint visits, the general accounting expression is

$$H_\pi = \mathbb{E}_\pi \left[H_{0,\pi}^{\text{run}} + \sum_{v \in \mathcal{V}_\pi} (g_v + Z_v h_v) \right],$$

where $Z_v = 1$ includes mandatory reviews and $\mathcal{V}_\pi$ contains every actual visit, including rollback replays. Applying a common fixed-set formula to policies with different visit counts would omit part of the cost.

Corollary 2.2 (Benefit of triggering over periodic review). Compare a periodic-review policy and a triggered policy on a common fixed candidate protocol. All quantities below are expected additive costs. Let H_π^{base} include planning, mandatory observations, and other controller work excluding the extra gate and gate-eligible reviews. Let G be the extra gate cost paid by the triggered policy, p_q its review probability, and $\bar{h}_q^{\text{per}}$ and $\bar{h}_q^{\text{trig}}$ the respective per-review means, with the latter conditioned on review. Then

$$\begin{aligned} C_{\text{per}} - C_{\text{trig}} = \; & \underbrace{D_{\text{per}} - D_{\text{trig}}}_{\text{diffusion-cost difference}} \\ & + \underbrace{\sum_{q \in \mathcal{C}_{\text{gate}}} (1 - p_q) \bar{h}_q^{\text{per}}}_{\text{skipped-review reference cost}} - G + \Xi, \end{aligned} \tag{29}$$

where

$$\Xi = H_{\text{per}}^{\text{base}} - H_{\text{trig}}^{\text{base}} + \sum_{q \in \mathcal{C}_{\text{gate}}} p_q (\bar{h}_q^{\text{per}} - \bar{h}_q^{\text{trig}}).$$

If base costs and per-review costs are shared, $\Xi = 0$, recovering the three-term decomposition into diffusion savings, skipped-review savings, and gate overhead. Otherwise the correction cannot be discarded.

Proof. Subtract $C_{\text{trig}} = D_{\text{trig}} + H_{\text{trig}}^{\text{base}} + G + \sum_q p_q \bar{h}_q^{\text{trig}}$ from $C_{\text{per}} = D_{\text{per}} + H_{\text{per}}^{\text{base}} + \sum_q \bar{h}_q^{\text{per}}$. At each gate-eligible candidate, add and subtract $p_q \bar{h}_q^{\text{per}}$ to obtain Eq. 29. □

Thus triggering reduces additive cost exactly when the right-hand side is positive. Neither its diffusion-cost term nor Ξ has a predetermined sign. The identity maps to the ablation only after the two variants use consistent workloads and cost definitions. For differing visit protocols, subtract the general visit-sum expressions instead. The common-score monotonicity in Appendix B does not establish a monotone end-to-end cost curve across fresh closed-loop runs.

Backbone scaling. For $\Delta H(s) > 0$, define

$$A(s) = \frac{\Delta D(s)}{\Delta H(s)}. \tag{30}$$

The additive-cost advantage holds exactly when $A(s) > 1$. For the following logarithmic comparison, additionally assume $\Delta D(s) > 0$ and that both differences are differentiable functions of a continuous scale parameter. Then

$$\frac{d \log A(s)}{ds} = \frac{d \log \Delta D(s)}{ds} - \frac{d \log \Delta H(s)}{ds}. \tag{31}$$

The amortization ratio increases when

$$\boxed{\frac{d \log \Delta D(s)}{ds} > \frac{d \log \Delta H(s)}{ds}.} \tag{32}$$

An increasing ratio need not yet exceed one and need not imply that the absolute cost margin increases. For a discrete set of model sizes, this is a comparative-statics model rather than a differentiability claim about the measured checkpoints.

Under the fixed-trace model $\Delta D(s) = \Delta N \, c_D(s)$ with constant $\Delta N > 0$, the condition reduces to $c_D'(s)/c_D(s) > \Delta H'(s)/\Delta H(s)$. If additionally $\Delta H(s) = H > 0$ is constant, the exact per-evaluation break-even value is

$$c_D^* = \frac{H}{\Delta N}. \tag{33}$$

If these quantities are only approximately constant, this is an approximation. Changing the backbone can change residuals, allocations, and review cost, so those quantities must be measured rather than assumed invariant.

Measurement protocol. *Denoising NFE* counts backbone evaluations, including repeated regional calls and any replay. Its implementation must state how batched regional predictions and classifier-free guidance are counted; batched forward calls and logical per-condition evaluations are not interchangeable. Different resolutions or extra backward computation can also change cost without a proportional change in raw NFE.

Diffusion GPU Time (s) measures elapsed device execution for diffusion segments using CUDA events, with the completion event synchronized before reading the elapsed time. Overlapping streams must be joined or timed over their enclosing diffusion interval rather than added as independent elapsed times. Controller-only gaps between diffusion segments are not labeled as diffusion execution. This is elapsed GPU time on the measured device, not aggregate GPU-seconds across devices.

End-to-End Wall Time (s) is measured from receipt of the prompt to delivery of the final image, including structuring, preview decoding, semantic-agent inference, scheduling, communication, synchronization, and diffusion. Synchronize pending device work at measurement boundaries; do not sum overlapping component timers to estimate this latency. Additional synchronization solely for fine-grained logging should not change the execution policy being benchmarked.

Controlled timing comparisons use the same hardware, precision, resolution, batch size, sampler, and software configuration. Warm-up is excluded and repeated runs are aggregated. Gate/dispatch cost and state-dependent review costs are logged separately to evaluate Eq. 29. Reviewer tokens remain a workload diagnostic, not a hardware-independent latency proxy. Calibration rollouts are offline costs and must be disclosed separately from online per-image inference; they are not silently included in or excluded from only one baseline's runtime.

D Additional Experiments

D.1 Additional Qualitative Analysis

Aggregate scores do not reveal how regional scheduling affects individual compositional requirements. We therefore compare the frozen backbone, uniform regional control, and **DIFFSCHED** on relation-heavy prompts that combine object identity, attributes, and spatial or interaction constraints. These



Figure 5: Additional qualitative comparison. Uniform regional control improves prompt adherence over the backbone, while **DIFFSCHED** better preserves the requested relations and achieves the highest AM in both examples.

cases are aligned with the relation-oriented evaluation settings used in compositional benchmarks such as T2I-CompBench++ (Huang et al., 2025).

As shown in Fig. 5, uniform regional control generally improves compositional fidelity over the backbone, but applying regional computation uniformly does not fully resolve the remaining relation errors. For “a purple car in front of a bear,” AM increases from 33.33 for the backbone to 46.42 with uniform regional control and 50.72 with **DIFFSCHED**. For “a lion jumping over a sparkling dog,” the corresponding scores are 32.27, 38.65, and 48.44.

The examples illustrate the role of adaptive scheduling beyond simply adding more regional computation. Uniform regional control applies the same type of intervention regardless of how the intermediate composition evolves, whereas **DIFFSCHED** can stop resolved regional actions and retain or strengthen control for requirements that remain difficult. The resulting improvement is most visible in the spatial and interaction relations, while the requested objects and attributes are largely preserved. These examples complement the aggregate results by showing that the gain comes from adapting *where* and *when* regional computation is used, rather than from uniformly increasing regional denoising.



Figure 6: Qualitative comparison between RAS (Liu et al., 2026b) and **DIFFSCHED** on Stable Diffusion 3.5 Medium. **DIFFSCHED** better preserves the counting and relational requirements in the shown examples while using fewer denoiser evaluations. GM and AM denote the corresponding Soft-TIFA scores.

D.2 Qualitative Generalization on Stable Diffusion 3.5

We further examine whether requirement-level scheduling transfers to a different diffusion backbone using Stable Diffusion 3.5 Medium (SD3.5-M) (Stability AI, 2024; Esser et al., 2024). We compare **DIFFSCHED** with Region-Adaptive Sampling (RAS) (Liu et al., 2026b) under the same SD3.5 backbone and generation configuration. RAS is a training-free spatially adaptive sampling method that allocates computation according to model-side regional focus and reuses cached predictions outside the selected regions. In contrast, **DIFFSCHED** associates regional computation with explicit prompt requirements and revises the allocation using diffusion-side residuals and selective semantic feedback. The comparison therefore examines whether model-side spatial importance alone provides the same information as the status of explicit compositional requirements.

Fig. 6 shows two representative compositional prompts. For "two motorcycles and exactly two donuts in front of a large clock," RAS preserves the major object categories but is less accurate in the joint counting and spatial composition, obtaining 21.63 GM and 61.51 AM. **DIFFSCHED** more clearly preserves the requested counts together with the motorcycle–donut–clock configuration, increasing the scores to 51.80 GM and 80.70 AM. This case is informative because the difficult requirement is not simply associated with a visually salient region: it combines exact numeracy with a relation among several objects.

For "exactly three yellow birds perched separately on a vintage bicycle," both methods recover the requested three yellow birds. RAS reaches 90.81 GM and 92.66 AM, but the birds are primarily organized around a generic perch and the requested bicycle relation is less clearly realized. **DIFFSCHED** preserves the count while more directly grounding the birds on the bicycle, reaching 95.89 GM and 96.29 AM. Thus, even when object presence and numeracy are largely correct, requirement-level scheduling can still help preserve the relation that connects those objects to the rest of the scene.

The comparison also exposes the difference between the scheduling signals. RAS allocates computation according to model-side spatial focus and temporal persistence (Liu et al., 2026b). Such signals are effective for identifying regions that merit additional backbone computation, but spatial activity does not directly indicate whether a requested count, attribute, or relation has already been satisfied. **DIFFSCHED** instead assigns computation to semantic groups derived from the prompt and can reconsider those assignments from intermediate generation evidence. Consequently, a visually prominent region need not continue receiving extra regional computation after its requirement is sufficiently resolved, while a visually simple but unresolved relation can retain targeted intervention.

Together, these examples provide two complementary observations. First, the scheduling mechanism remains applicable when moving from the SDXL backbone used in the main experiments to the transformer-based SD3.5-M (MMDiT-X) backbone. Second, explicit requirement status provides information complementary to the model-side spatial focus used by RAS. These observations support the broader motivation for combining adaptive diffusion computation with requirement-level semantic feedback.

Method	GM↑	AM↑	2D↑	Non-sp.↑	Num.↑	Complex↑	NFE↓	Diff. Time (s)↓	Reviews↓	Tokens↓
Region-Size	9.153	51.830	0.311	**0.324**	**0.667**	0.321	64.44	10.54	0.00	0
Visual-Complexity	9.287	52.629	0.311	0.322	**0.700**	**0.331**	85.38	13.67	0.00	0
w/o Planning Agent	10.316	49.079	0.103	0.320	0.567	0.311	**23.16**	**3.73**	1.52	5219.9
w/o Review Agent	10.027	51.109	0.303	0.321	0.633	0.327	33.66	5.46	0.00	0
w/o Trigger	**10.782**	52.467	0.296	**0.321**	0.633	0.320	40.66	6.79	2.96	8914.4
DIFFSCHED	10.708	**53.106**	**0.310**	0.320	**0.667**	**0.326**	37.52	6.07	**1.40**	**4827.9**

Table 5: Controlled scheduling study. GenEval 2 uses the frozen 100-prompt subset; T2I-CompBench++ uses five official prompts per category and three paired seeds. Lower is better for NFE, diffusion time, reviews, and tokens.

D.3 Additional Experimental Evidence

We provide additional evidence for the scheduling results in the main paper. The appendix focuses on four points: controlled allocation rules, a larger cost audit, category-level compositional behavior, and Trigger sensitivity. We also include a qualitative SD3.5 comparison with RAS to check whether the same scheduling idea remains useful beyond the SDXL setting used for the main controlled experiments.

Protocol and cost accounting. The controlled GenEval 2 study uses a frozen 100-prompt subset with seed 42. All internal variants share the SDXL backbone, 20 global denoising steps, resolution, sampler, initial noise, prompt set, semantic supports, and logging code. The T2I-CompBench++ study uses five official prompts from each reported category and three paired seeds, giving 15 images per method and category. These results are subset measurements rather than the full benchmark.

We count denoiser-equivalent evaluations (NFE) across global denoising, regional denoising, and BoxDiff probes. Diffusion time is measured with synchronized CUDA events and excludes semantic-agent inference and VAE decoding. Reviews per image and reviewer tokens describe controller workload. End-to-end wall time includes planning, preview decoding, semantic review, scheduling, communication, and diffusion. Because networked semantic-model latency can vary, reviewer calls and tokens are the more stable controller-side measure; wall-clock comparisons are interpreted only within a common execution environment.

Controlled scheduling and component ablations. Table 5 keeps the generation setting fixed and changes only the scheduling or controller component. Relative to Region-Size, **DIFFSCHED** improves GM/AM by 1.55/1.28 points while reducing NFE from 64.44 to 37.52 (41.8%) and diffusion time from 10.54 to 6.07 s (42.4%). Relative to Visual-Complexity, it improves GM/AM by 1.42/0.48 points while reducing NFE by 56.1% and diffusion time by 55.6%. The w/o Trigger variant obtains a slightly higher GM, but its AM is lower and it uses more diffusion computation and semantic review. The full method gives the highest AM among the controlled scheduling variants while avoiding the largest controller cost.

The two heuristic schedulers do not use online semantic feedback. In an audit of 220 generation records, both produced zero review calls, zero preview decodes, and zero schedule patches, and used the same `per_box` regional executor. Their higher NFE therefore comes from the fixed regional allocation itself rather than hidden controller work. The ablations separate this effect from the semantic controller: removing planning reduces compute but also lowers AM substantially, whereas removing review saves controller cost but loses the benefit of intermediate visual diagnosis.

Category-level behavior. Table 6 reports the balanced T2I-CompBench++ subset for four internal arms. **DIFFSCHED** obtains the highest complex-composition score (0.326) while remaining close to the fixed regional baseline on 2D, non-spatial, and numeracy. The result is consistent with the role of the scheduler: its main advantage appears when several requirements must be kept active, stopped, or revisited jointly, rather than when a single primitive can be handled by a fixed intervention.

Trigger sensitivity. We scale the calibrated threshold as $\tau_\beta = \beta\tau_0$. Across the 50-prompt sensitivity run in Table 7, AM varies by less than 0.09 points over an eight-fold threshold range and NFE varies by only 0.34 evaluations. Moving from $\beta = 0.25$ to the default $\beta = 1$ reduces reviews from 2.50 to 2.06 and reviewer tokens from 5505.7 to 4682.1, with essentially unchanged AM (52.364 vs. 52.358). The small non-monotonicity in realized review counts is expected for fresh closed-loop

Method	2D	Non-sp.	Num.	Complex
Dense Backbone	0.105	0.320	0.633	0.313
Uniform Regional	**0.311**	**0.323**	**0.700**	0.317
w/o Trigger	0.310	0.320	0.667	0.319
DIFFSCHED	0.310	0.320	0.667	**0.326**

Table 6: T2I-CompBench++ balanced official-prompt subset: five prompts per category and three paired seeds (15 images per method/category). Uniform Regional is an MCCD-style fixed-regional proxy, not a fresh run of the external MCCD implementation.

runs: changing an earlier decision can change the later states and scores. The monotonic statement in Sec. 3.4 applies to a common replayed score sequence. Values above one are empirical operating points outside the default conservative region.

β	GM	AM	NFE	Reviews	Tokens
0.25	**8.878**	52.364	39.06	2.50	5505.7
0.50	8.699	52.358	38.78	2.30	4902.8
0.75	8.699	52.358	38.78	**2.04**	**4613.0**
1.00	8.699	52.358	38.78	2.06	4782.1
1.25	8.726	**52.430**	38.78	**2.04**	4620.0
1.50	8.711	52.401	38.88	2.06	4781.3
2.00	8.694	52.346	**38.72**	2.06	4780.0

Table 7: Trigger sensitivity on 50 prompts. $\beta \leq 1$ is the conservative regime; $\beta > 1$ is evaluated empirically.

E PSEUDOCODE FOR DIFFSCHED

Algorithm 1 summarizes the complete inference procedure. Stage I constructs explicit requirements, semantic groups, spatial supports, and an initial regional schedule. Stage II keeps the global diffusion branch active while executing only the regional actions selected for the current interval. At each checkpoint, the regional–global residual provides a cheap one-sided execution signal. A sufficiently low score terminates the current regional interventions and bypasses semantic review; otherwise, or at a mandatory checkpoint, the Review Agent diagnoses the intermediate result and the Scheduler updates only the next execution interval. The resulting receding-horizon loop adapts both regional computation and semantic observation to the evolving generation state.

Algorithm 1 DIFFSCHED: Closed-Loop Semantic Computation Scheduling

Require: Prompt p; frozen denoiser ϵ_θ; VAE decoder $\mathcal{D}$; checkpoints $\mathcal{C} = \{c_0, \dots, c_Q\}$; allocation set $\mathcal{K}$; calibrated threshold $\tau_\beta = \beta\tau_0$
Ensure: Generated image x

Stage I: Semantic structuring and initial scheduling
$(\mathcal{R}, \mathcal{U}, \mathcal{M}, \mathcal{H}) \leftarrow \textsc{SemanticAgents}(p)$ ▷ requirements, groups, masks, risk/observability
$\mathcal{S} \leftarrow (\mathcal{R}, \mathcal{U}, \mathcal{M}, \mathcal{H})$
$\Pi_0 \leftarrow \textsc{CoordinatorPlan}(\mathcal{S}, \mathcal{K})$ ▷ initial regional actions
Sample initial latent z_0

Stage II: Closed-loop compositional generation
for $q = 0, \dots, Q-1$ **do**
  $\mathcal{I}_q \leftarrow [c_q, c_{q+1})$
  for $n \in \mathcal{I}_q$ **do**
    $\epsilon_n^g \leftarrow \epsilon_\theta(z_n, t_n, c_p)$ ▷ global prediction
    $\mathcal{A}_n \leftarrow \textsc{ActiveGroups}(\Pi_q, n)$
    for all $u_m \in \mathcal{A}_n$ **do**
      $\epsilon_n^m \leftarrow \epsilon_\theta(z_n, t_n, c_m)$
      $r_{m,n} \leftarrow \dfrac{\|M_m \odot (\epsilon_n^m - \epsilon_n^g)\|_2}{\|M_m \odot \epsilon_n^g\|_2 + \varepsilon}$
    end for
    $\widetilde{\epsilon}_n \leftarrow \textsc{MaskedFusion}(\epsilon_n^g, \{\epsilon_n^m, M_m, \lambda_m(n)\}_{m \in \mathcal{A}_n})$
    $z_{n+1} \leftarrow \textsc{SamplerStep}(z_n, t_n, \widetilde{\epsilon}_n)$
  end for
  if $q = Q-1$ **then**
    break
  end if

  // Checkpoint decision at c_{q+1}
  $\mathcal{A} \leftarrow \textsc{ReachingGroups}(\Pi_q, c_{q+1})$
  if $\mathcal{A} = \varnothing$ **then**
    $\Pi_{q+1} \leftarrow \textsc{AdvancePlan}(\mathcal{S}, \Pi_q)$
    continue
  end if
  for all $u_m \in \mathcal{A}$ **do**
    $(\bar{r}_{m,q+1}, \delta r_{m,q+1}) \leftarrow \textsc{ResidualSummary}(\{r_{m,n}\})$
    $s_{m,q+1} \leftarrow \bar{r}_{m,q+1} + \gamma[\delta r_{m,q+1}]_+$
  end for
  $s_{q+1} \leftarrow \max_{m \in \mathcal{A}} s_{m,q+1}$
  if $\textsc{MandatoryReview}(\mathcal{S}, c_{q+1})$ **then**
    REVIEW $\leftarrow$ TRUE
  else if $s_{q+1} \le \tau_\beta$ **then**
    $\Pi_{q+1} \leftarrow \textsc{StopResolved}(\Pi_q, \mathcal{A})$ ▷ stop continuation; skip review
    REVIEW $\leftarrow$ FALSE
  else
    REVIEW $\leftarrow$ TRUE
  end if
  if REVIEW **then**
    $\widehat{z}_0^{(q+1)} \leftarrow \textsc{CleanEstimate}(z_{c_{q+1}})$
    $\widehat{x}_{q+1} \leftarrow \mathcal{D}(\widehat{z}_0^{(q+1)})$
    $Y_{q+1} \leftarrow \textsc{ReviewAgent}(\widehat{x}_{q+1}, \mathcal{R}, \mathcal{U})$ ▷ PASS/UNCERTAIN/FAIL
    $\Pi_{q+1} \leftarrow \textsc{Schedule}(\mathcal{S}, \Pi_q, Y_{q+1}, \{s_{m,q+1}\}_{m \in \mathcal{A}})$ ▷ stop / continue / strengthen / change
    if $\textsc{RollbackRequired}(\Pi_{q+1})$ **then**
      $z_{c_{q+1}} \leftarrow \textsc{RestoreCachedLatent}(\Pi_{q+1})$
    end if
  end if
end for
$x \leftarrow \mathcal{D}(z_T)$
return x