RELEASE, DON'T REDRAW: INSTRUCTION-CONDITIONED RELEASE CONTROL FOR LONG-HORIZON MULTI-TURN IMAGE EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world image editing is often multi-turn, yet recursively applying an editor can resynthesize unrelated content, causing identity drift, overwritten edits, and lost objects. We introduce **ReleaseEdit**, a training-free framework that inherits the accepted image and releases only content causally implicated by the current instruction. An instruction-conditioned contract separates proposal formation from acceptance. It uses evidence- and carrier-aware localized release for subject editing, prospective creation evidence with audited generation-as-commit for object addition, and boundary-aware entity transport for background replacement. Under localized acceptance, unreleased content is directly inherited rather than indiscriminately regenerated, and the accepted output becomes the sole persistent visual state for the next turn. On 400 six-turn sequences (2,400 edits), ReleaseEdit outperforms five baselines in average CLIP-T (0.3318), CLIP-Dir (0.1173), CLIP-I (0.9142), and DINO-I (0.8049), and remains best at Turn 6. It ranks first in 73.75% of 640 blinded comparisons, versus 18.91% for the strongest baseline. Ten-turn comparisons further illustrate reduced qualitative drift while preserving current-turn editability.

1 INTRODUCTION

Image editing is often a sequence of decisions. A user may replace a scene, modify a subject's appearance, and add objects, expecting each accepted change to survive subsequent requests. Instruction-based generative models have made such interactions increasingly accessible (Brooks et al., 2023; Black Forest Labs et al., 2025). However, completing the latest request is only part of the task. An editor must also retain subject details and earlier modifications that remain relevant. Figure 1 compares this recursive process across methods: after background replacement, two subject edits, and two object additions, baseline sequences exhibit different forms of composition, identity, and appearance drift. Figure 2 isolates the same failure mode over a longer sequence, where changes accumulate in identity-critical regions that none of the instructions asks to modify even though individual outputs remain plausible. The content to preserve is therefore defined by the current image and instruction, and evolves throughout the interaction.

Recent methods improve successive edits through trajectory stabilization, spatial attention control, high-frequency preservation, and learned temporal consistency (Zhou et al., 2025; Liao et al., 2025; Xu et al., 2026; Huang et al., 2026). A central difficulty nevertheless remains in allocating preservation and editing freedom. Strong source constraints can impede the requested transformation, whereas insufficient constraints permit unrelated details to change. This tension depends on the operation. A local attribute edit affects a limited part of an existing subject; object creation requires space for content absent from the input; background replacement changes much of the image while retaining foreground entities. Applying the same form of protection to all three operations need not provide an appropriate balance.

Spatial masks provide an important starting point, but the spatial extent of an edit does not fully specify how source information should enter the generation process. Local diffusion editing already combines masked synthesis with source preservation (Avrahami et al., 2022; 2023), and feature-based methods control source information inside the generator (Avrahami et al., 2025; Wei et al., 2025;

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.

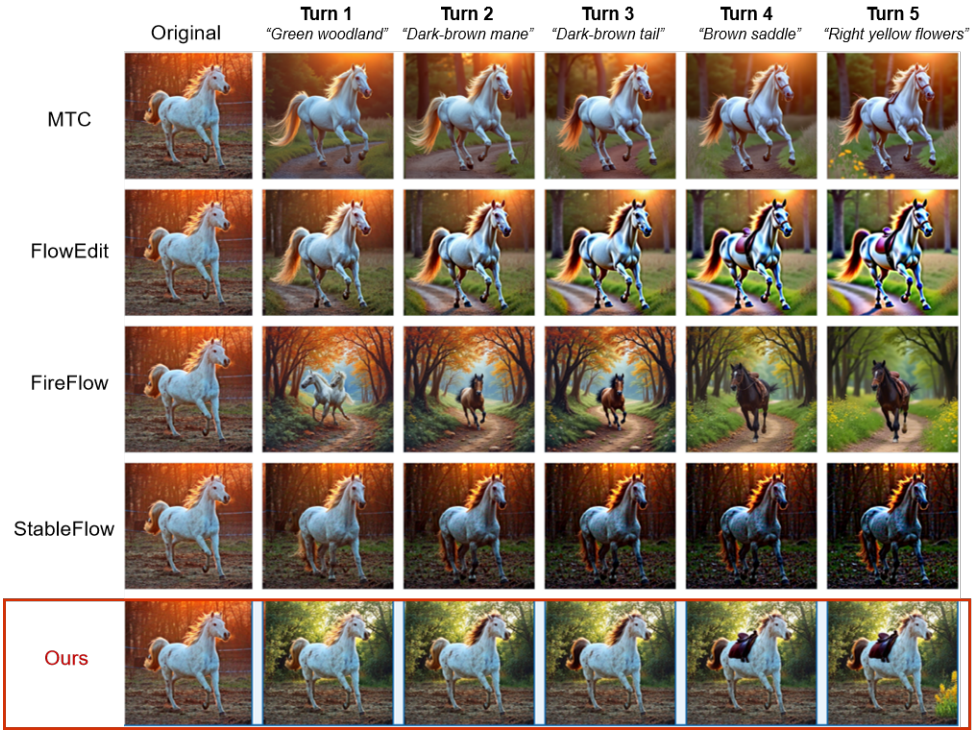

Figure 1: **Qualitative comparison on recursive editing.** All methods receive the same input and instruction sequence, with each output becoming the next input. Existing editors accumulate composition, identity, or appearance drift, whereas ReleaseEdit better retains the horse's pose, identity, and accepted edits while completing background, attribute, and object changes.

Ouyang et al., 2025). This distinction is particularly consequential for large scene changes and at foreground boundaries. A transformer token may contain both an entity and its old background. Strong reference transfer can preserve unwanted background content, while weak transfer can remove small objects or thin structures. The question is therefore both where to permit change and how to retain the reference information needed by the current operation.

We introduce **ReleaseEdit**, a training-free framework that combines *selective spatial release* with *reference transport* for heterogeneous multi-turn editing (Figure 3). Its central design is to match the preservation mechanism to the requested change. Subject edits use separate proposal and commit fields; object addition uses prospective creation evidence with an audited generation-as-commit policy; and background replacement permits full-canvas latent evolution while transporting foreground representations from a source-conditioned reference. These mechanisms share the same sequential contract: each output becomes the next input, and preservation is specified against that evolving state. The framework thus accommodates different editing operations without requiring an identical sampler or identical spatial constraint for every operation.

For local subject editing, ReleaseEdit combines early target-token responses with a grounded subject region to construct the release support. For object addition, it uses the target-token response to estimate a prospective creation region, since the requested object cannot yet be segmented in the input. For background replacement, it grounds and segments the entities to retain, then conditions representation transfer on their size and occupancy within image tokens. The boundary-aware entity transport modulates source value features and attention outputs separately: it reduces transfer in weakly occupied boundary tokens and provides additional attention-output support to the cores of small entities. This is designed to reconcile foreground preservation with changes in the surrounding scene.

Our contributions are threefold:

- We present a task-dependent preservation framework for mixed multi-turn editing, with an evolving ledger and an explicit separation between proposal formation and acceptance.
- We develop dual-field subject editing and prospective creation evidence for object addition, allowing each branch to use the spatial support appropriate to its operation.
- We develop boundary-aware, continuous-scale entity transport that separately regulates source value features and attention outputs during scene replacement.

2 Related Work

Diffusion and flow-based editing. InstructPix2Pix learns instruction-conditioned image transformations (Brooks et al., 2023), while FLUX.1 Kontext supports generation and editing through in-context conditioning (Black Forest Labs et al., 2025). Training-free methods modify the inference process of pretrained generators. RF-Inversion derives dynamic control for inversion and editing (Rout et al., 2025); FireFlow improves the numerical efficiency of rectified-flow inversion (Deng et al., 2025); and FlowEdit constructs a direct source-to-target trajectory without explicit inversion (Kulikov et al., 2025). These works establish different ways to reconcile source fidelity with target-directed generation. ReleaseEdit studies task-dependent preservation in a sequential setting through paired reference trajectories and entity-conditioned representation transfer.

Attention control and regional preservation. Prompt-to-Prompt manipulates cross-attention to connect textual changes with image structure (Hertz et al., 2023). Stable Flow identifies vital layers for reference-feature injection (Avrahami et al., 2025), and FreeFlux analyzes layer-specific roles to design editing interventions, including region-preserving value injection for background replacement (Wei et al., 2025). ProEdit mitigates excessive source influence through regional feature mixing and latent perturbation (Ouyang et al., 2025). Mask-based methods such as Blended Diffusion and Blended Latent Diffusion preserve surrounding content through spatial blending (Avrahami et al., 2022; 2023); segmentation models such as SAM 2 provide region estimates for downstream control (Ravi et al., 2024). ReleaseEdit's background branch combines full-canvas latent updates with soft entity-conditioned representation transfer so that the proposal itself retains foreground structure. The transported proposal is accepted directly and audited for foreground preservation, without post-hoc pixel compositing in the reported background runs. Its transport separates boundary and core occupancy and varies attention-output support continuously with entity size.

Multi-turn image editing. MTC combines flow inversion, dual-objective LQR guidance, and adaptive attention to mitigate accumulated error (Zhou et al., 2025). FreqEdit preserves high-frequency information through reference-velocity injection, spatial modulation, and path compensation (Liao et al., 2025). Learning-based approaches introduce additional sequential structure: AnchorEdit uses an autoregressive diffusion framework with causal memory (Xu et al., 2026), and MT-EditFlow optimizes editing trajectories using reinforcement learning (Huang et al., 2026). IMAGAgent coordinates editing through constraint-aware planning and reflection (Shen et al., 2026). RegionCache uses semantic correspondence and attention localization to reuse unchanged diffusion states across turns (Li et al., 2026). ReleaseEdit focuses on the preservation operator within each transition: subject changes use localized release, addition uses audited generation-as-commit, and scene replacement uses entity transport. Its sequential interface reuses the preceding output, without introducing a learned temporal memory. Whether this control improves current-edit completion and retention of valid earlier edits must be assessed jointly, rather than inferred from image similarity alone.

3 Method

3.1 Problem formulation

We consider a sequence of editing turns indexed by k. At turn k, the editor receives the accepted image I_{k-1}, its source description c_{k-1}, and a target description c_k derived from the new instruction. The requested operation belongs to one of three edit families, $T_k \in \{\textsc{Subject}, \textsc{Addition}, \textsc{Background}\}$. The output I_k becomes the reference image for turn $k+1$. A successful turn must therefore satisfy the current request while retaining both source content and earlier edits that have not been superseded.

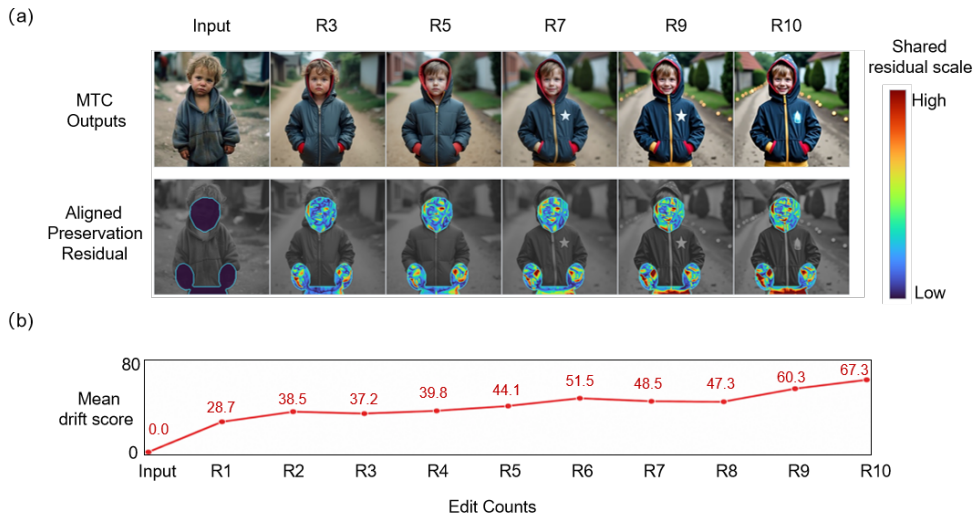

Figure 2: **Protected content drifts across editing turns.** MTC outputs (top), aligned Lab residuals in never-targeted identity regions (middle), and their mean residual (bottom) show accumulating drift over ten edits, motivating explicit preservation of the accepted state.

ReleaseEdit separates two decisions that are often conflated: (i) the spatial and semantic support over which a proposal should be formed, and (ii) the permission with which that proposal is committed to the accepted image. The first decision is task dependent because changing an existing attribute, creating absent content, and replacing a scene require different generative dynamics. The second is represented by a common release contract whenever a localized commit is appropriate. This distinction allows proposal formation to use a broad semantic context without silently broadening the final edited region.

Figure 3 summarizes this shared lifecycle and the three task-conditioned proposal paths. The complete equation-centric procedure is provided in Appendix B (Algorithm 1).

3.2 Algorithm Overview

Algorithm 1 in Appendix B factorizes one turn into a task-conditioned proposal $Q_k^{T_k}$ and an explicit acceptance policy. Subject edits use the localized release operator Π_k, whereas addition and background directly accept coherent proposals and audit the resulting entity changes. The proposal geometries remain specialized, while the evolving state and audit record provide a common contract across turns.

3.3 Scale-Normalized Release Roles

Let $E_k \in [0,1]^{H\times W}$ denote automatically observed edit evidence. We convert E_k into continuous Preserve, Boundary, and Edit roles. Define $\Omega_k = \{x : E_k(x) > 0\}$; empty or full-image support is treated as an observation failure. From Ω_k, we compute a signed Euclidean distance $d_k(x)$, positive inside the support and negative outside. Its scale is normalized by a robust quantile of the interior distance,

$$\bar{d}_k(x) = \frac{d_k(x)}{Q_{0.9}(\{d_k(x) : d_k(x) > 0\}) + \varepsilon}. \tag{1}$$

With a dimensionless boundary temperature τ_b, we define

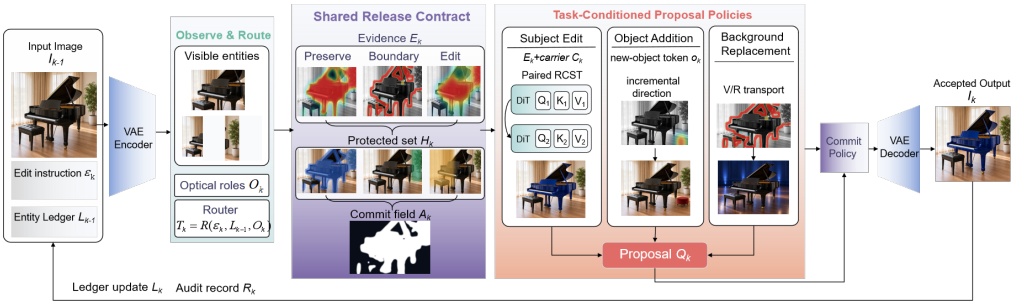

Figure 3: **ReleaseEdit framework.** The observer routes each instruction and builds evidence E_k, protected set H_k, and commit field A_k. A task-specific policy produces proposal Q_k for subject editing, object addition, or background replacement; the commit policy accepts I_k and updates the ledger for the next turn.

$$\begin{aligned} b_k(x) &= \exp\big(-|\bar{d}_k(x)|/\tau_b\big)\,, \quad & s_k(x) &= \sigma\big(\bar{d}_k(x)/\tau_b\big)\,, \\ u_k(x) &= s_k(x)\,[1-b_k(x)], & p_k(x) &= [1-s_k(x)]\,[1-b_k(x)]. \end{aligned} \tag{2}$$

The three fields are normalized to sum to one at each pixel. The commit field is

$$A_k(x) = E_k(x)\,[u_k(x) + b_k(x)]. \tag{3}$$

Thus, semantic evidence supplies permission, while the role construction controls confidence near the observed boundary. The construction contains no fixed pixel-width feather and transfers across resolutions and object scales.

For localized subject proposals, the release projection is

$$I_k = I_{k-1} + A_k \odot (Q_k - I_{k-1}), \quad I_k(x) = I_{k-1}(x), \quad H_k(x) = 1, \tag{4}$$

where $H_k \in \{0,1\}^{H\times W}$ is the binary union of the zero-release complement and any explicitly protected entities. Equation (4) is the closed-form solution of a per-pixel quadratic projection between the proposal and the accepted image. The projection is elementary and is not claimed as a standalone contribution; its role is to provide an auditable localized acceptance rule when that policy is used.

3.4 Dual-field subject editing

For a subject edit, the observed attribute evidence E_k may cover only part of the entity over which a coherent appearance change must be formed. Let $C_k \in [0,1]^{H\times W}$ denote the semantic carrier of the attribute. We retain E_k as the final commit permission but construct a separate proposal field

$$G_k = \mathrm{clip}(E_k + \alpha_{\mathrm{prop}}[C_k - E_k]_+, 0, 1)\,, \qquad 0 \le \alpha_{\mathrm{prop}} \le 1, \tag{5}$$

where α_{prop} is a task-level proposal reach. We use $\alpha_{\mathrm{prop}} = 1$ by default and keep it fixed across examples. When $E_k = C_k$, as in a whole-subject appearance edit, Equation (5) reduces to the original full-carrier proposal. When E_k is local, the proposal may use the surrounding subject context, but Equation (4) still prevents G_k from expanding the final commit support.

We measure the carrier coverage from the observed evidence before proposal propagation,

$$\chi_k = \mathrm{clip}\left(\frac{\sum_x \min\{E_k(x), C_k(x)\}}{\sum_x C_k(x) + \varepsilon}, 0, 1\right), \qquad \rho_k = 1 - \chi_k, \tag{6}$$

where an empty carrier is a localization failure. The clipped, unpropagated overlap keeps $\rho_k \geq 0$ and prevents $\alpha_{\text{prop}} = 1$ from forcing $\chi_k = 1$.

We instantiate the subject proposal with a paired, role-conditioned reference trajectory. Source and target tokens are aligned automatically, and the non-shared token contribution augments the native source–target attention direction after median-norm normalization. Preserve, Boundary, and Edit queries derived from G_k then receive distinct transition budgets across layers and denoising time.

Let $z_{\text{ref}}(t) = (1-t)z + t\xi$ be the inversion reference from source latent z to noise ξ, v_s and v_t the paired source and target velocity predictions, and c_{align} an alignment correction. ReleaseEdit exposes the inherited coverage-adaptive proposal through the offset form

$$\begin{aligned} a_k(t) &= \chi_k \left[v_s(t) + c_{\text{align}}(t)\right] + \rho_k \, \dot{z}_{\text{ref}}(t), \\ \dot{h}_k(t) &= v_t(z_{\text{ref}}(t) + h_k(t)) - a_k(t), \qquad\qquad z_k(t) = z_{\text{ref}}(t) + h_k(t). \end{aligned} \tag{7}$$

The offset is integrated through the proposal roles obtained from G_k. After decoding, the raw proposal is passed directly to Equation (4). We deliberately do not smooth the proposal by default: fixed attenuation of high-frequency innovation removes valid texture in fur, hair, fabric, and foliage. A scale-normalized coherent delta regularizer remains an explicit ablation for proposals containing isolated artifacts.

3.5 Target-trajectory object addition

An added object is absent from I_{k-1} and therefore cannot be localized by segmenting the input alone. ReleaseEdit first evaluates the target generation trajectory and aggregates attention associated with the concise new-object phrase, yielding prospective creation evidence E_k^{add}. Spatial relations remain in the complete target prompt but are not averaged into the object-token observation. This prevents carrier and relation words from diluting the creation signal.

The default addition proposal follows a target-trajectory incremental policy with direction gain α_{add} inside E_k^{add}, while the source reference anchors its complement. Unlike an appearance edit, object creation jointly determines silhouette, occlusion, contact shadows, and nearby appearance. Projecting this coupled change through a narrow post-hoc mask can fragment an otherwise coherent object. We therefore use generation-as-commit for addition by default,

$$I_k = Q_k^{\text{add}}, \tag{8}$$

and audit changes to previously registered entities. Two constrained modes are retained for controlled experiments: projection through the target- trajectory semantic field and projection through a post-proposal grounded instance. These modes trade proposal coherence for exact outside-region preservation and are reported separately rather than silently selected after visual inspection.

3.6 Background replacement with optical entity roles

Background replacement releases most of the canvas while preserving foreground entities. The observer maintains separate masks M_k^{opq} for opaque entities and, when present, M_k^{tr} for transmissive entities. The proposal follows a paired inversion trajectory with full-canvas scene evolution. Inside the transformer, optical-role transport interpolates source and target value features and attention outputs according to entity occupancy and boundary confidence. Its inward preservation field uses the dimensionless relative width ω, while transmissive entities retain a weak target-scene anchor,

$$V_t' = w_V \odot V_s + (1 - w_V) \odot V_t, \quad R_t' = w_R \odot R_s + (1 - w_R) \odot R_t. \tag{9}$$

Opaque cores receive strong reference support, mixed boundary tokens receive weaker support to avoid transporting the old background, and transmissive regions retain partial target-scene influence. The decoded proposal Q_k^{B} is accepted directly,

$$I_k = Q_k^{\text{B}}. \tag{10}$$

The foreground masks are used to audit entity changes rather than to form a post-hoc pixel composite. This matches the reported implementation: transport must preserve a coherent foreground inside the proposal itself. Figure 6 therefore isolates the effect of transport on the same proposal-direct background path used in the benchmark.

3.7 Sequential state and auditable policy choices

ReleaseEdit has no learned temporal memory: each turn records the accepted output, observations, release evidence, proposal, policy metadata, and audit flags, and the next turn starts from I_k. Frozen policies do not depend on object identity, requested color, or visual seed selection; the audit flags failures but neither modifies the image nor decides semantic success.

The branches use distinct samplers and acceptance policies but share explicit proposal–acceptance separation, the evolving reference, and an entity audit. Their paired-RCST, incremental target-trajectory, and optical-role proposals retain provenance. Our claim concerns the dual-field interface and sequential contract, not the underlying solvers; unlike FlowEdit, ReleaseEdit uses inversion/reference paths rather than an inversion-free direct source-to-target ODE.

4 Experiments

4.1 Evaluation protocol

We evaluate four questions: whether one frozen task-conditioned policy suite handles subject editing, object addition, and background replacement on unseen images; whether it preserves non-target content and still-active earlier edits; which proposal and acceptance components account for this behavior; and how performance changes across recursive turns. Failed edits remain in the evaluation, and no seed or candidate is selected after viewing test outputs.

Tasks and sequences. The held-out benchmark covers subject, addition, and background task families. Subject edits cover color, material, and local appearance over multiple classes. Additions include isolated and attached objects, counts, contact, partial occlusion, and spatial relations. Background edits stress thin or small foreground entities and transmissive materials. Multi-turn sequences mix all three families; every method recursively edits its own preceding output. An earlier requirement remains active unless a later instruction supersedes it, and the complete sequence is the unit of aggregation.

Frozen execution and baselines. Routing, prompts, observation models and thresholds, numerical parameters, random seeds, and candidate budgets are frozen before test inspection; observations are then computed automatically from each input. ReleaseEdit uses one candidate per instruction; a grounding, generation, or validation failure is counted as failure. We compare RF-Inversion (Rout et al., 2025), FireFlow (Deng et al., 2025), FlowEdit (Kulikov et al., 2025), Stable Flow (Avrahami et al., 2025), and MTC (Zhou et al., 2025) using their official interfaces at matched resolution.

4.2 Metrics

We report cumulative-prompt CLIP-T, original-source CLIP-I and DINO-I, per-turn CLIP-Dir, Laplacian variance, Edit SR (successful edits divided by 2,400 attempted turns), and blinded first-place preference rate. The same units are used for every method; Laplacian variance is only a sharpness diagnostic. The frozen success rubric is specified in Appendix A.

Sequential and human evaluation. Following the user-study design of FreeFlux (Wei et al., 2025), we measure marginal edits with per-turn CLIP-Dir and collect 640 blinded six-way Turn-6 rankings from 32 participants; Appendix A gives the complete protocol.

Automatic metrics. Table 1 reports all methods on 400 six-turn sequences. Across turns, ReleaseEdit obtains the highest average CLIP-T, CLIP-Dir, CLIP-I, and DINO-I, while FireFlow leads Laplacian sharpness. At Turn 6, ReleaseEdit again leads all four semantic and preservation metrics, while FireFlow retains the highest Laplacian score; ReleaseEdit also has the highest Edit SR (74.50%),

Table 1: **Six-turn quantitative comparison.** Avg. aggregates 2,400 turns; T6 reports the final turn and blinded first-place preference rate (PR). Edit SR is successful edits/2,400 turns and stands alone. Best and second-best are **bold** and underlined.

	Avg. Turns 1–6					Turn 6						Edit SR
Methods	CLIP_T	CLIP_{dir}	CLIP_I	DINO_I	Lap.	CLIP_T	CLIP_{dir}	CLIP_I	DINO_I	Lap.	PR	(%) ↑
FlowEdit	0.3222	0.0579	0.8596	0.6725	1070.2	0.3236	0.0449	0.8209	0.6063	1075.4	4.06	47.75
StableFlow	0.3130	0.0321	0.8915	0.8003	860.0	0.3152	0.0194	0.8312	0.7099	810.6	2.19	25.21
MTC	0.3161	0.0520	0.8579	0.6452	995.6	0.3205	0.0565	0.8395	0.6125	885.9	18.91	52.25
FireFlow	0.3189	0.0501	0.8730	0.6920	**1229.6**	0.3255	0.0457	0.8445	0.6395	**1274.9**	0.63	35.13
RF-Inversion	0.3190	0.0455	0.8364	0.6094	293.8	0.3178	0.0472	0.8115	0.5706	108.6	0.47	9.88
ReleaseEdit	**0.3318**	**0.1173**	**0.9142**	**0.8049**	1091.7	**0.3324**	**0.1315**	**0.9063**	**0.7341**	1083.6	**73.75**	**74.50**

versus 52.25% for MTC. These quantities describe different trade-offs rather than a single quality ordering: source similarity can penalize intended creation, whereas CLIP alignment does not verify local retention. CLIP-T compares I_k with cumulative c_k; CLIP-I and DINO-I compare I_k with the original source I_0. Because prompts and task mixtures vary, the two CLIP-T columns support within-column method comparison, not monotonic drift analysis. ReleaseEdit ranks first in 472 of 640 trials (73.75%), followed by MTC with 121 (18.91%); FlowEdit, StableFlow, FireFlow, and RF-Inversion receive 26, 14, 4, and 3.

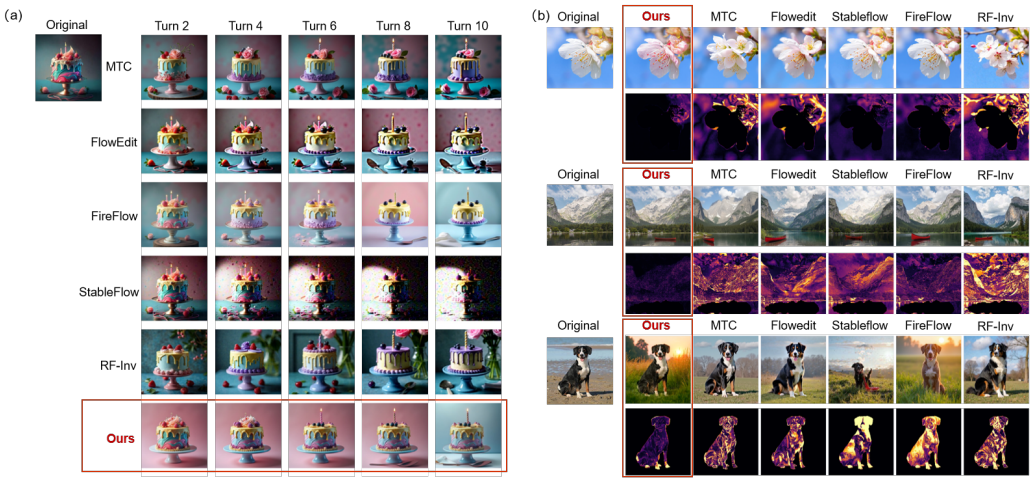

Figure 4: **Qualitative comparison.** *Left:* ten-turn cake editing. *Right:* subject editing, object addition, and background replacement with non-edit-region difference maps. Red boxes mark ReleaseEdit, which better preserves established content.

Turn-wise baseline behavior. Appendix Figure A.1 separates marginal edit alignment from source preservation over 400 sequences. ReleaseEdit maintains the strongest edit-direction alignment while maintaining the highest DINO-I at every turn.

4.3 Ablation Experiments



Figure 5: **Subject guidance.** Recoloring with $g \in \{2.5, 3.5, 4.5, 5.5\}$; the red box marks $g = 3.5$.

Table 2: **Dataset-level component ablation on 400 six-turn sequences.** Avg. aggregates all 2,400 turns, and T6 reports the final recursive turn. All columns are higher-is-better. Best and second-best values are **bold** and underlined, respectively.

	Avg. Turns 1–6				Turn 6			
Variant	CLIP_T	CLIP_{dir}	CLIP_I	DINO_I	CLIP_T	CLIP_{dir}	CLIP_I	DINO_I
ReleaseEdit	**0.3318**	**0.1173**	**0.9142**	**0.8049**	**0.3324**	**0.1315**	**0.9063**	**0.7341**
w/o localized release	0.2047	0.0412	0.8624	0.5946	0.2129	0.0483	0.7982	0.5969
w/o direct addition commit	0.2236	0.0689	0.6275	0.6817	0.2314	0.0617	0.6548	0.6882
w/o boundary/scale transport	<u>0.3291</u>	<u>0.1048</u>	<u>0.9086</u>	<u>0.7983</u>	<u>0.3288</u>	<u>0.1176</u>	<u>0.9011</u>	<u>0.7285</u>

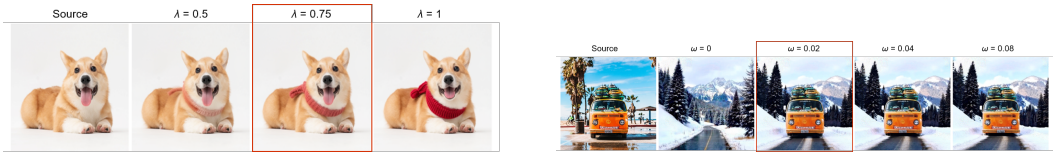

Figure 6: **Addition and background controls.** *Left:* scarf addition with $\alpha_{\text{add}} \in \{0.508, 0.75, 1.0\}$. *Right:* proposal-direct background outputs $Q_k^{\text{B}} = I_k$ with no transport versus $\omega \in \{0.02, 0.04, 0.08\}$. Red boxes mark the fixed-case operating points; the frozen benchmark also uses $\omega = 0.02$.

Component ablation. Table 2 evaluates each removal on the same 400 six-turn sequences as the main comparison. The complete method is best in all eight columns. Removing localized release causes the largest structural decline in DINO-I: the average score drops from 0.8049 to 0.5946, and the Turn-6 score from 0.7341 to 0.5969. Removing the direct addition commit instead produces the largest CLIP-I loss, from 0.9142 to 0.6275 on average and from 0.9063 to 0.6548 at Turn 6. These distinct failure signatures match the intended roles of the two branches: localized release protects persistent structure, whereas direct addition commit prevents an accepted creation from being discarded or overwritten. Removing boundary/scale transport leaves the global scores closer to the full model but is consistently second-best rather than tied. Its largest gap is in CLIP-Dir (0.0125 on average and 0.0139 at Turn 6), while its CLIP-I and DINO-I gaps remain below 0.007. This pattern is consistent with transport being a targeted coherence control for background cases, rather than the main driver of dataset-wide alignment or preservation.

Parameter sensitivity. Figures 5 and 6 visualize fixed-case sweeps for subject guidance, addition gain, and background transport width; Appendix Table 3 gives the values and score definitions. Subject guidance exposes a completion–retention trade-off, whereas excessive addition gain reduces integrity for little extra completion. For background replacement, a narrow band retains proposal coherence and the fixed case favors $\omega = 0.02$, matching Appendix B. These diagnostic sweeps are not pooled with the 400-sequence test set.

5 Limitations and Broader Impact

The framework depends on automatic grounding and segmentation; errors in those observations can misallocate release, and branch-specific proposals retain distinct costs. Longer sequences and transmissive cases remain open tests, and deployments should retain provenance because editing can enable deception.

6 Conclusion

ReleaseEdit treats recursive editing as task-conditioned proposals governed by an evolving preservation contract, separating evidence for proposal formation from permission for acceptance. This supports localized subject changes, object creation, and foreground-aware scene replacement in one auditable framework. Across 400 six-turn sequences, it leads five baselines on average semantic-alignment and source-preservation metrics and receives 73.75% of first-place votes. Ten-turn comparisons further indicate that selective release reduces accumulated drift without sacrificing current-turn editability.

AI USE STATEMENT

In this work, generative AI tools were used to assist with literature organization, conceptual and mathematical formulation, implementation prototyping, experimental-design feedback, figure preparation, qualitative analysis, and drafting and reformatting the paper. Generative AI was not used to generate experimental results, fabricate citations, or make final scientific judgments. All AI-assisted material has been reviewed by the authors, and all algorithms, code, experimental runs, metrics, citations, and claims have been independently verified. The authors take full responsibility for the final content of this work, including any text, claims, code, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The method edits images and may be used to create misleading visual content. We recommend maintaining provenance and edit histories, labeling generated or edited media where appropriate, and avoiding deployment for impersonation, fraud, or other deceptive uses. No private personal images are used. The 32-participant preference study records only anonymous rankings and minimal quality-control information. The final submission will document recruitment, consent, compensation, exclusion criteria, and the applicable institutional review or exemption procedure.

REPRODUCIBILITY STATEMENT

Section 3 specifies the algorithmic components, and Section 4 records the frozen prompts, routing, seeds, sampling budgets, metrics, aggregation, and evaluation protocol. Appendices B and E additionally record the execution environment, background diagnostic, and ablation-score definitions. The anonymized reviewer package includes the task-conditioned control layer, exact evaluation-facing defaults, and invariant tests. The full experiment harness, data manifests, and generated artifacts will be released after anonymous review.

REFERENCES

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended Diffusion for Text-driven Editing of Natural Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. URL https://arxiv.org/abs/2111.14818.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended Latent Diffusion. *ACM Transactions on Graphics*, 42(4):1–11, 2023. doi: 10.1145/3592450. URL https://arxiv.org/abs/2206.02779.

Omri Avrahami, Or Patashnik, Ohad Fried, Egor Nemchinov, Kfir Aberman, Dani Lischinski, and Daniel Cohen-Or. Stable Flow: Vital Layers for Training-Free Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2411.14430.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space, 2025. URL https://arxiv.org/abs/2506.15742.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://arxiv.org/abs/2211.09800.

Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. FireFlow: Fast Inversion of Rectified Flow for Image Semantic Editing. In *Proceedings of the International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2412.07517.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2208.01626.

Jiahui Huang, Yasi Zhang, Tianyu Chen, Shu Wang, Jianwen Xie, Oscar Leong, Mingyuan Zhou, Nanzhu Wang, and Ying Nian Wu. MT-EditFlow: Reinforcement Learning for Multi-Turn Image Editing with Flow Matching, 2026. URL https://arxiv.org/abs/2606.01985.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2412.08629.

Peizheng Li, Xin Ai, Hanyuan Liu, Qiange Wang, and Yanfeng Zhang. RegionCache: Semantic-Aware Region Reuse for Efficient Multi-Turn Image Generation, 2026. URL https://arxiv.org/abs/2608.29809.

Yucheng Liao, Jiajun Liang, Kaiqian Cui, Baoquan Zhao, Haoran Xie, Wei Liu, Qing Li, and Xudong Mao. FreqEdit: Preserving High-Frequency Features for Robust Multi-Turn Image Editing, 2025. URL https://arxiv.org/abs/2512.01755.

Zhi Ouyang, Dian Zheng, Xiao-Ming Wu, Jian-Jian Jiang, Kun-Yu Lin, Jingke Meng, and Wei-Shi Zheng. ProEdit: Inversion-based Editing From Prompts Done Right, 2025. URL https://arxiv.org/abs/2512.22118.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment Anything in Images and Videos, 2024. URL https://arxiv.org/abs/2408.00714.

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic Image Inversion and Editing using Rectified Stochastic Differential Equations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10792.

Fei Shen, Chengyu Xie, Lihong Wang, Zhanyi Zhang, Xin Jiang, Xiaoyu Du, and Jinhui Tang. IMAGAgent: Orchestrating Multi-Turn Image Editing via Constraint-Aware Planning and Reflection, 2026. URL https://arxiv.org/abs/2603.29602.

Tianyi Wei, Yifan Zhou, Dongdong Chen, and Xingang Pan. FreeFlux: Understanding and Exploiting Layer-Specific Roles in RoPE-Based MMDiT for Versatile Image Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2503.16153.

Hang Xu, Xiaoxiao Ma, Guohui Zhang, Yu Hu, Siming Fu, Jie Huang, Lin Song, Haoyang Huang, Nan Duan, and Feng Zhao. AnchorEdit: Maintaining Temporal Consistency in Multi-turn Image Editing via Causal Memory, 2026. URL https://arxiv.org/abs/2606.11751.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn Consistent Image Editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2505.04320.

A TURN-WISE EVALUATION

Edit-success rubric. A turn is successful only when the current request and all still-active earlier requirements are satisfied, no material unintended change is judged, and the output passes the frozen visual-quality rubric. Grounding, generation, and validation failures, as well as uncertain judgments, remain in the denominator; no failed case is rerolled or discarded.

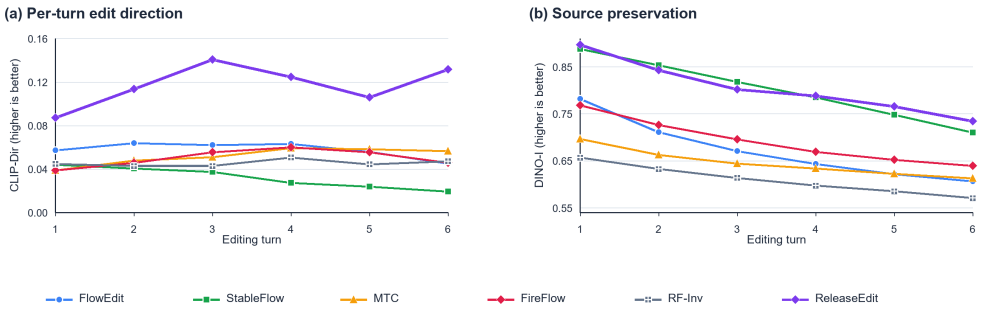

Figure A.1: **Behavior across recursive editing turns.** Mean per-turn CLIP direction alignment and DINOv2 source similarity over 400 six-turn sequences. CLIP-Dir compares $I_{k-1} \to I_k$ with $c_{k-1} \to c_k$; DINO-I measures similarity to the original source.

A.1 Per-turn CLIP direction

We extend the standard single-turn CLIP-Dir used by FreeFlux (Wei et al., 2025) to recursive editing. With normalized CLIP image and text embeddings E_I and E_T, we compute

$$\begin{aligned} d_k^I &= \mathrm{norm}(E_I(I_k) - E_I(I_{k-1}))\,, \quad d_k^T = \mathrm{norm}(E_T(c_k) - E_T(c_{k-1}))\,, \\ \mathrm{CLIPDir}_k &= \langle d_k^I, d_k^T \rangle. \end{aligned} \tag{11}$$

At $k = 1$, I_0 and T_0 denote the original image and source description, recovering the usual single-turn metric. For $k > 1$, each method's own preceding output is the image reference, while $c_{k-1} \to c_k$ isolates the new instruction from the cumulative prompt. The resulting values measure marginal per-turn alignment and are not directly comparable to single-turn scores from different datasets or task mixtures.

A.2 Blinded user-preference protocol

Following FreeFlux (Wei et al., 2025), we evaluate the final image after six recursive turns. Thirty-two participants each rank six anonymized Turn-6 outputs for 20 randomly assigned sequences, yielding 640 valid trials. Each trial shows the original image, the complete instruction history, and outputs from ReleaseEdit, FlowEdit, StableFlow, MTC, FireFlow, and RF-Inversion in independently randomized positions. Participants rank all methods without ties by jointly considering source identity and unedited-content preservation, and completion of the requested edit history. For rank $r_{n,m}$ assigned to method m in trial n, we report

$$\mathrm{PR}_m = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}[r_{n,m} = 1] \times 100\%. \tag{12}$$

B Complete ReleaseEdit algorithm

Algorithm 1 gives the complete equation-centric procedure for one recursive editing turn. It makes explicit the shared observation, routing, release, audit, and ledger-update stages, while retaining the three task-conditioned proposal policies described in Section 3.

Execution environment. All reported experiments were run on a server equipped with eight NVIDIA GeForce RTX 3090 GPUs. This describes the hardware available for our runs; the method does not require all eight GPUs to be occupied simultaneously.

Frozen configuration snapshot. We use FLUX.1-dev at 512×512, one candidate per request, seed 42, and guidance 3.5. Subject and background proposals use 25 sampling steps; addition uses its

verified 15-step schedule and incremental-direction gain $\alpha_{\text{add}} = 0.75$. Qwen2.5-VL-3B and SAM 2 provide the automatic grounding/segmentation observations. The background operator accepts the transported proposal directly and uses relative interior width $\omega = 0.02$ and transmissive anchor 0.25. Prompts, routing, observations, injection schedules, and branch parameters are frozen before test inspection; grounding, generation, or validation failures are retained as failures rather than rerolled.

Algorithm 1: ReleaseEdit as Task-Conditioned Release

Input: $\mathsf{S}_{k-1} = (I_{k-1}, c_{k-1}, \mathcal{L}_{k-1})$, target c_k, frozen editor Θ. **Output:** S_k, $\mathcal{R}_k$.

$\Delta c_k = \text{Diff}(c_{k-1}, c_k)$, $(T_k, H_k) = (\text{Route}, \text{Protect})(\Delta c_k, \mathcal{L}_{k-1})$.

$E_k = \text{Obs}_{T_k}(I_{k-1}, \Delta c_k, \mathcal{L}_{k-1})$, $r_k = (p_k, b_k, u_k) = \text{Role}(E_k)$, $A_k = E_k \odot (b_k + u_k)$.

// Subject proposal

$C_k = \mathcal{C}_\Theta(I_{k-1}, \Delta c_k)$, $G_k = \text{clip}(E_k + \alpha_{\text{prop}}[C_k - E_k]_+, 0, 1)$, $(\chi_k, \rho_k) = \left(\text{clip}\left(\frac{\langle \min(E_k, C_k), \mathbf{1}\rangle}{\langle C_k, \mathbf{1}\rangle + \varepsilon}, 0, 1\right), 1 - \chi_k\right)$.

$\dot{h}_k(t) = v_t(z_{\text{ref}}(t) + h_k(t)) - \chi_k[v_s(t) + c_{\text{align}}(t)] - \rho_k \dot{z}_{\text{ref}}(t)$, $Q_k^{\text{S}} = \mathcal{D}(z_{\text{ref}}(0) + h_k(0))$.

// Addition proposal

$E_k^+ = \text{Agg}_{t,\ell}\, \text{Attn}_\Theta^{t,\ell}(\text{New}(\Delta c_k) \mid c_k)$, $Q_k^{\text{A}} = \mathcal{P}_{\text{add}}(I_{k-1}, c_k, E_k^+; \alpha_{\text{add}}, \Theta)$.

// Background proposal

$(M_k^{\text{opq}}, M_k^{\text{tr}}) = \mathcal{O}_\Theta(I_{k-1}, \mathcal{L}_{k-1})$, $(w_V, w_R) = \mathcal{W}(M_k^{\text{opq}}, M_k^{\text{tr}}, r_k; \omega)$, $(\bar{V}_t, \bar{R}_t) = \big(w_V \odot V_s + (1 - w_V) \odot V_t,\, w_R \odot R_s + (1 - w_R) \odot R_t\big)$.

$Q_k^{\text{B}} = \mathcal{P}_{\text{bg}}(I_{k-1}, c_k; \bar{V}_t, \bar{R}_t, \Theta)$, $Q_k = Q_k^{T_k} \in \{Q_k^{\text{S}}, Q_k^{\text{A}}, Q_k^{\text{B}}\}$.

// Task-conditioned acceptance

$\Pi_k(Q) = (1 - H_k) \odot \big[I_{k-1} + A_k \odot (Q - I_{k-1})\big] + H_k \odot I_{k-1}$.

$I_k = \mathbf{1}[T_k = \text{SUBJECT}]\, \Pi_k(Q_k) + \mathbf{1}[T_k \in \{\text{ADDITION}, \text{BACKGROUND}\}]\, Q_k$, $\mathcal{R}_k = \text{Audit}(Q_k, I_k; E_k, A_k, H_k)$.

$\mathcal{L}_k = \text{Update}(\mathcal{L}_{k-1}, I_k, c_k, \mathcal{R}_k)$, $\mathsf{S}_k = (I_k, c_k, \mathcal{L}_k)$.

C Compared baselines: formulations and multi-turn implications

This section summarizes the five rectified-flow baselines used in our main comparison. We retain the mechanisms and objectives of the original papers, but use a common notation to make their differences explicit. Let $v_\theta(z_t, t; c)$ denote a pretrained conditional rectified-flow velocity, c_s and c_e the source and edit conditions, and z_t a latent at time $t \in [0, 1]$. These summaries are explanatory rather than reimplementations; all reported baseline results use the authors' released procedures and frozen settings described in Section 4.

C.1 FlowEdit: inversion-free velocity transport

FlowEdit (Kulikov et al., 2025) avoids first mapping the input to one fixed noise code. Instead, at every integration time it perturbs the source latent with a fresh Gaussian realization,

$$\widehat{z}_t^s = (1 - t) z_0^s + t\epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I), \tag{13}$$

and evaluates the difference between target- and source-conditioned velocity fields,

$$\Delta v_t(a, b) = v_\theta(b, t; c_e) - v_\theta(a, t; c_s). \tag{14}$$

The edited state follows the expected difference field

$$dz_t^{\text{FE}} = \mathbb{E}_{\epsilon_t}\big[\Delta v_t\big(\widehat{z}_t^s, z_t^{\text{FE}} + \widehat{z}_t^s - z_0^s\big) \big| z_0^s\big]\, dt, \qquad z_1^{\text{FE}} = z_0^s. \tag{15}$$

In practice the expectation is approximated by one or a few noise samples per step. The construction gives a short, noise-free coarse-to-fine editing path and requires neither explicit inversion nor test-time optimization. Under recursive editing, however, each accepted image becomes the next source, so the method itself does not preserve an explicit ledger of protected content; small global changes can therefore compound over turns.

C.2 StableFlow: vital-layer feature injection

StableFlow (Avrahami et al., 2025) first identifies transformer layers whose removal most changes model output. For prompts $p \in \mathcal{P}$ and seeds $s \in \mathcal{S}$, the paper defines

$$\mathrm{vit}(\ell) = 1 - \frac{1}{K} \sum_{s \in \mathcal{S},\, p \in \mathcal{P}} d(M_{\mathrm{full}}(s,p), M_{-\ell}(s,p)), \tag{16}$$

$$\mathcal{V} = \{\ell \in \mathcal{L} : \mathrm{vit}(\ell) \geq \tau_{\mathrm{vit}}\}, \tag{17}$$

where d is DINOv2 perceptual similarity, $M_{-\ell}$ bypasses layer ℓ, and $K = |\mathcal{S}||\mathcal{P}|$. Source and edited trajectories are then run in parallel, and image embeddings are injected only at the vital layers. In common notation, its selective replacement can be summarized as

$$\widetilde{h}_\ell^e = \begin{cases} h_\ell^s, & \ell \in \mathcal{V}, \\ h_\ell^e, & \ell \notin \mathcal{V}, \end{cases} \tag{18}$$

where h_ℓ^s and h_ℓ^e are source and edit image embeddings. For real-image inversion, StableFlow additionally uses inverse Euler integration and scales the initial latent by 1.15 ("latent nudging") to reduce reconstruction error. Its fixed layer set gives strong single-turn structural retention, but it does not distinguish which particular entity should be released or protected at a later turn.

C.3 MTC: dual-objective LQR and adaptive attention

MTC (Zhou et al., 2025) treats multi-turn consistency as an optimal-control problem. Given targets X_i with weights λ_i, it minimizes

$$\mathcal{J}(c) = \int_0^1 \frac{1}{2}\|c(Z_t,t)\|_2^2 dt + \sum_{i=1}^n \frac{\lambda_i}{2}\|Z_1 - X_i\|_2^2, \qquad dZ_t = c(Z_t,t)dt, \qquad Z_0 = X_0. \tag{19}$$

Writing $\lambda = \sum_i \lambda_i$ and $\widehat{X} = \lambda^{-1}\sum_i \lambda_i X_i$ reduces Equation (19) to a single terminal target; its limiting optimal controller is

$$c^*(Z_t,t) = \frac{\widehat{X} - Z_t}{1-t}. \tag{20}$$

At edit turn k, MTC uses the original image $X_{0,0}$ and previous output $X_{k-1,0}$ through

$$X_{\mathrm{dual}} = X_{0,0} + \lambda_m\big(X_{k-1,0} - X_{0,0}\big), \tag{21}$$

and accelerates the controlled integration with FireFlow. To recover editability lost to strong LQR regularization, it also modulates the next attention map using medium–low-activation layers,

$$S_{k+1,\ell} = \mathrm{softmax}\Big(QK^\top/\sqrt{d}\Big) \odot M_k. \tag{22}$$

Thus MTC explicitly couples the original and immediately preceding images, which is well suited to recursive editing. Its control target is nevertheless global: it does not maintain entity-level permissions or select a distinct proposal policy for subject editing, object addition, and background replacement.

C.4 FireFlow: one-evaluation second-order integration

FireFlow (Deng et al., 2025) accelerates rectified-flow inversion with a modified midpoint solver. Standard midpoint integration requires two neural function evaluations per interval. FireFlow instead reuses the midpoint velocity saved from the preceding interval,

$$\widehat{v}_t = v_\theta\big(z_{t-\Delta t+\Delta t/2}, t - \Delta t + \Delta t/2\big), \tag{23}$$

$$\widehat{z}_{t+\Delta t/2} = z_t + \frac{\Delta t}{2}\widehat{v}_t, \qquad z_{t+\Delta t} = z_t + \Delta t\, v_\theta\big(\widehat{z}_{t+\Delta t/2}, t + \Delta t/2\big). \tag{24}$$

After initialization, Equation (24) uses one new model evaluation per step while retaining second-order global accuracy under the paper's smoothness assumptions. FireFlow primarily improves the numerical solver; its editing behavior still depends on the surrounding inversion and feature-injection rule. Consequently, faster accurate integration does not by itself define which content may change in a multi-turn sequence.

C.5 RF-INVERSION: LQR-CONTROLLED RECTIFIED FLOWS

RF-Inversion (Rout et al., 2025) derives a controller that trades faithfulness to the observed image against consistency with the pretrained flow distribution. For terminal target Y_1, the forward control problem is

$$\min_c \int_0^1 \frac{1}{2}\|c(Z_t,t)\|_2^2 dt + \frac{\lambda}{2}\|Z_1 - Y_1\|_2^2, \qquad dZ_t = c(Z_t,t)dt, \qquad Z_0 = y_0, \tag{25}$$

whose limiting optimal controller is $c^*(Z_t,t) = (Y_1 - Z_t)/(1-t)$. The controlled inversion interpolates between the learned marginal field u_t and this conditional field,

$$dY_t = \big[u_t(Y_t) + \gamma\big(u_t(Y_t \mid Y_1) - u_t(Y_t)\big)\big]\, dt, \qquad Y_0 = y_0. \tag{26}$$

Editing uses the analogous reverse process

$$dX_t = \big[v_t(X_t) + \eta\big(v_t(X_t \mid y_0) - v_t(X_t)\big)\big]\, dt, \qquad X_0 = y_1, \tag{27}$$

where larger η favors reconstruction and smaller η permits stronger semantic change. The method supplies a principled global faithfulness–editability control, but repeated use constrains each turn toward an image-level reference rather than an evolving set of entity- and role-specific commitments.

Relationship to ReleaseEdit. The baselines above act at complementary levels: FlowEdit changes the transport path, StableFlow chooses reusable transformer layers, FireFlow changes the numerical solver, and RF-Inversion/MTC impose global optimal-control objectives. ReleaseEdit is evaluated against all five because its claimed contribution is orthogonal: it observes the current state, routes the instruction to a task-conditioned proposal policy, and applies a shared release-and-audit contract before committing the next recursive state, using localized projection where appropriate.

D DATASET CONSTRUCTION AND VISUALIZATION

We construct our evaluation set by uniformly sampling 400 source images from the MTC data pool. For each source image, ChatGPT generates a sequence of six editing instructions conditioned on the visible image content and the edit history. The resulting benchmark therefore evaluates recursive editing rather than six independent single-turn transformations: at turn k, each method receives its own accepted result from turn $k-1$ together with the next instruction.

Figure D.1 visualizes 40 source images sampled uniformly from the 400-image set with fixed seed 2027. The sample includes photographic and illustrated content spanning people, animals, objects, indoor scenes, and natural environments. Sampling is independent of method outputs, metric values, and qualitative success or failure.

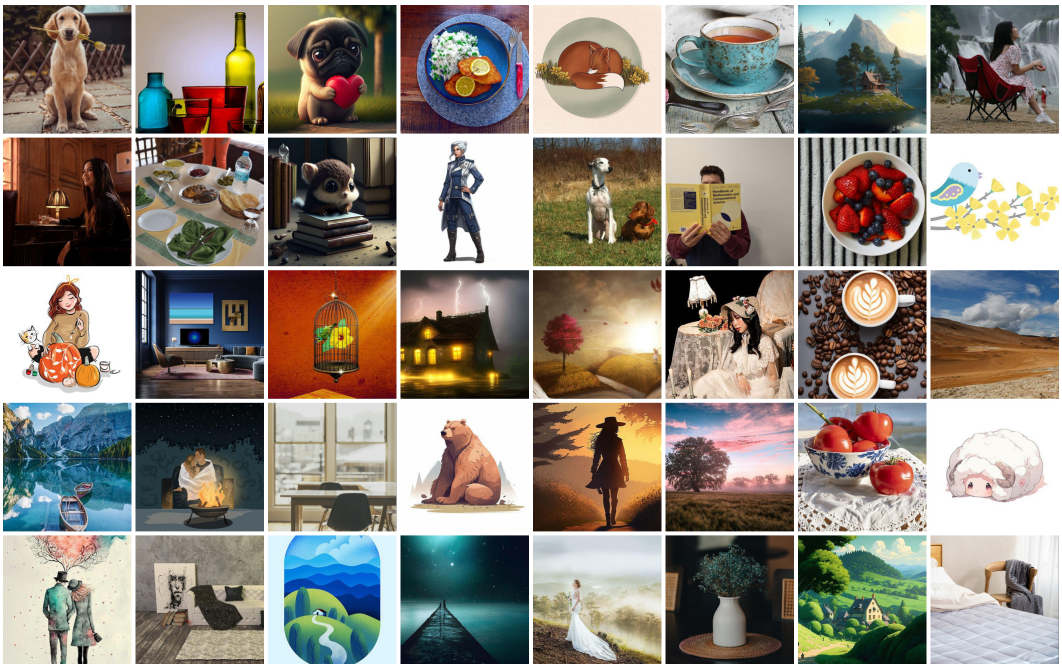

Figure D.1: **Random source images from our multi-turn benchmark.** Forty examples are uniformly sampled from the 400-image source set using a fixed random seed. Each source is paired with a six-turn instruction chain; only source images are shown here for compactness.

E ADDITIONAL ABLATION CONTROLS

Table 3: **Development sensitivity of three branch-specific controls.** Each block is a fixed single-case sensitivity sweep. Boldface and shading mark the selected operating point rather than a column-wise optimum; background scores use proposal-direct outputs, and the frozen benchmark uses $\omega = 0.02$. These values are not dataset-level component ablations. Metric definitions follow below.

Subject Editing			Object Addition			Background Replacement		
g	Edit ↑	Pres. ↑	α_{add}	Edit ↑	Pres. ↑	ω	Edit ↑	Pres. ↑
2.5	0.049	1.000	0.508	0.029	0.494	w/o	0.443	−0.581
3.5	**0.782**	**0.637**	**0.75**	**0.090**	**0.062**	**0.02**	**0.250**	**0.532**
4.5	0.937	0.292	1.0	0.108	−0.550	0.04	0.243	0.517
5.5	0.500	0.000	–	–	–	0.08	0.238	0.493

Subject guidance. Figure 5 recolors a pale-orange-and-white cat to uniformly charcoal gray while scanning target guidance over $\{2.5, 3.5, 4.5, 5.5\}$. The input, prompts, automatically estimated whole-cat release field, seed 42, 25 steps, transport reach, raw-proposal policy, and one-candidate budget are fixed. Guidance 3.5 gives the development balance: 2.5 under-edits, whereas larger values progressively darken the subject and reduce texture correlation. The benchmark therefore fixes guidance at 3.5.

Addition gain. The left panel of Figure 6 adds a red knitted scarf to a corgi while scanning incremental-direction gains $\{0.508, 0.75, 1.0\}$. The observed release field, prompt, seed, schedule, guidance, spatial-control ratio, and candidate budget are fixed. Gain 0.75 lies on the completion–preservation frontier in this development case: the smaller gain under-edits, while gain 1.0 provides only a modest completion increase at a large integrity cost.

Opaque-subject transport. The right panel of Figure 6 fixes the source image, snowy-road prompt, foreground mask, inversion, seed, 25-step budget, and guidance, then disables value transport or scans its relative interior width over $\{0.02, 0.04, 0.08\}$. The panel visualizes proposal-direct background outputs $Q_k^{\mathrm{B}} = I_k$, matching the benchmark acceptance policy. Without transport, the van is lost in the proposal despite strong background alignment; width 0.02 gives the best measured trade-off in this single case and is the frozen benchmark default. Thus the sweep diagnoses local sensitivity around the selected configuration. This opaque example does not validate the transmissive branch.

Controlled final-output transport pilot. Figure E.1 complements the raw-proposal width sweep with one final-output comparison on a held-out church image. A0 disables representation transport; A1 injects source values using raw patch occupancy, without boundary or scale adaptation; and A2 uses the complete boundary/scale-aware transport policy. The input, target background, seed 42, 25-step schedule, guidance 3.5, automatic protected mask, release field, and diagnostic projection are identical. This projection is imposed only to control the pilot comparison and is not the benchmark background acceptance policy. Background instruction gain is $0.093/0.110/0.116$ for A0/A1/A2, while the integrity diagnostic is $0.800/0.884/0.869$. Thus this case clearly exposes the failure without transport and shows that both transported variants recover the foreground; it does not establish a uniform A2-over-A1 advantage. The church occupies about 15% of the image and is classified as a large opaque entity, so A2 activates boundary-aware value transport while its scale policy assigns zero residual-transport mass. The maximum protected-pixel difference after the diagnostic projection is zero in all three variants. This pilot is a mechanism illustration, not test-set evidence.

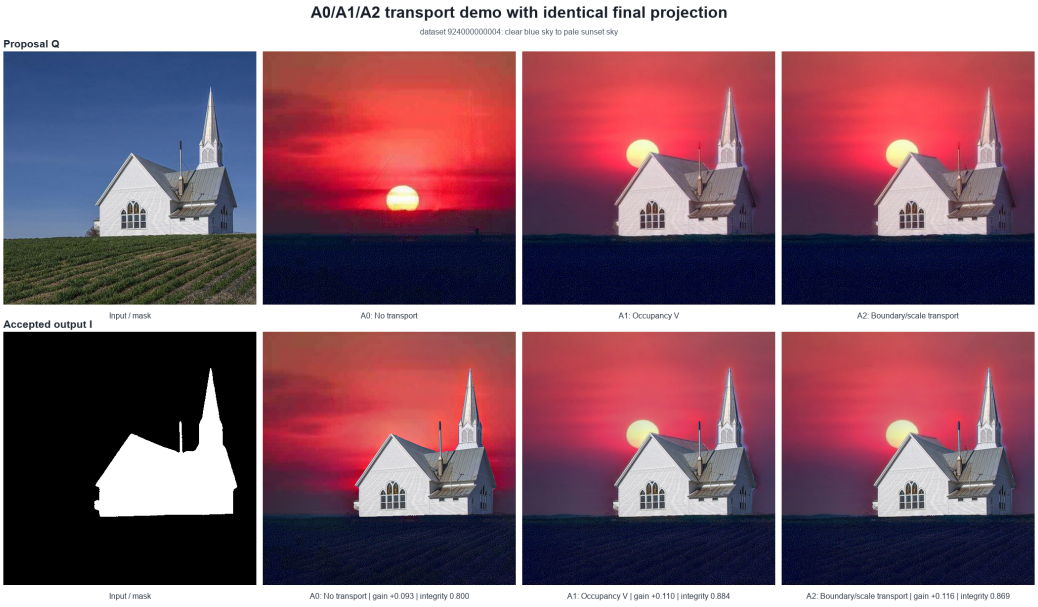

Figure E.1: **A0/A1/A2 background-transport pilot under an identical diagnostic projection.** The top row shows the raw proposals Q; the bottom row shows projected controls. A0 removes transport, A1 uses occupancy-only value injection, and A2 uses the boundary/scale-aware policy. For this large opaque foreground, A2's residual branch is inactive by design; the benchmark itself uses proposal-direct background acceptance.

Ablation metrics. For addition and background, Edit is the CLIP instruction gain: the change in candidate similarity from the source phrase to the requested phrase, minus the same change measured on the input. Preservation is the integrity diagnostic $\rho_{\mathrm{edge}} - |\log r_{\mathrm{detail}}| - 2c_{\mathrm{clip}} - p_{\mathrm{incoh}}$, combining edge correlation, detail-ratio deviation, excess clipping, and incoherent-delta fraction on the protected support. It is not lower bounded, so negative values are possible. For the charcoal-fur subject sweep, completion averages two lower-is-better diagnostics measured on the subject support: warm-residual

fraction ($R > B + 18$) and mean channel range $\max(R, G, B) - \min(R, G, B)$. Preservation averages a higher-is-better edge correlation on that support and a lower-is-better maximum change outside it. Each diagnostic d is independently min–max normalized within the frozen sweep as $N_{\uparrow}(d) = (d - d_{\min})/\max(d_{\max} - d_{\min}, 10^{-12})$ or $N_{\downarrow}(d) = 1 - N_{\uparrow}(d)$; a zero-range higher-is-better diagnostic therefore contributes 0 and its lower-is-better counterpart contributes 1. The displayed Subject Edit and Preservation values are the corresponding two-diagnostic means. Accordingly, 1.000 denotes the best normalized value in this sweep, not perfect preservation, and scores support comparisons only within a task block.

F ADDITIONAL SIX-TURN QUALITATIVE COMPARISONS

Figures F.1 and F.2 show complete mixed-operation sequences on corgi and cake images. In the corgi sequence, the instructions change subject appearance, add two objects, replace the background, and introduce a local chest marking. This combination makes both retention of earlier edits and accumulated subject drift directly visible. The ReleaseEdit R4 label notes that the background branch uses proposal-direct acceptance without a final pixel projection, matching the benchmark implementation. The cake sequence stresses appearance, background, local attributes, object count, and support-object color; its ReleaseEdit row explicitly marks the R1 pink-icing diagnostic used for that qualitative trajectory.

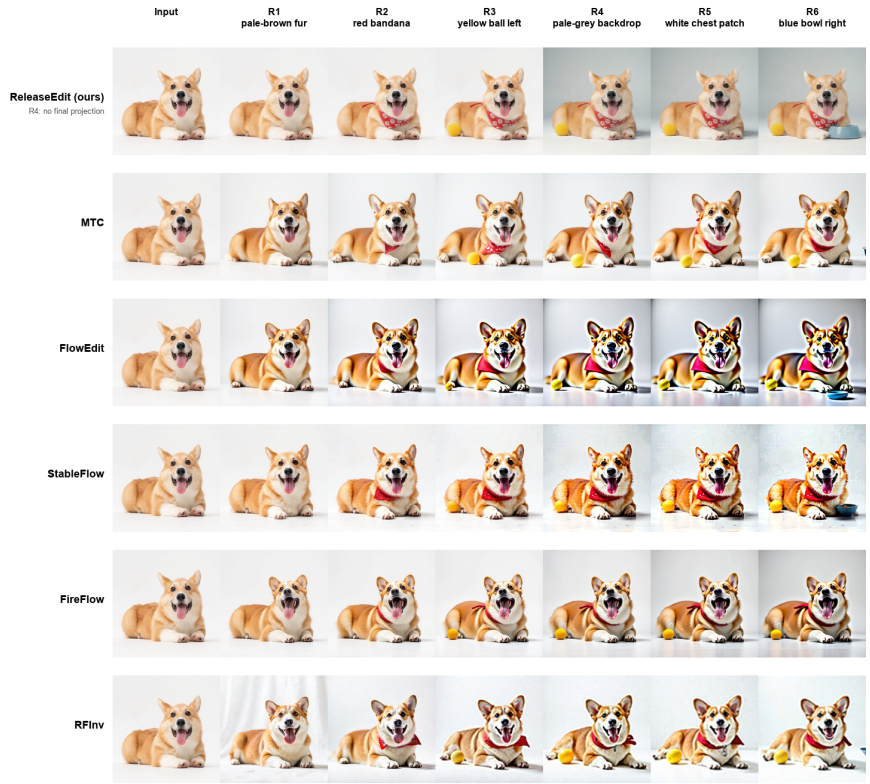

Figure F.1: **Complete six-turn qualitative comparison on the corgi sequence.** Columns show the input followed by R1–R6: pale-brown fur, a red bandana, a yellow ball on the left, a pale-grey background, a white chest patch, and a blue bowl on the right. Rows compare ReleaseEdit, MTC, FlowEdit, StableFlow, FireFlow, and RF-Inversion. ReleaseEdit R4 uses the standard proposal-direct background acceptance policy, as marked in the panel.

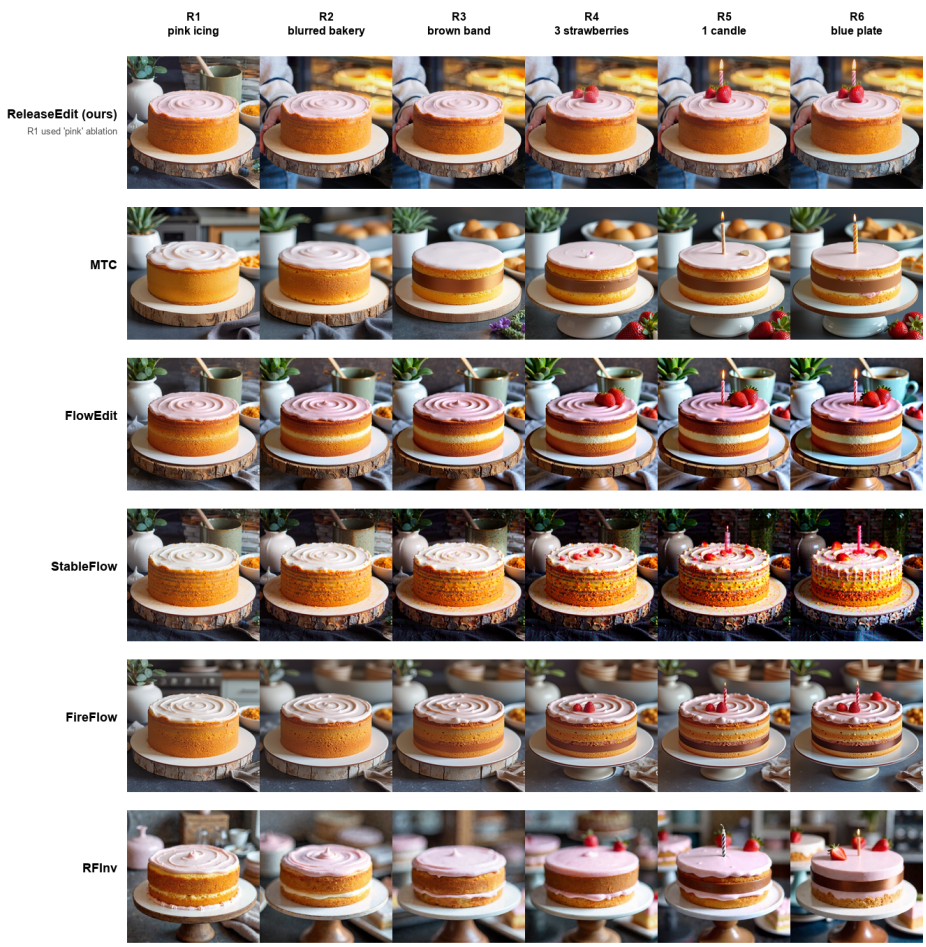

Figure F.2: **Complete six-turn qualitative comparison on the cake sequence.** Columns show R1–R6: pink icing, a blurred bakery background, a brown band, three strawberries, one candle, and a blue plate. Rows compare ReleaseEdit, MTC, FlowEdit, StableFlow, FireFlow, and RF-Inversion. The ReleaseEdit row uses the explicitly marked R1 pink-icing diagnostic variant; this panel is qualitative and is not pooled with the benchmark metrics.

G Additional ten-turn qualitative comparisons

Figures G.1 and G.2 provide the complete trajectories behind the long-horizon qualitative analysis. Every edited column is recursive: each method receives its own output from the preceding turn. The full sequences make accumulated changes in identity, geometry, color statistics, and retained objects easier to inspect than the subsampled views in the main paper.



Figure G.1: **Complete ten-turn comparison on the child sequence.** Rows correspond to ReleaseEdit, MTC, FlowEdit, FireFlow, StableFlow, and RF-Inversion; columns show the input followed by Turns 1–10. The figure is shown in full to expose long-horizon subject and scene drift that is difficult to diagnose from the final turn alone.

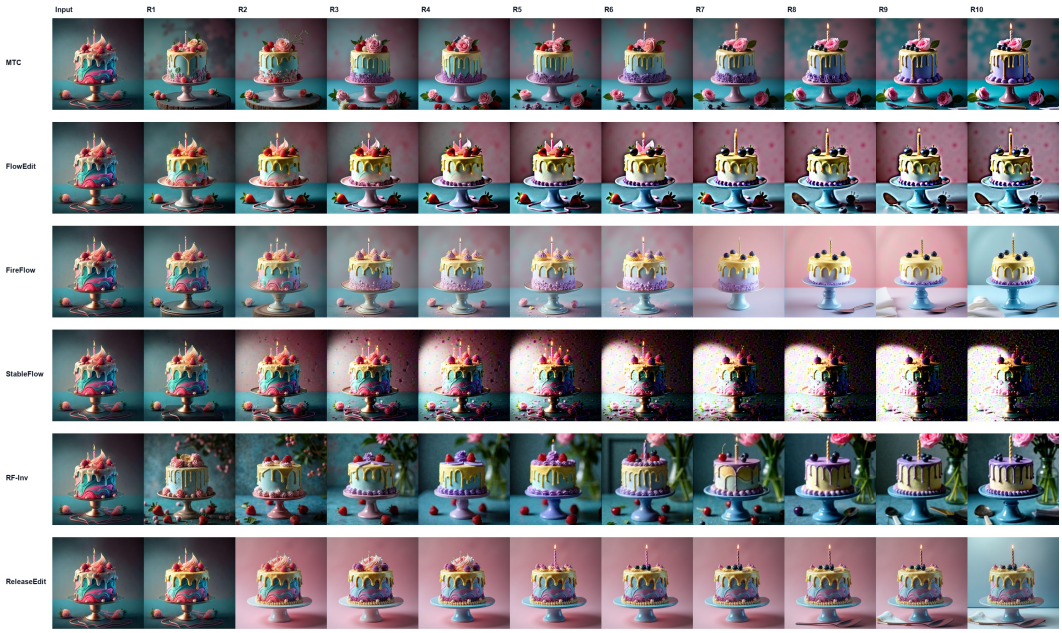

Figure G.2: **Complete ten-turn comparison on the cake sequence.** Rows correspond to MTC, FlowEdit, FireFlow, StableFlow, RF-Inversion, and ReleaseEdit; columns show the input followed by Turns 1–10. This complete trajectory complements the subsampled cake comparison in Figure 4.