PROGRESSNET: SKETCHING AND PROMPTING WITH A FROZEN TEXT-TO-IMAGE MODEL

Anonymous authors
Paper under double-blind review

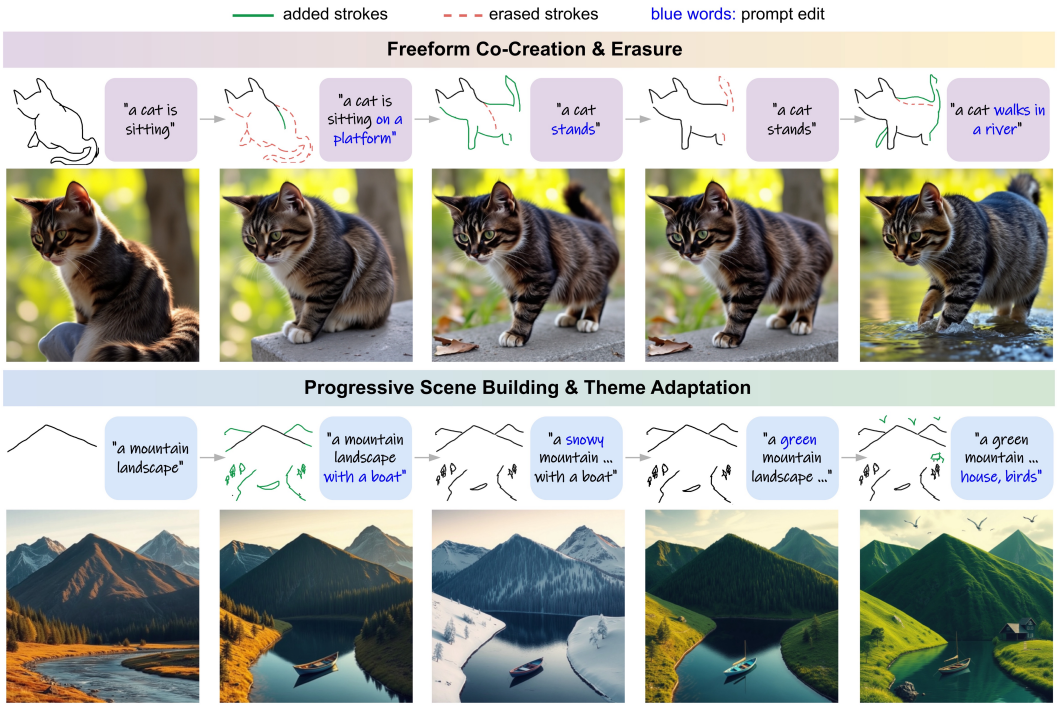


Figure 1: ProgressNet follows a drawing session turn by turn, training-free on frozen FLUX. **Top**: the cat's body is erased and it moves onto a platform, stands once legs are drawn, loses its tail when it is erased, and walks into a river once the tail is redrawn. **Bottom**: a scene is built stroke by stroke, re-themed by the prompt from snowy to green, and finished with a house and birds.

ABSTRACT

Humans draw progressively: a few strokes, a look at the result, a stroke erased, a prompt revised. Image generators do not work this way. They typically take a finished sketch and produce the image in a single pass, so every edit starts the picture again, and the models that do keep state across turns are driven by text, cannot take a stroke, and are too slow to draw with. We present *ProgressNet*, a training-free framework that lets a frozen text-to-image model follow a drawing session as it unfolds: strokes are added and erased, the prompt is revised, and the image keeps up at about a second per turn. It needs no new parameters because the frozen model already has what a progressive generator needs, a pathway through which the previous turn can be remembered, layers that can carry appearance forward without freezing structure, and an internal signal of how far to trust an unfinished sketch; three inference-time mechanisms (*Previous-Concept Memory*, *Layer-Selective K/V Injection* and *Banded Adaptive Control*) use each in turn. As a sketch fills in, every existing method degrades, the FID of the FLUX+ControlNet baseline doubling between 10% and 100% completion on FS-COCO, while ProgressNet's barely moves; it maintains strong fidelity and progressive coherence across three sketch domains and is preferred by users over five competitors, most widely on erasure.

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.

1 INTRODUCTION

Humans are visual creatures. We drew before we wrote, and drawing remains the most natural way to think through a visual idea and share it. Generative AI has opened a remarkable channel for this: a sentence now produces a picture (Rombach et al., 2022; Black Forest Labs, 2024), in well under a second with few-step distillation (Luo et al., 2023; Sauer et al., 2023), and because a sentence is a poor description of a picture, control has steadily grown beyond text, to edges, depth and layout (Zhang et al., 2023; Mou et al., 2024), to reference images (Ye et al., 2023; Black Forest Labs et al., 2025), and most recently to a sketch drawn beside the prompt box (OpenAI, 2026). Sketch is the natural end of that road: it is how people convey shape and intent when words run out (Eitz et al., 2012; Koley et al., 2024). Yet every one of these systems takes the sketch as a finished input and produces the image in a single pass, and nobody draws that way. A drawing is made over time: a few strokes, a look, a stroke erased, a line redrawn, often a change of mind about what the picture is of (Fig. 1).

What a person needs is not a picture from a sketch but a picture that keeps up with the sketching: one that follows each added stroke, lets go of each erased one, absorbs a revised prompt, leaves everything else untouched, and answers fast enough to feel like a response rather than a resubmission. This paper is about that interaction. *ProgressNet* lets a frozen text-to-image model support all of it, drawing, erasing and prompt revision, at about a second per turn and with no training.

To the best of our knowledge, no existing system does all. Sketch-conditioned generation (Zhang et al., 2023; Koley et al., 2024; Li et al., 2024) is stateless: each turn regenerates the scene from nothing, so appearance drifts with every stroke and an erased stroke leaves a ghost, because nothing withdraws the evidence it added (Fig. 2). Interactive sketch systems lower the cost of regeneration without removing its statelessness (Sarukkai et al., 2024; Song et al., 2024), and Subjective Camera (Chen et al., 2025b) follows a drawing order by latent optimisation that takes minutes per scene. Image editing models (Black Forest Labs et al., 2025; Zhou et al., 2025; Liao et al., 2026; Xu et al., 2026) do handle turn-by-turn interaction, but for a different kind of turn: the image enters as a second token stream in joint attention and the change is given in words, so there is no pathway for a stroke map and no way to say that a stroke is gone, and tens of denoising steps over a doubled sequence make each turn take seconds rather than the fraction of a second a drawing person will wait. ProgressNet therefore builds on the architecture that accepts strokes, a frozen flow-matching backbone with a ControlNet, and stays in the few-step regime that makes interaction possible.

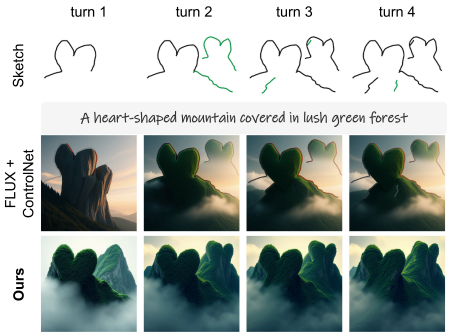
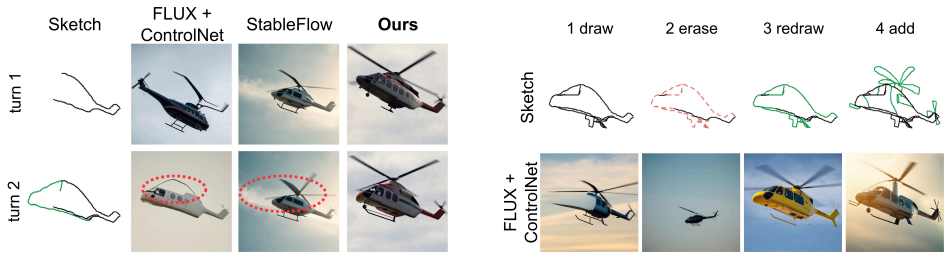


Figure 2: Flux+ControlNet imprints raw stroke edges and visually drifts across progressive turns; ours adapts smoothly while maintaining temporal coherence.

Following a drawing session with a frozen model raises three problems, and each has a simple answer inside the model. *Remembering.* The model must remember what it generated last turn. The ControlNet's tile branch accepts an RGB image, so the previous output can be fed back as an explicit appearance memory, and because that memory lives in pixel space the user's operations map onto it directly: an added stroke keeps it, an erased stroke white-fills its region, a new prompt relaxes the regions the user has not committed to (*Previous-Concept Memory*, PCM; Section 4.1). *Carrying forward.* The model must carry that memory without freezing what should change. Training-free editors inject cached attention at the layers most vital to forming the image (Avrahami et al., 2025); under progression those are exactly the layers that must absorb new strokes, and locking them yields images that stop responding as the drawing grows. Injecting at the least vital layers instead keeps appearance while structure stays free (*Layer-Selective K/V Injection*, LS-KVI; Section 4.2). *Trusting the sketch.* The model must balance adherence to an unfinished sketch with preservation of previous appearance. We regulate this balance indirectly, using changes in cached ControlNet memory features between the two preceding completed turns to modulate memory strength across layer bands and solver steps, while structural control remains fixed (Banded Adaptive Control, BAC; Section 4.3). Together the three add 0.068 s to a two-step FLUX generation.



(a) Progressive sketching. (b) Stroke removal and re-addition.

Figure 3: Pilot studies illustrating failure modes of existing methods: temporal drift, prompt-sketch conflict under partial strokes, and contradiction upon stroke removal. **(a)** Under progressive sketching, *FLUX+ControlNet* and *StableFlow* show temporal instability and artifacts on the partial strokes (red borders), while ours remains stable across turns. **(b)** When strokes are removed and re-added (Seq. 1-3), *FLUX+ControlNet* fails to preserve established concepts, shifting identity across turns.

The result is a generator whose quality does not decay as the drawing fills in: on FS-COCO (Chowdhury et al., 2022) the FID of FLUX+ControlNet doubles between 10% and 100% completion and every baseline degrades the same way, while ProgressNet's moves by under one point. Because frame-to-frame similarity alone rewards a method that ignores the new stroke, we introduce a progressive fidelity-coherence score (PFC) combining temporal coherence with completion-level distributional quality. Our contributions are *(i)* the first training-free system for progressive generation from sketch and text that supports stroke addition, erasure and prompt revision at interactive rates; *(ii)* PCM, LS-KVI and BAC, with the findings that vital-layer injection over-locks under progression and that inter-turn ControlNet feature redistribution is a usable online control signal, validated across two backbones and four sketch domains; *(iii)* a protocol and score for progressive generation, with results on three sketch domains and user study in which ProgressNet is preferred over five competitors, most widely on erasure. None of the three mechanisms is sketch-specific.

2 Related Work

Controllable image generation spans ControlNet-style residual conditioning (Zhang et al., 2023), multi-condition control (Zhao et al., 2023), lightweight adapters (Mou et al., 2024; Ye et al., 2023), layout grounding (Li et al., 2023b; Bar-Tal et al., 2023), and training-free feature steering (Mo et al., 2024; Tumanyan et al., 2023), recently ported to DiT and rectified-flow backbones (Esser et al., 2024; Black Forest Labs, 2024; Tan et al., 2025). Inversion-based editing (Hertz et al., 2023), attention injection (Avrahami et al., 2025), and few-step distillation (Song et al., 2024; Sauer et al., 2023) improve editability and latency, bringing sketch interaction near real time (Sarukkai et al., 2024). Others mitigate text-sketch conflict by re-weighting cross-attention (Chefer et al., 2023; Phung et al., 2024) or rescaling the control residual (Liu et al., 2024; Cohen et al., 2025). All operate single-shot: none tracks how sketch–text conflict evolves across turns, nor preserves prior content as the condition changes. ProgressNet targets this non-stationary setting via training-free temporal memory, selective feature reuse, and ControlNet-driven adaptive control, detailed in Appendix E.

3 Preliminaries

ProgressNet builds on flow-matching backbones (Black Forest Labs, 2024; Bandyopadhyay et al., 2025; Lipman et al., 2023) with Union ControlNet. Joint attention over latent and context tokens exposes the K/V features later reused by LS-KVI (Section 4.2), while spatial conditions such as Canny edges and tile memory are introduced through ControlNet residuals, $\mathbf{h}' = \mathbf{h} + \lambda \mathbf{r}$. This formulation also makes the control strength λ a critical factor under few-step inference, where over- or under-conditioning cannot be corrected over many denoising steps (Section 4.3).

Our pilot studies (Figs. 3a and 3b) show that existing formulations struggle to maintain a coherent progressive state: appearance can drift between turns, a fixed control strength can either imprint partial strokes too strongly or under-enforce them, and erased content may persist when the sketch is revised. ProgressNet addresses these coupled failures through explicit concept memory, selective temporal feature reuse, and adaptive control modulation.

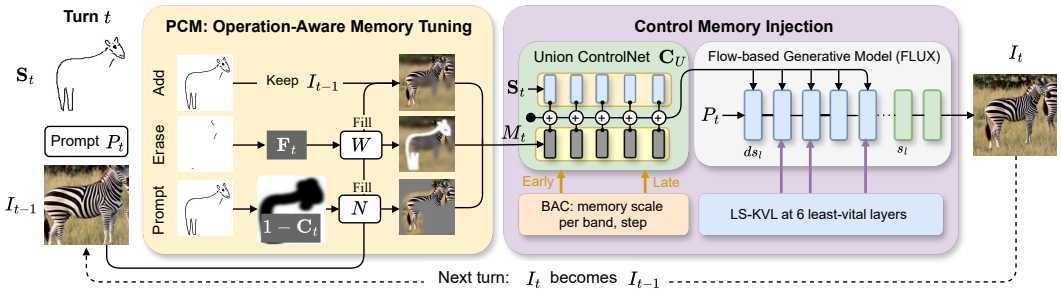


Figure 4: Overview of ProgressNet. **Left**: the previous output I_{t-1} is prepared according to the current operation: addition keeps I_{t-1}, removal white-fills (W) the erased region $\mathbf{F}_t$, and a prompt update gray-fills (N) the uncommitted region $1 - \mathbf{C}_t$. **Middle**: the memory M_t and current sketch $\mathbf{S}_t$ condition Union ControlNet $\mathbf{C}_U$, while BAC scales the memory stream per layer band and solver step. **Right**: LS-KVI reuses previous-turn K/V features at six low-vitality FLUX layers.

4 Proposed Methodology

Overview. Given a sketch $\mathbf{S}_t$, prompt P_t, and previous output I_{t-1}, we generate $I_t = \mathcal{G}(\mathbf{S}_t, P_t, I_{t-1}; \theta)$ while keeping pretrained weights θ frozen and supporting stroke addition, removal, and prompt updates. ProgressNet combines three mechanisms: (i) *PCM* uses operation-aware tile conditioning to preserve prior appearance (Section 4.1); (ii) *LS-KVI* reuses previous-turn features only at selected layers to maintain context while permitting structural change (Section 4.2); and (iii) BAC applies retrospective, step- and band-specific modulation to the memory-conditioning ControlNet stream using cached inter-turn comparisons.

4.1 Previous-Concept Memory (PCM)

Progressive sketching is underdetermined: partial strokes constrain geometry but say little about appearance or context, so regenerating each turn independently can alter established content (Koley et al., 2024; Liu et al., 2025; Luo et al., 2023). Existing sketch-control methods (Zhang et al., 2023; Zhao et al., 2023) condition only on current strokes, with no memory of previous turns.

Memory-induced correction. We retain the previous generation I_{t-1} as appearance memory, adapted per interaction: adding strokes preserves appearance, removing strokes withdraws obsolete evidence, and prompt updates retain only committed content. Let $\mathbf{F}_t$ be the erased region, $\mathbf{C}_t$ the soft commitment mask, and W, N white and gray images. The memory condition is:

$$M_t: \quad s_A \mapsto I_{t-1}, \qquad s_R \mapsto (1-\mathbf{F}_t)\odot I_{t-1} + \mathbf{F}_t \odot W, \qquad p_U \mapsto \mathbf{C}_t \odot I_{t-1} + (1-\mathbf{C}_t)\odot N. \quad (1)$$

Where, s_A, s_R, and p_U denote stroke addition, removal, and prompt update, respectively; $\mathbf{C}_t$ marks committed content derived from the current processed sketch (Section B.1). Neutral filling withdraws obsolete evidence, while $\mathbf{C}_t$ permits uncommitted regions to adapt to the updated prompt.

Memory-structure fusion. The prepared RGB memory M_t is passed through the tile branch of Union ControlNet ($\mathbf{C}_U$), while the current sketch $\mathbf{S}_t$ supplies the binary structural sketch condition. Their residuals are fused at each controlled layer:

$$\mathbf{h}'_\ell = \mathbf{h}_\ell + \lambda_s \, \mathbf{r}^\ell_{\text{str}}(\mathbf{S}_t) + \lambda_m \, \mathbf{r}^\ell_{\text{mem}}(M_t), \quad (2)$$

where λ_s and λ_m weight structural and appearance evidence, respectively. The current sketch thus determines *what should change*, while the PCM determines *what should remain*. Since the memory is an RGB image, it can be edited directly according to the user operation before conditioning, with no extra learned module or additional learned preprocessing. PCM thus turns a stateless sketch-conditioned generator into an operation-aware progressive one, entirely at inference time.

4.2 Layer-Selective K/V Injection (LS-KVI)

PCM anchors successive generations via memory conditioning, but internal representations can still drift as attention is recomputed at each step. Training-free editors reduce such drift by reusing attention features, either injecting source features into key DiT layers (Avrahami et al., 2025) or reusing layer-specific K/V features (Wei et al., 2025; Dalva et al., 2025; Kim et al., 2025). However, they target prompt-only editing, not the control-driven edits central to progressive drawing.

From image editing to progressive control. Progressive generation differs from text-guided editing. In editing, feature injection preserves a source image while text updates selected semantics. In progressive control, each turn adds a new spatial signal via ControlNet, and new strokes may change geometry, pose, or even the concept. Carrying strong previous-turn features through influential layers can thus conflict with the current structure and over-preserve content that should evolve, as we observe when adapting vital-layer injection to progressive sketching (Fig. 5). This suggests a different principle: progressive generation should not freeze the layers most responsible for forming the current concept. We therefore leave vital layers without direct temporal K/V injection and reuse features at selected lower-vitality layers, assessing the resulting preservation and adaptation empirically.

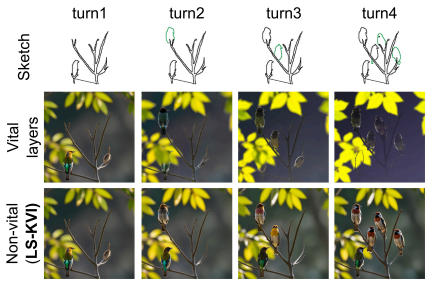


Figure 5: Vital vs. non-vital layer. Vital-layer injection locks in early details, losing adaptability, while non-vital LS-KVI preserves appearance and sharpness as structure follows sketch.

Progressive layer selection. Following the residual-bypass analysis used to study layer importance (Avrahami et al., 2025), we measure layer vitality directly under progressive controlled generation rather than prompt-only editing. For each transformer layer ℓ, we bypass that layer through its residual connection and measure the output change across progressive sketch stages. Let $\hat{I}^{\mathrm{ref}}_{x,k}$ denote the reference generation for sketch x at progression stage k, and let $\hat{I}^{(-\ell)}_{x,k}$ denote the corresponding generation with layer ℓ bypassed. We define:

$$\mathcal{V}(\ell) = \frac{1}{|\mathcal{D}||\mathcal{S}|} \sum_{x\in\mathcal{D}} \sum_{k\in\mathcal{S}} \left\| \hat{I}^{\mathrm{ref}}_{x,k} - \hat{I}^{(-\ell)}_{x,k} \right\|_2^2. \tag{3}$$

where $\mathcal{D}$ denotes the set of progressive sequences and $\mathcal{S}$ the evaluated completion stages. We use $\mathcal{V}(\ell)$ as a layer-selection criterion: lower scores identify layers whose bypass causes smaller output deviations in the evaluated progressive setting. We select L_{inj} from these layers and assess the injection choice separately through the vital vs. non-vital comparisons in Section B and Fig. 10.

Selective temporal injection. For each turn $t > 0$, we cache the preceding K/V projections and inject them only for $\ell \in \mathcal{L}_{\mathrm{inj}}$ as $\mathbf{K}^{t\prime}_\ell = \mathbf{K}^t_\ell + \alpha_{\mathrm{kv}}\mathbf{K}^{t-1}_\ell$ and $\mathbf{V}^{t\prime}_\ell = \mathbf{V}^t_\ell + \alpha_{\mathrm{kv}}\mathbf{V}^{t-1}_\ell$, while layers outside L_{inj} receive no direct temporal K/V injection. The cache is refreshed after every turn and cleared upon stroke removal; this prevents stale K/V correspondences from resisting erasure, as quantified in Table 8.

Putting it all together, PCM carries the previous concept through external conditioning, whereas LS-KVI reuses internal attention features at selected layers. Neither, however, determines how strongly the preceding memory should constrain generation with progressive drawing. We address this through adaptive memory control in Section 4.3.

4.3 Banded Adaptive Control (BAC)

PCM and LS-KVI preserve information across turns, but PCM retains a fixed memory-control strength. During progressive sketching, however, how strongly the preceding appearance should constrain the current generation changes as new structural evidence arrives. Rather than learning a mixing policy or modifying conditioning flow (Liu et al., 2024; Cohen et al., 2025), we ask whether the pretrained model itself reveals how its control response changes under progressive updates.

Control is band- and step-asymmetric. We first study how ControlNet changes the backbone during progressive generation. Let $\mathbf{h}^{\mathrm{ctrl}}_\ell(t,s)$ and $\mathbf{h}^{\mathrm{base}}_\ell(t,s)$ denote hidden states at transformer layer ℓ, progressive stage t, and solver step s, with and without spatial control. We measure the induced perturbation as $\Delta^{(t,s)}_\ell = \mathbf{h}^{\mathrm{ctrl}}_\ell(t,s) - \mathbf{h}^{\mathrm{base}}_\ell(t,s)$. The perturbation reflects the backbone response across depth (Fig. 11) and varies across layer groups and solver steps (Fig. 6). Complementary diagnostics show related asymmetries on SD3.5 and four sketch domains (Sections C.2 and C.3). These observations motivate band- and step-specific memory modulation rather than a single global factor.

Retrospective ControlNet feature comparison. BAC must determine its factor before sampling. A current-turn band statistic is available only after that band is traversed, requiring replay to apply it

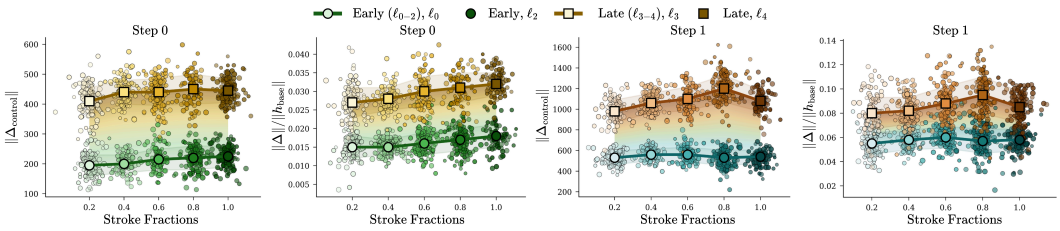


Figure 6: Layer- and solver-step variation in control responses. Control-induced perturbations and their normalized magnitudes vary across the evaluated layer groups and solver steps during progression. Appendix Sections C.2 and C.3 report complementary feature-redistribution diagnostics on SD3.5 and across four sketch domains.

consistently. We therefore use the latest complete transition, $t{-}2 \to t{-}1$, as a causal proxy for recent memory redistribution. Let $\bar{\mathbf{c}}^k_{u,s,\text{mem}}$ denote the raw memory-conditioning feature cached after turn u from ControlNet block k at step s. For each block k and step s, we compute channel-wise JSD,

$$J^k_{t,s} = \text{JSD}_{\text{cw}}\left(\bar{\mathbf{c}}^k_{t-2,s,\text{mem}}, \bar{\mathbf{c}}^k_{t-1,s,\text{mem}}\right), \qquad f^b_{t,s} = \text{clip}\left(\frac{1}{|\mathcal{K}_b|}\sum_{k\in\mathcal{K}_b} J^k_{t,s}, f_{\min}, f_{\max}\right). \tag{4}$$

Here, JSD_{cw} averages divergence across feature channels, $\mathcal{K}_b$ denotes blocks in $b \in \{\text{early}, \text{late}\}$ and $f_{\min}, f_{\max}$ bound modulation. Larger recent redistribution indicates a less stable appearance state, retaining stronger memory conditioning to counter drift, while stable representations permit attenuation. BAC modulates only the current memory-conditioning stream, $\tilde{\mathbf{c}}^k_{t,s,\text{mem}} = f^{b(k)}_{t,s}\mathbf{c}^k_{t,s,\text{mem}}$, where $b(k)$ maps block k to its band. Accordingly, Eq. (2) becomes

$$\mathbf{h}'_{\ell,s} = \mathbf{h}_{\ell,s} + \lambda_s\, \mathbf{r}^\ell_{\text{str}}(\mathbf{S}_t) + \lambda_m\, \tilde{\mathbf{r}}^\ell_{\text{mem},s}(M_t); \qquad \{\tilde{\mathbf{c}}^k_{t,s,\text{mem}}\}_k \longmapsto \tilde{\mathbf{r}}^\ell_{\text{mem},s}. \tag{5}$$

BAC produces four factors under two-step inference: $f^{\text{early}}_{t,0}, f^{\text{late}}_{t,0}, f^{\text{early}}_{t,1}, f^{\text{late}}_{t,1}$.

5 Experiments

Datasets. We evaluate on three sketch domains: FS-COCO (Chowdhury et al., 2022) (freehand scene sketches), Photo-Sketching (Li et al., 2019) (edge-aligned contour sketches), and Sketchy (Sangkloy et al., 2016) (single-object freehand sketches). Together, they span scene- and object-level generation as well as freehand and contour-based representations.

Implementation Details. We use FLUX.1-schnell (Black Forest Labs, 2024) with Union ControlNet at $1024{\times}1024$ resolution and two denoising steps. We set $\alpha_{kv}{=}0.3$, $f_{\min}{=}0.05$, and $f_{\max}{=}1.0$. Experiments are conducted on a single NVIDIA H200 GPU.

Evaluation Protocol and Metrics. We evaluate generation quality at specific sketch-completion stages (Table 1) and progressive stability across representative transitions (Table 2), while preserving stroke order and captions. We measure distributional quality with FID-I (Heusel et al., 2017) and FID-C (Kynkäänniemi et al., 2023), semantic fidelity with DINOv2 (Oquab et al., 2024), and text alignment with CLIPScore (Hessel et al., 2021; Radford et al., 2021) and T5-Cos (BLIP-2 captions (Li et al., 2023a) + T5 embeddings (Raffel et al., 2020)). For progression, PFC-DINO and PFC-LPIPS measure temporal stability. We additionally report quality-adjusted cross-completion robustness in distributional (Dist.), visual (Vis.), textual (Text), and overall dimensions, normalized to the FLUX baseline (100). Full definitions are provided in Appendix A.2.

5.1 Quantitative Results

We compare against Sketch-a-Sketch (S-a-S) (Sarukkai et al., 2024), SDXS (Song et al., 2024), Conditional Balance (CB) (Cohen et al., 2025), FLUX+ControlNet (Black Forest Labs, 2024; Zhang et al., 2023), and StableFlow (SF) (Avrahami et al., 2025). Table 1 evaluates generation quality at 10%, 50%, 100% sketch completion across three sketch datasets. ProgressNet maintains strong semantic fidelity: DINOv2 is highest or tied-highest at nearly all completion levels, with competitive CLIP/T5 alignment. This suggests the method stays faithful to the target image and prompt throughout the drawing, rather than only at the final drawing state. Table 2 presents progressive fidelity and inter-turn coherence using PFC-DINO and PFC-LPIPS, which assess coherence while

Table 1: Comparison with SOTA methods across sketch completion levels and domains. Latency is measured per turn, independent of completion level. **Bold**: best; underline: second best.

	10% Completion						50% Completion						100% Completion					
Metric	S-a-S	SDXS	CB	Flux	SF	**Ours**	S-a-S	SDXS	CB	Flux	SF	**Ours**	S-a-S	SDXS	CB	Flux	SF	**Ours**
FS-COCO																		
FID-I ↓	192.3	40.3	195.9	**32.9**	36.8	36.5	200.1	41.4	209.9	44.7	40.7	**36.0**	219.2	49.6	208.3	67.9	52.8	**37.3**
FID-C ↓	0.432	0.161	0.400	**0.149**	0.185	0.152	0.434	0.170	0.396	0.188	0.201	**0.141**	0.475	0.188	0.407	0.264	0.217	**0.143**
DINOv2 ↑	0.286	0.331	0.328	0.375	0.397	**0.400**	0.305	0.337	0.340	0.386	0.384	**0.415**	0.276	0.328	0.317	0.361	0.359	**0.413**
CLIPScore ↑	0.267	0.299	0.293	0.298	0.301	**0.306**	0.268	0.297	0.291	0.285	0.296	**0.305**	0.256	0.291	0.289	0.265	0.289	**0.303**
T5-Cos ↑	0.637	0.649	0.645	0.651	**0.659**	0.657	0.631	0.647	0.637	0.642	**0.656**	**0.656**	0.630	0.645	0.640	0.641	0.654	**0.655**
Photo-Sketching																		
FID-I ↓	183.9	194.9	173.4	**66.0**	81.0	76.2	188.5	188.7	176.7	**70.6**	**70.6**	73.3	193.4	195.0	175.5	77.8	77.5	**72.5**
FID-C ↓	0.358	0.294	0.335	**0.116**	0.171	0.148	0.364	0.287	0.333	**0.121**	0.156	0.137	0.374	0.303	0.334	0.136	0.139	**0.135**
DINOv2 ↑	0.259	0.259	0.341	0.325	**0.381**	**0.381**	0.275	0.283	0.326	0.355	0.355	**0.382**	0.274	0.268	0.326	0.381	0.354	**0.382**
CLIPScore ↑	0.285	0.310	0.321	0.319	0.325	**0.334**	0.281	0.311	0.317	0.310	0.323	**0.332**	0.278	0.308	0.315	0.309	0.323	**0.331**
T5-Cos ↑	0.721	0.725	0.777	0.760	**0.781**	0.780	0.714	0.726	0.770	0.760	0.777	**0.784**	0.710	0.728	0.767	0.766	**0.783**	**0.783**
Sketchy																		
FID-I ↓	123.8	119.7	117.1	104.7	**100.5**	101.6	125.3	121.1	116.4	111.8	115.2	**100.2**	124.1	122.0	116.4	115.5	137.1	**99.6**
FID-C ↓	**0.264**	0.282	0.378	0.267	0.299	0.292	0.281	0.280	0.378	0.277	0.308	**0.270**	0.281	0.281	0.382	0.292	0.334	**0.271**
DINOv2 ↑	0.339	0.360	0.425	0.446	0.474	**0.479**	0.345	0.361	0.433	0.447	0.430	**0.487**	0.358	0.365	0.434	0.451	0.395	**0.496**
CLIPScore ↑	0.269	0.291	**0.307**	0.297	0.301	0.302	0.268	0.289	**0.306**	0.293	0.291	0.300	0.268	0.289	0.299	0.290	0.281	**0.316**
T5-Cos ↑	0.526	0.520	0.577	0.522	0.524	**0.627**	0.523	0.520	0.529	0.517	0.525	**0.626**	0.526	0.519	0.523	0.517	0.524	**0.623**
Latency (s) ↓	2.525	**0.049**	9.483	0.761	0.762	0.829			*completion-invariant*				**Ours: +0.068 s vs. Flux**					

Table 2: Progressive stability across sketch domains. PFC-DINO and PFC-LPIPS evaluate temporal coherence and generation quality at the three reported FID-I anchors. Cross-Comp. reports cross-completion robustness, normalized to the Flux baseline (100). **Bold**: best; underline: second best.

		PFC-DINO ↑				PFC-LPIPS ↑				Cross-Comp. Robustness ↑			
Dataset	Method	10►20	30►50	80►100	Mean	10►20	30►50	80►100	Mean	Dist.	Vis.	Text	Overall
FS-COCO	Sketch-a-Sketch	0.237	0.228	0.222	0.229	0.331	0.322	0.299	0.317	39.51	76.22	97.50	66.47
	SDXS	1.883	1.824	1.579	1.762	1.864	1.820	1.546	1.744	133.14	90.18	105.10	108.06
	Cond. Balance	0.261	0.236	0.229	0.242	0.318	0.298	0.299	0.305	42.74	87.68	104.06	73.06
	Flux Baseline	1.836	1.387	0.999	1.407	1.896	1.425	0.982	1.435	100.00	100.00	100.00	100.00
	StableFlow	2.198	2.120	1.682	2.000	2.174	2.151	1.726	2.017	119.70	100.23	105.41	108.14
	Ours	**2.244**	**2.333**	**2.343**	**2.307**	**2.190**	**2.257**	**2.223**	**2.223**	**167.49**	**110.64**	**107.99**	**126.02**
Photo-Sketch	Sketch-a-Sketch	0.202	0.202	0.207	0.204	0.343	0.337	0.334	0.338	37.63	78.94	92.07	64.91
	SDXS	0.325	0.345	0.367	0.345	0.384	0.403	0.403	0.397	41.45	78.41	97.88	68.26
	Cond. Balance	0.430	0.406	0.440	0.425	0.501	0.490	0.512	0.501	41.39	97.56	101.79	74.35
	Flux Baseline	0.735	0.732	0.749	0.739	0.932	0.887	0.838	0.886	**100.00**	100.00	100.00	100.00
	StableFlow	1.001	**1.229**	1.163	**1.131**	1.029	**1.285**	1.201	**1.172**	86.26	105.74	103.73	98.17
	Ours	**1.024**	1.095	**1.183**	1.101	**1.085**	1.154	**1.206**	1.148	95.80	**114.76**	**105.24**	**104.98**
Sketchy	Sketch-a-Sketch	0.384	0.377	0.407	0.389	0.509	0.505	0.516	0.510	96.54	76.17	96.74	89.27
	SDXS	0.675	0.644	0.674	0.664	0.646	0.634	0.657	0.646	98.37	80.71	99.99	92.59
	Cond. Balance	0.759	0.728	0.752	0.746	0.773	0.758	0.780	0.770	86.37	95.70	101.65	94.36
	Flux Baseline	0.634	0.591	0.613	0.613	0.587	0.556	0.556	0.566	100.00	100.00	100.00	100.00
	StableFlow	0.843	0.764	0.657	0.755	**0.813**	0.772	0.674	0.753	86.62	90.37	98.94	91.84
	Ours	**0.868**	**0.886**	**0.928**	**0.894**	0.795	**0.830**	**0.881**	**0.835**	**106.40**	**107.77**	**111.16**	**108.42**

also accounting for quality. ProgressNet obtains the highest mean PFC-DINO/PFC-LPIPS on FS-COCO and Sketchy, and the highest final-transition scores across all three sketch domains. The cross-completion indices additionally summarize quality and its variation across completion levels.

5.2 Qualitative Results

Figs. 7 and 8 examine how progressive memory supports generation as the sketch is continuously extended, revised, or locally edited. In Fig. 7, comparisons against standard baselines show that ProgressNet better preserves appearance and scene context during progressive sketching and remains coherent when previously drawn strokes are removed. Fig. 8 extends this evaluation to cross-method comparisons and more demanding interactions against Nano Banana Pro, including fine-grained giraffe and scene-level edits as well as open-ended creative generation. These examples also show reduced temporal drift and stroke-induced artifacts under progressive updates. Across these cases, the progressive sketching process allows local structure to evolve without repeatedly reinterpreting the established image, while the retained visual state provides an important basis for subsequent edits. Together, these illustrate the practical benefit of progressive memory: users can

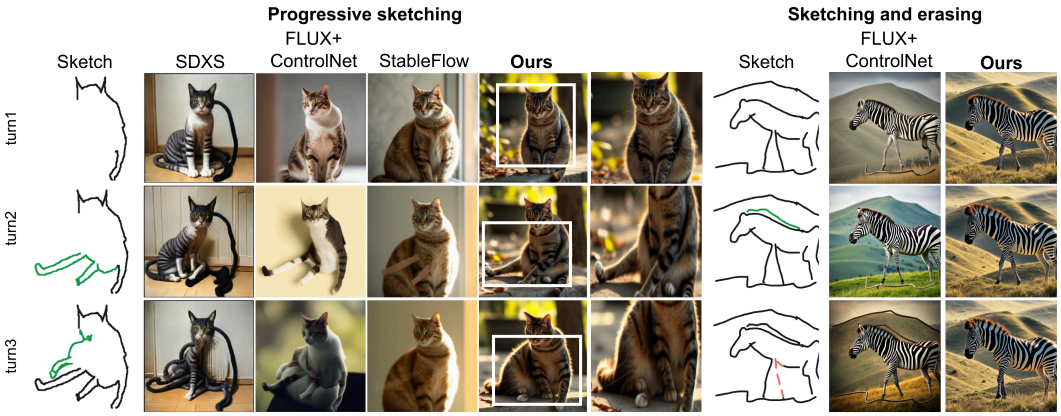


Figure 7: Progressive sketching (left) and sketch removal (right). Competing methods frequently drift or introduce structural artifacts as strokes evolve, while ProgressNet preserves appearance and scene consistency while adapting to added or removed strokes.



Figure 8: Qualitative results. **Top-left**: Cross-method comparison on progressive giraffe and zebra sketching; red circles indicate temporal drift and artifacts, while ours remains coherent. **Bottom-left**: Creative comparison with Nano Banana Pro, where ours better preserves the evolving sketch structure and intended concept. **Right**: Fine-grained progressive edits to the giraffe ear, mane, and scenery; ours performs localized adaptation while maintaining global consistency.

incrementally develop, revise, and refine an image while preserving the context established earlier.

5.3 User study

We perform two user studies to compare outputs across preference, temporal coherency, generation quality, erasure, and prompt editing where ProgressNet is consistently preferred over all baselines (Table 3). StableFlow is the strongest competitor on coherency and fine-drawing, while ProgressNet leads every reported criterion, with its largest share on erasure.

Table 3: User study preferences.

	Study 1: Overall			Study 2: Operation-based					
	Pref.	Coher.	Qual.	Erase	Edit	Prog.	F-Draw	F-Erase	Preserv.
S-a-S	2.9	4.9	3.4	1.4	1.9	3.0	5.1	4.2	3.8
SDXS	4.1	13.2	4.3	1.8	2.3	4.2	7.1	5.8	5.1
CB	5.1	10.1	7.2	2.1	4.1	12.1	7.6	7.2	7.4
Flux	5.9	6.8	10.2	2.8	5.3	7.6	8.1	7.8	8.2
SF	10.2	20.3	15.1	6.7	8.8	18.3	22.4	19.7	17.6
Ours	**71.8**	**44.7**	**59.8**	**85.2**	**77.6**	**54.8**	**49.7**	**55.3**	**57.9**

Bold: best, underline: second best.

5.4 Ablation Studies

We evaluate cumulative additions to the baseline (Table 4), interpreting aggregate metrics together with layer-selection comparisons (Figs. 5 and 10) and the with/without-BAC examples (Fig. 9).

Table 4: Ablation on FS-COCO across sketch completion levels. Each row cumulatively enables PCM, LS-KVI, and BAC. **Bold**: best; underline: second best.

				50% Completion					100% Completion						
Variant	PCM	KV	BAC	FID-I ↓	FID-C ↓	DINOv2 ↑	CLIP ↑	T5 ↑	FID-I ↓	FID-C ↓	DINOv2 ↑	CLIP ↑	T5 ↑	Latency ↓	
Baseline	✗	✗	✗	44.7	0.188	0.386	0.285	0.642	67.9	0.264	0.361	0.265	0.641	**0.761**	
+ PCM	✓	✗	✗	**33.9**	0.145	**0.420**	0.304	0.654	**36.6**	0.150	0.414	0.301	0.653	0.816	**+0.055**
+ LS-KVI	✓	✓	✗	35.7	0.161	0.415	0.303	0.654	37.9	0.162	**0.415**	0.300	0.653	0.816	**+0.055**
Ours	✓	✓	✓	36.0	**0.141**	0.415	**0.305**	**0.656**	37.3	**0.143**	0.413	**0.303**	**0.655**	0.829	**+0.068**

(i) Importance of PCM. PCM supplies the dominant improvement in reference and joint fidelity. At full completion, FID-I decreases substantially from 67.9 to 36.6, while DINOv2 increases from 0.361 to 0.414. It establishes the appearance memory on top of which the remaining components are evaluated.

(ii) Impact of Layer-Selective K/V Injection. Adding LS-KVI to PCM maintains comparable DINOv2 and text alignment, with a trade-off in FID-I and FID-C. Separately, Figs. 5 and 10 show that the selected non-vital-layer configuration better preserves appearance than the vital-layer configuration in the illustrated sequences. For stroke removal, Table 8 validates LS-KVI state management: clearing stale K/V enables substantially stronger recovery toward the pre-edit state.



Figure 9: Without BAC, sketch-edge artifacts leak under prompt and sketch conflict or imperfect sketches from Pixelor (Bhunia et al., 2020) and human drawn sketches. BAC reduces visible artifacts associated with these difficult sketch-prompt configurations.

(iii) Impact of Banded Adaptive Control. Adding BAC yields modest quantitative gains in FID-C and CLIP/T5 across later completion stages. At full completion, FID-C improves from 0.162 to 0.143, while CLIP/T5 increase from 0.300/0.653 to 0.303/0.655. These metrics may not fully capture its main effect: reducing visible sketch-edge artifacts under conflicting or imperfect strokes by adapting the strength of appearance-memory conditioning. Accordingly, Fig. 9 shows this correction directly, supporting an essential role for adaptive control beyond uniform metric improvement.

6 Conclusion and Limitations

We proposed ProgressNet, a training-free framework for progressive generation that stabilizes interactions between evolving strokes and prompts. PCM and LS-KVI preserve temporal consistency, while BAC uses retrospective memory-feature divergence to adapt memory. ProgressNet enables progressive drawing by improving fidelity and alignment across completion levels with near-real-time performance. Our work opens avenues for more modality-agnostic progressive generation.

Limitations. *(i)* ProgressNet is primarily evaluated with sketch-based control; extending its memory and adaptive-control mechanisms to other modalities, such as depth and pose, remains unexplored. *(ii)* BAC uses predefined layer bands and modulation bounds; automatically identifying control-sensitive regions on different conditioning modalities is a promising future direction.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and improving readability. All scientific content, experimental design, analyses, results, figures, and conclusions were produced, verified, and finalized by the authors.

REFERENCES

Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-Image Attention for Zero-Shot Appearance Transfer. In *ACM SIGGRAPH*, 2024.

Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended Latent Diffusion. *ACM TOG*, 2023.

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 *CVPR*, 2025.

Hmrishav Bandyopadhyay, Rahim Entezari, Jim Scott, Reshinth Adithyan, Yi-Zhe Song, and Varun Jampani. SD3. 5-Flash: Distribution-Guided Distillation of Generative Flows. *arXiv preprint arXiv:2509.21318*, 2025.

Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. MultiDiffusion: Fusing Diffusion Paths for Controlled Image Generation. In *ICML*, 2023.

Ayan Kumar Bhunia, Ayan Das, Umar Riaz Muhammad, Yongxin Yang, Timothy M Hospedales, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Pixelor: A Competitive Sketching AI Agent. So You Think You Can Sketch? *ACM TOG*, 2020.

Black Forest Labs. FLUX. GitHub repository, 2024.

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. *arXiv preprint arXiv:2506.15742*, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. InstructPix2Pix: Learning to Follow Image Editing Instructions. In *CVPR*, 2023.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. MasaCtrl: Tuning-Free Mutual Self-Attention Control for Consistent Image Synthesis and Editing. In *ICCV*, 2023.

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. In *ACM SIGGRAPH*, 2023.

Dar-Yen Chen, Hmrishav Bandyopadhyay, Kai Zou, and Yi-Zhe Song. NitroFusion: High-Fidelity Single-Step Diffusion through Dynamic Adversarial Training. In *CVPR*, 2025a.

Haoyang Chen, Dongfang Sun, Caoyuan Ma, Shiqin Wang, Kewei Zhang, Zheng Wang, and Zhixiang Wang. Subjective camera 1.0: Bridging human cognition and visual reconstruction through sequence-aware sketch-guided diffusion. In *ICCV*, 2025b.

Pinaki Nath Chowdhury, Aneeshan Sain, Ayan Kumar Bhunia, Tao Xiang, Yulia Gryaditskaya, and Yi-Zhe Song. Fs-COCO: Towards Understanding of Freehand Sketches of Common Objects in Context. In *ECCV*, 2022.

Nadav Z Cohen, Oron Nir, and Ariel Shamir. Conditional Balance: Improving Multi-Conditioning Trade-Offs in Image Generation. In *CVPR*, 2025.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based Semantic Image Editing with Mask Guidance. In *ICLR*, 2023.

Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. FluxSpace: Disentangled Semantic Editing in Rectified Flow Models. In *CVPR*, 2025.

Mathias Eitz, James Hays, and Marc Alexa. How Do Humans Sketch Objects? *ACM TOG*, 2012.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *ICML*, 2024.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis. In *ICLR*, 2023.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion. In *ICLR*, 2023.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control. In *ICLR*, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*, 2017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-Based Real Image Editing with Diffusion Models. In *CVPR*, 2023.

Jimyeong Kim, Jungwon Park, Yeji Song, Nojun Kwak, and Wonjong Rhee. ReFlex: Text-Guided Editing of Real Images in Rectified Flow via Mid-Step Feature Extraction and Attention Adaptation. In *ICCV*, 2025.

Akio Kodaira, Chenfeng Xu, Toshiki Hazama, Takanori Yoshimoto, Kohei Ohno, Shogo Mitsuhori, Soichi Sugano, Hanying Cho, Zhijian Liu, Masayoshi Tomizuka, et al. Streamdiffusion: A pipeline-level solution for real-time interactive generation. In *ICCV*, 2025.

Subhadeep Koley, Ayan Kumar Bhunia, Deeptanshu Sekhri, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. It's All About Your Sketch: Democratising Sketch Control in Diffusion Models. In *CVPR*, 2024.

Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-Concept Customization of Text-to-Image Diffusion. In *CVPR*, 2023.

Aleyna Kütük and Tevfik Metin Sezgin. Class-agnostic visio-temporal scene sketch semantic segmentation. In *WACV*, 2025.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The Role of ImageNet Classes in Fréchet Inception Distance. In *ICLR*, 2023.

Jaerin Lee, Daniel Sungho Jung, Kanggeon Lee, and Kyoung Mu Lee. SemanticDraw: Towards Real-Time Interactive Content Creation from Image Diffusion Models. In *CVPR*, 2025.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, 2023a.

Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *WACV*, 2019.

Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. ControlNet++: Improving Conditional Controls with Efficient Consistency Feedback. In *ECCV*, 2024.

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 *CVPR*, 2023b.

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. In *CVPR*, 2026.

Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-Adapter: An Efficient and Versatile Framework for Adapting Diverse Controls to Any Diffusion Model. In *ICLR*, 2025.

Kuan Heng Lin, Sicheng Mo, Ben Klingher, Fangzhou Mu, and Bolei Zhou. Ctrl-X: Controlling Structure and Appearance for Text-To-Image Generation Without Guidance. In *NeurIPS*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *ICLR*, 2023.

Feng-Lin Liu, Hongbo Fu, Xintao Wang, Weicai Ye, Pengfei Wan, Di Zhang, and Lin Gao. SketchVideo: Sketch-based Video Generation and Editing. In *CVPR*, 2025.

Xiaoyu Liu, Yuxiang Wei, Ming Liu, Xianhui Lin, Peiran Ren, Xuansong Xie, and Wangmeng Zuo. SmartControl: Enhancing ControlNet for Handling Rough Visual Conditions. In *ECCV*, 2024.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting Using Denoising Diffusion Probabilistic Models. In *CVPR*, 2022.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent Consistency Models: Synthesizing High-Resolution Images with Few-Step Inference. *arXiv preprint arXiv:2310.04378*, 2023.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *ICLR*, 2022.

Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. FreeControl: Training-Free Spatial Control of Any Text-to-Image Diffusion Model with Any Condition. In *CVPR*, 2024.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. NULL-Text Inversion for Editing Real Images Using Guided Diffusion Models. In *CVPR*, 2023.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2I-Adapter: Learning Adapters to Dig Out More Controllable Ability for Text-to-Image Diffusion Models. In *AAAI*, 2024.

OpenAI. Introducing ChatGPT Images 2.5. `https://openai.com/index/introducing-chatgpt-images-2-5/`, 2026. [Accessed 23-09-2026].

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *TMLR*, 2024.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *ICCV*, 2023.

Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. In *CVPR*, 2024.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *ICCV*, 2023.

Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, et al. UniControl: A Unified Diffusion Model for Controllable Visual Generation In the Wild. In *NeurIPS*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models from Natural Language Supervision. In *ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 2020.

Vasco Ramos, Regev Cohen, Idan Szpektor, and Joao Magalhaes. Beyond the Noise: Aligning Prompts with Latent Representations in Diffusion Models. *arXiv preprint arXiv:2512.08505*, 2025.

Royi Rassin, Shauli Ravfogel, and Yoav Goldberg. DALL-E 2 is Seeing Double: Flaws in Word-to-Concept Mapping in Text2Image Models. *arXiv preprint arXiv:2210.10606*, 2022.

Royi Rassin, Eran Hirsch, Daniel Glickman, Shauli Ravfogel, Yoav Goldberg, and Gal Chechik. Linguistic Binding in Diffusion Models: Enhancing Attribute Correspondence through Attention Map Alignment. In *NeurIPS*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *CVPR*, 2023.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM TOG*, 2016.

Vishnu Sarukkai, Lu Yuan, Mia Tang, Maneesh Agrawala, and Kayvon Fatahalian. Block and Detail: Scaffolding Sketch-to-Image Generation. In *ACM UIST*, 2024.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial Diffusion Distillation. *arXiv preprint arXiv:2311.17042*, 2023.

Yuda Song, Zehao Sun, and Xuanwu Yin. SDXS: Real-Time One-Step Latent Diffusion Models with Image Conditions. *arXiv preprint arXiv:2403.16627*, 2024.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-Robust Large Mask Inpainting with Fourier Convolutions. In *WACV*, 2022.

Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. OminiControl: Minimal and Universal Control for Diffusion Transformer. In *ICCV*, 2025.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Türe. What the DAAM: Interpreting stable diffusion using cross attention. In *ACL*, 2023.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation. In *CVPR*, 2023.

Luozhou Wang, Guibao Shen, Wenhang Ge, Guangyong Chen, Yijun Li, and Yingcong Chen. Text-Anchored Score Composition: Tackling Condition Misalignment in Text-to-Image Diffusion Models. In *ECCV*, 2024.

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 *ICCV*, 2025.

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 *ICCV*, 2023.

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. *arXiv preprint arXiv:2606.11751*, 2026.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text Compatible Image Prompt Adapter for Text-to-Image Diffusion Models. *arXiv preprint arXiv:2308.06721*, 2023.

Denis Zavadski, Johann-Friedrich Feiden, and Carsten Rother. ControlNet-XS: Rethinking the Control of Text-to-Image Diffusion Models as Feedback-Control Systems. In *ECCV*, 2024.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *ICCV*, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-ControlNet: All-in-One Control to Text-to-Image Diffusion Models. In *NeurIPS*, 2023.

Zijun Zhou, Yingying Deng, Xiangyu He, Weiming Dong, and Fan Tang. Multi-turn Consistent Image Editing. In *ICCV*, 2025.

A Quantitative Evaluations

A.1 Dataset and Progressive Evaluation Protocol

We evaluate three complementary sketch domains using fixed evaluation subsets and identical inputs for every compared method. FS-COCO (Chowdhury et al., 2022) contributes 10,000 temporally ordered freehand scene sketches with paired photographs and captions. Photo-Sketching (Li et al., 2019) contributes 500 contour-sketch/image pairs from its 5,000 available pairs. Sketchy (Sangkloy et al., 2016) contributes 1,250 single-object freehand sketches, stratified uniformly across all 125 categories with 10 samples per category. The resulting generation-quality evaluation contains 11,750 samples. For each sample, the ordered sketch sequence and associated text condition are fixed once and shared across all methods; progressive states are evaluated at $\{10, 20, 30, 50, 70, 80, 100\}\%$ completion.

Table 5: Datasets used for generation-quality and progressive evaluation. The same sample identities and progressive states are supplied to every method.

Dataset	Eval. N	Sketch domain	Sampling / pairing
FS-COCO	10,000	Freehand scenes	Full evaluation set; paired image/text
Photo-Sketching	500	Contour tracing	Evaluation subset of paired sketch/image data
Sketchy	1,250	Freehand objects	10/category across 125 categories

A.2 Progressive Evaluation Metrics

Let $I_{i,k}$ denote the generated image for sample i at sketch-completion level k. For a progressive transition $a \to b$, we measure consecutive-frame semantic coherence using DINOv2 cosine similarity,

$$D_{a\to b} = \frac{1}{N}\sum_{i=1}^{N} \cos(\phi_{\mathrm{DINO}}(I_{i,a}), \phi_{\mathrm{DINO}}(I_{i,b}))\,, \tag{6}$$

and perceptual change using LPIPS,

$$L_{a\to b} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{LPIPS}(I_{i,a}, I_{i,b}). \tag{7}$$

Quality-adjusted progressive fidelity–coherence. We evaluate temporal coherence jointly with the distributional quality of the corresponding sketch-completion regime. To use the same reported quality anchors for all three datasets, we associate the three representative progressive transitions

$$\tau_{10} = 10\to 20, \qquad \tau_{50} = 30\to 50, \qquad \tau_{100} = 80\to 100 \tag{8}$$

with FID-I measured at the corresponding reported anchors $q \in \mathcal{A} = \{10, 50, 100\}$. Let $\mathrm{FID}_{I,q}$ denote FID-I at anchor q.

We define the DINO-based progressive fidelity–coherence score as

$$\text{PFC-DINO}_q = \frac{100\, D_{\tau_q}}{\mathrm{FID}_{I,q}}, \qquad q \in \mathcal{A}, \tag{9}$$

and its LPIPS counterpart as

$$\text{PFC-LPIPS}_q = \frac{100}{\mathrm{FID}_{I,q}\,(1 + L_{\tau_q})}. \tag{10}$$

Both scores are higher-is-better. The factor 100 only sets a convenient numerical scale and does not affect rankings. These quantities are quality-adjusted composite summaries rather than pure temporal-coherence measures. PFC-DINO combines high semantic inter-turn similarity with strong distributional quality, while PFC-LPIPS analogously combines low perceptual change with the same quality adjustment. Their reported means are

$$\overline{P}_D = \frac{1}{|\mathcal{A}|}\sum_{q\in\mathcal{A}} \text{PFC-DINO}_q, \qquad \overline{P}_L = \frac{1}{|\mathcal{A}|}\sum_{q\in\mathcal{A}} \text{PFC-LPIPS}_q. \tag{11}$$

Cross-completion robustness. PFC evaluates selected consecutive transitions, whereas cross-completion robustness measures whether standard generation metrics remain both strong and stable over the three common completion anchors $\mathcal{A} = \{10, 50, 100\}$. For each metric x, let

$$\mu_x = \frac{1}{|\mathcal{A}|} \sum_{q \in \mathcal{A}} x_q, \qquad \mathrm{CV}_x = \frac{\sigma_x}{\mu_x}, \tag{12}$$

where σ_x is the population standard deviation across the three anchors.

For the two distributional metrics, we first define their relative variation

$$V_{\mathrm{Dist}} = \sqrt{\frac{\mathrm{CV}^2_{\text{FID-I}} + \mathrm{CV}^2_{\text{FID-C}}}{2}}, \tag{13}$$

and combine this with their average quality:

$$Q_{\mathrm{Dist}} = \frac{1}{\sqrt{\mu_{\text{FID-I}}\mu_{\text{FID-C}}}\,(1 + V_{\mathrm{Dist}})}. \tag{14}$$

Visual robustness uses paired DINOv2 fidelity,

$$Q_{\mathrm{Vis}} = \frac{\mu_{\mathrm{DINO}}}{1 + \mathrm{CV}_{\mathrm{DINO}}}, \tag{15}$$

while textual robustness jointly considers CLIPScore and T5-Cos:

$$V_{\mathrm{Text}} = \sqrt{\frac{\mathrm{CV}^2_{\mathrm{CLIP}} + \mathrm{CV}^2_{\mathrm{T5}}}{2}}, \qquad Q_{\mathrm{Text}} = \frac{\sqrt{\mu_{\mathrm{CLIP}}\mu_{\mathrm{T5}}}}{1 + V_{\mathrm{Text}}}. \tag{16}$$

To provide a common and interpretable scale across sketch domains, each robustness axis is normalized to the FLUX baseline evaluated on the same dataset:

$$R_x = 100 \frac{Q_x}{Q_x^{\mathrm{Flux}}}, \qquad x \in \{\mathrm{Dist}, \mathrm{Vis}, \mathrm{Text}\}. \tag{17}$$

Thus, 100 denotes the FLUX baseline and values above 100 indicate stronger quality-adjusted cross-completion robustness. Finally, we summarize the three complementary axes by their geometric mean,

$$R_{\mathrm{Overall}} = \left(R_{\mathrm{Dist}} R_{\mathrm{Vis}} R_{\mathrm{Text}}\right)^{1/3}. \tag{18}$$

All four cross-completion scores are higher-is-better. Unlike PFC, these quantities measure how well standard generation quality and alignment are maintained across completion levels.

A.3 Extended Quantitative Results

Along with the main paper, we additionally record results for quantitative comparison and ablation at intermediate sketch completion levels. Table 6 extends the six-method comparison of main paper Table 1 to the full seven-level sweep $\{10, 20, 30, 50, 70, 80, 100\}\%$, revealing how each method's distributional quality and semantic fidelity evolve as strokes progressively accumulate. Table 7 correspondingly extends the component ablation of main paper Table 4 to the intermediate levels, isolating the individual contribution of each proposed component, PCM, LS-KVI, and BAC, at every stage of sketch completion. Together, these tables show that the gains reported at the three anchor levels $\{10, 50, 100\}\%$ are not an artifact of those selected levels.

On FID-I at 10% completion. At 10% sketch completion, sketches are highly sparse and provide limited structural constraint. ProgressNet (36.5) is therefore marginally above the Flux Baseline (32.9) at this stage. From 20% completion onward, ProgressNet matches or leads on FID-I and FID-C across all completion levels, consistent with effective appearance anchoring as the sketch becomes more informative.

Table 6: Extended comparison at intermediate sketch completion levels on FS-COCO. Results at $\{10, 50, 100\}\%$ are in the main paper (Table 1).

	20% Completion						30% Completion						70% Completion						80% Completion					
Metric	S-a-S (Sarukkai et al., 2024)	SDXS (Song et al., 2024)	CB (Cohen et al., 2025)	Flux (Black Forest Labs, 2024)	SF (Avrahami et al., 2025)	**Ours**	S-a-S (Sarukkai et al., 2024)	SDXS (Song et al., 2024)	CB (Cohen et al., 2025)	Flux (Black Forest Labs, 2024)	SF (Avrahami et al., 2025)	**Ours**	S-a-S (Sarukkai et al., 2024)	SDXS (Song et al., 2024)	CB (Cohen et al., 2025)	Flux (Black Forest Labs, 2024)	SF (Avrahami et al., 2025)	**Ours**	S-a-S (Sarukkai et al., 2024)	SDXS (Song et al., 2024)	CB (Cohen et al., 2025)	Flux (Black Forest Labs, 2024)	SF (Avrahami et al., 2025)	**Ours**
FID-I ↓	195.1	39.8	198.4	37.0	36.5	**35.8**	202.9	40.2	193.4	38.5	38.3	**36.4**	210.7	44.0	203.4	52.5	43.6	**36.1**	212.6	45.7	208.5	58.0	47.6	**36.7**
FID-C ↓	0.442	0.163	0.406	0.163	0.190	**0.158**	0.433	0.165	0.396	0.166	0.198	**0.144**	0.465	0.177	0.398	0.212	0.203	**0.142**	0.470	0.180	0.421	0.229	0.209	**0.141**
DINOv2 ↑	0.288	0.333	0.331	0.379	0.392	**0.402**	0.281	0.336	0.349	0.383	0.387	**0.409**	0.290	0.335	0.319	0.381	0.378	**0.415**	0.289	0.334	0.313	0.375	0.370	**0.415**
CLIPScore ↑	0.256	0.299	0.292	0.294	0.300	**0.304**	0.266	0.298	0.291	0.291	0.297	**0.305**	0.266	0.295	0.291	0.278	0.294	**0.304**	0.252	0.294	0.289	0.274	0.291	**0.304**
T5-Cos ↑	0.630	0.648	0.644	0.648	0.656	**0.658**	0.623	0.648	0.639	0.646	0.656	**0.657**	0.629	0.646	0.644	0.642	0.654	**0.656**	0.629	0.646	0.630	0.641	0.654	**0.655**

Bold: best, underline: second best.

Table 7: Extended ablation at intermediate completion levels on FS-COCO. Results at $\{50, 100\}\%$ are in the main paper (Table 4).

				10% Completion					20% Completion					30% Completion					70% Completion					80% Completion				
Variant	PCM	KV	BAC	FID-I↓	FID-C↓	DINO↑	CLIP↑	T5↑	FID-I↓	FID-C↓	DINO↑	CLIP↑	T5↑	FID-I↓	FID-C↓	DINO↑	CLIP↑	T5↑	FID-I↓	FID-C↓	DINO↑	CLIP↑	T5↑	FID-I↓	FID-C↓	DINO↑	CLIP↑	T5↑
Baseline	✗	✗	✗	**32.9**	**0.149**	0.375	0.298	0.651	37.0	0.163	0.379	0.294	0.648	38.5	0.166	0.383	0.291	0.646	52.5	0.212	0.381	0.278	0.642	58.0	0.229	0.375	0.274	0.641
+ PCM	✓	✗	✗	36.5	0.152	**0.400**	**0.306**	**0.657**	**35.0**	**0.148**	**0.407**	**0.306**	0.656	**34.3**	0.147	**0.413**	**0.305**	0.656	**34.5**	0.146	**0.422**	0.303	0.654	**34.7**	0.147	**0.421**	0.303	0.653
+ LS-KVI	✓	✓	✗	36.5	0.152	**0.400**	**0.306**	**0.657**	37.0	0.163	0.402	0.304	0.656	36.4	0.163	0.408	0.303	0.655	36.2	0.159	0.419	0.302	0.653	36.5	0.160	0.418	0.301	0.653
Full (+ BAC)	✓	✓	✓	36.5	0.152	**0.400**	**0.306**	**0.657**	35.8	0.158	0.402	0.304	**0.658**	36.4	**0.144**	0.409	**0.305**	**0.657**	36.1	**0.142**	0.415	**0.304**	**0.656**	36.7	**0.141**	0.415	**0.304**	**0.655**

Bold: best, underline: second best.

On T5 alignment at early completion. At 10% completion, StableFlow marginally leads on T5-Cos (0.659 vs. 0.657); the two methods are tied at 50% (0.656 each), with ProgressNet leading at full completion (0.655 vs. 0.654). Moreover, ProgressNet leads on every intermediate completion level as presented in Table 6. At the very sparse 10% stage, text-alignment measurements are more sensitive to the under-specified visual condition. From 20% completion onward, where the sketch provides stronger structural evidence, ProgressNet consistently leads or matches on the reported text-alignment metrics.

A.4 Interpreting the Ablation Progression

FID-I and FID-C measure distributional quality in Inception and CLIP image-feature spaces, respectively; DINOv2 measures paired visual-semantic fidelity, while CLIPScore and T5-Cos assess text alignment. We therefore interpret temporal and fidelity metrics jointly, complemented by qualitative and operation-level evaluations. From 30% completion onward on FS-COCO, the full system (PCM + LS-KVI + BAC) obtains the strongest FID-C and remains at or above the other ablation variants on CLIPScore and T5-Cos across the reported stages (Tables 4 and 7). Its FID-C remains low and stable across 30–100% completion (0.144, 0.141, 0.142, 0.141, 0.143), while CLIPScore and T5-Cos remain competitive throughout. Together, the full system maintains strong FID-I while improving FID-C and textual alignment across later completion stages; the qualitative BAC analysis separately examines difficult sketch–prompt configurations.

A.5 Inference and Reproducibility Details

Inference configuration. All ProgressNet experiments use FLUX.1-schnell with Union ControlNet at 1024×1024 resolution. Unless otherwise stated, the structural and memory-conditioning scales are fixed to $\lambda_s = 0.5$ and $\lambda_m = 0.65$, respectively. Live progressive generation uses two inference steps in bfloat16 on a single NVIDIA H200 GPU.

Sketch preprocessing. The line-based sketch condition is converted to a binary structural map before being resized to the 1024×1024 ControlNet input resolution. We convert the sketch to grayscale, threshold it at intensity 200, invert the binary map so that drawn strokes form the active condition, and resize using nearest-neighbour interpolation.



Figure 10: Extended qualitative comparison of vital-layer vs. non-vital-layer K/V injection across four FS-COCO samples at progressive sketch completions (20%, 50%, 80%, 100%). Vital-layer ((Avrahami et al., 2025)) injection progressively corrupts appearance, shifting illumination (clock tower darkens), altering identity (horse rider changes clothing and pose), and inducing color drift (banana flower saturates), while non-vital-layer ($\{5, 6, 7, 10, 13, 14\}$) injection preserves consistent appearance, lighting, and identity throughout the progression. This extends the single-sample comparison in Fig. 5 of the main paper.

B Component Implementation and LS-KVI Analysis

B.1 Previous-Concept Memory implementation

PCM derives its commitment mask C_t from the processed current sketch. We combine a dilated stroke-proximity support with flood-filled enclosed regions; the resulting mask marks content that should remain committed under subsequent interaction. At the two extremes, $C_t{=}1$ denotes fully committed content retained from I_{t-1}, whereas $C_t{=}0$ denotes fully relaxed regions during a prompt update; intermediate values provide a soft blend between the previous output and the neutral prior. For stroke addition, the previous output is used unchanged as memory. For stroke removal, evidence inside the erased region is withdrawn by neutral white filling before tile conditioning. For prompt updates, committed content is preserved while uncommitted regions are replaced by the neutral gray prior, as defined in Eq. (1) of the main paper. The same mask construction is used for all datasets and all interaction sequences.

Commitment-mask construction. The soft commitment mask combines two complementary supports. For stroke-proximity support, pixels with grayscale intensity below 200 are treated as

sketch pixels, dilated by 24 pixels using an elliptical kernel, and Gaussian-smoothed with a 41×41 kernel. For enclosed regions, we first morphologically close sketch gaps using a 17×17 elliptical kernel, corresponding to an 8-pixel closing radius. We then flood-fill the exterior from the image boundary and retain the non-exterior region as the enclosed support, followed by Gaussian smoothing with a 21×21 kernel. The final commitment mask is the pixelwise maximum of the proximity and enclosed-region supports and is clipped to $[0, 1]$. These parameters are held fixed throughout all reported experiments.

B.2 Choice of vital and non-vital layers

The mechanism is made visible in Fig. 10, which extends the single-sample comparison of Fig. 5 across four diverse FS-COCO scenes. In each case, vital-layer injection progressively overrides appearance attributes that should remain stable: the clock tower's warm sunset illumination darkens into an unnatural palette; the horse rider's clothing, skin tone, and pose drift across completions; and the banana flower saturates in color as strokes accumulate.

In contrast, non-vital-layer injection (bottom rows) maintains consistent lighting, identity, and color across the four examples. Empirically, injecting at these lower-vitality layers preserves appearance more effectively while allowing structural adaptation. The giraffe retains its pose and forest backdrop; the clock tower preserves warm tones and adapts precise details correctly; the horse rider maintains consistent attire and posture; and the banana flower evolves structurally without color shift. This is precisely the property exploited by LS-KVI under progression: temporal smoothing at layers whose individual absence leaves generation quality intact (Section 4.2, main paper), enabling the concept to structurally evolve with new strokes while preserving what the image *was*. We therefore inject at non-vital layers throughout. Under the FLUX vitality analysis, the six lowest-vitality layers selected for injection are $\{5, 6, 7, 10, 13, 14\}$, with α_{kv}=0.3 in all reported experiments. The cache is refreshed after each generation and cleared on stroke removal so that correspondence pathways associated with erased content are not reintroduced.

K/V tensor and cache implementation. LS-KVI attaches forward hooks to the key and value projections of joint attention at the selected transformer blocks. The full projected key and value tensors are retained rather than spatially or channel-wise subsampled. For a selected layer ℓ, the current turn uses

$$\mathbf{K}'^{t}_{\ell} = \mathbf{K}^{t}_{\ell} + \alpha_{\text{kv}}\mathbf{K}^{t-1}_{\ell}, \qquad \mathbf{V}'^{t}_{\ell} = \mathbf{V}^{t}_{\ell} + \alpha_{\text{kv}}\mathbf{V}^{t-1}_{\ell},$$

with $\alpha_{\text{kv}} = 0.3$. During generation, the current projection outputs are captured for the next turn; after the generation finishes, the captured tensors are committed as the new reference cache. The cache is indexed by transformer layer and is refreshed after every completed turn. On stroke removal, the cached K/V state is cleared before the subsequent generation so that features associated with erased content cannot be re-injected.

Layer-selection protocol. The injection subset is selected once using the residual-bypass vitality analysis described in Section 4.2: each candidate transformer layer is bypassed individually over progressive sketch stages and ranked by the resulting output deviation. LS-KVI uses the lowest-vitality layers and keeps this subset and $\alpha_{\text{kv}} = 0.3$ fixed across the reported datasets rather than re-selecting layers per dataset.

Table 8: Erasure-specific LS-KVI ablation on 100 FS-COCO sequences.

LS-KVI state	Return LPIPS ↓	Recovery ↑	DINOv2 ↑	Erased-region L1 ↓	Outside-change L1 ↓
Retain previous K/V	0.295	0.019	0.854	0.121	**0.096**
Invalidate previous K/V	**0.204**	**0.303**	**0.885**	**0.085**	0.123

Erasure-specific LS-KVI state invalidation. We quantitatively evaluate stroke removal on 100 FS-COCO sequences using a controlled $S_{50} \to S_{70} \to S_{50}$ add–erase cycle, where the final sketch is identical to the initial 50% state. All generation settings and preceding history are fixed; we vary only whether the previous LS-KVI cache is retained or invalidated during removal. Return LPIPS (Zhang et al., 2018) and DINOv2 (Oquab et al., 2024) measure recovery toward the original I_{50} output, while Recovery denotes the LPIPS improvement relative to the preceding $I_{50} \to I_{70}$

change. We additionally report error inside the erased region and appearance change outside it. Invalidating stale LS-KVI state reduces return LPIPS by 30.9% and improves recovery in $93/100$ sequences (paired Wilcoxon $p = 7.1\times10^{-15}$). It also reduces erased-region error in $91/100$ cases, showing that stale temporal correspondences can resist destructive updates.

C Empirical Validation of Banded Adaptive Control

We examine how control responses vary with network depth, solver step, and progressive stage, and whether related patterns persist across backbones and sketch domains. These diagnostics motivate the banded, step-specific design; the component ablations evaluate its effects on generated outputs.

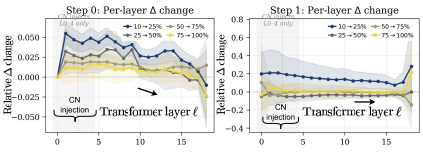


Figure 11: Layerwise relative change in the measured control-induced perturbation across progressive sketch transitions at solver steps 0 and 1. The response varies systematically with backbone depth and solver step.

Implemented BAC comparison. For each ControlNet block k and solver step s, BAC compares the cached memory-conditioning features from the two preceding turns. We compute channel-wise Jensen–Shannon divergence (JSD) between their activation distributions and average the resulting divergences within the early and late ControlNet bands. Following Eq. (4) of the main paper, these band-wise values are clipped to $[f_{\min}, f_{\max}]$ to obtain the four step- and band-specific BAC correction factors.

C.1 Layer- and Step-Dependent Control Responses

Fig. 11 reports layerwise changes in the measured control-induced perturbation across progressive stages. The profiles differ across backbone depth and solver step. In standard Diffusers FLUX, ControlNet residual samples are distributed across backbone blocks rather than injected only into blocks $\ell_{0\text{-}4}$. The resulting profiles provide evidence of depth- and step-dependent control responses across progressive stages.

For FLUX's five ControlNet blocks, BAC uses $\mathcal{K}_{\text{early}}=\{0,1,2\}$ and $\mathcal{K}_{\text{late}}=\{3,4\}$, following the early/late partition used in the perturbation analysis. The implemented BAC factor follows the retrospective cached-feature comparison defined above. JSD_{cw} is computed independently at each ControlNet block and solver step, after which block-wise divergences are aggregated within the early and late bands to obtain the correction factors. No additional forward pass is required during generation.

BAC Correction-Factor Variants. We compare four divergence measures for computing the BAC correction factor: cosine similarity, KL divergence, Jensen–Shannon divergence (JSD), and channel-wise JSD. Fig. 12 presents qualitative results on an out-of-distribution sketch from a stroke-based drawing agent, Pixelor (Bhunia et al., 2020). Without any correction (W/o BAC), the generation retains a solid background with visible sketch-edge artifacts. KL and standard JSD show stronger visible distortions at higher completion levels. Cosine similarity yields cleaner outputs but still transfers fine sketch details into the generated image.

C.2 Cross-backbone validation on SD3.5 Large Turbo

To test whether the band- and step-asymmetric control behavior motivating BAC is specific to FLUX.1-schnell, we repeat the diagnostic on SD3.5 Large Turbo using 3,000 FS-COCO samples. SD3.5 uses an MMDiT backbone, four solver steps, and separate Canny and blur ControlNets, providing a substantially different control architecture.

Late-layer redistribution remains stronger on SD3.5 despite the different architecture and the use of separate control branches. The SD3.5 comparison was recorded at the five transitions shown in Table 9; the additional $70\rightarrow80\%$ transition is included in the FLUX cross-dataset analysis below. The inferred band boundary is also modality-dependent: the Canny branch separates at layer 11/19 ($\sim 58\%$ depth), close to FLUX's 3/5 (60%) split, whereas the blur-memory branch separates earlier at layer 5/19 ($\sim 26\%$). This distinction cannot be observed in FLUX's Union ControlNet because structure and memory share one control stack.

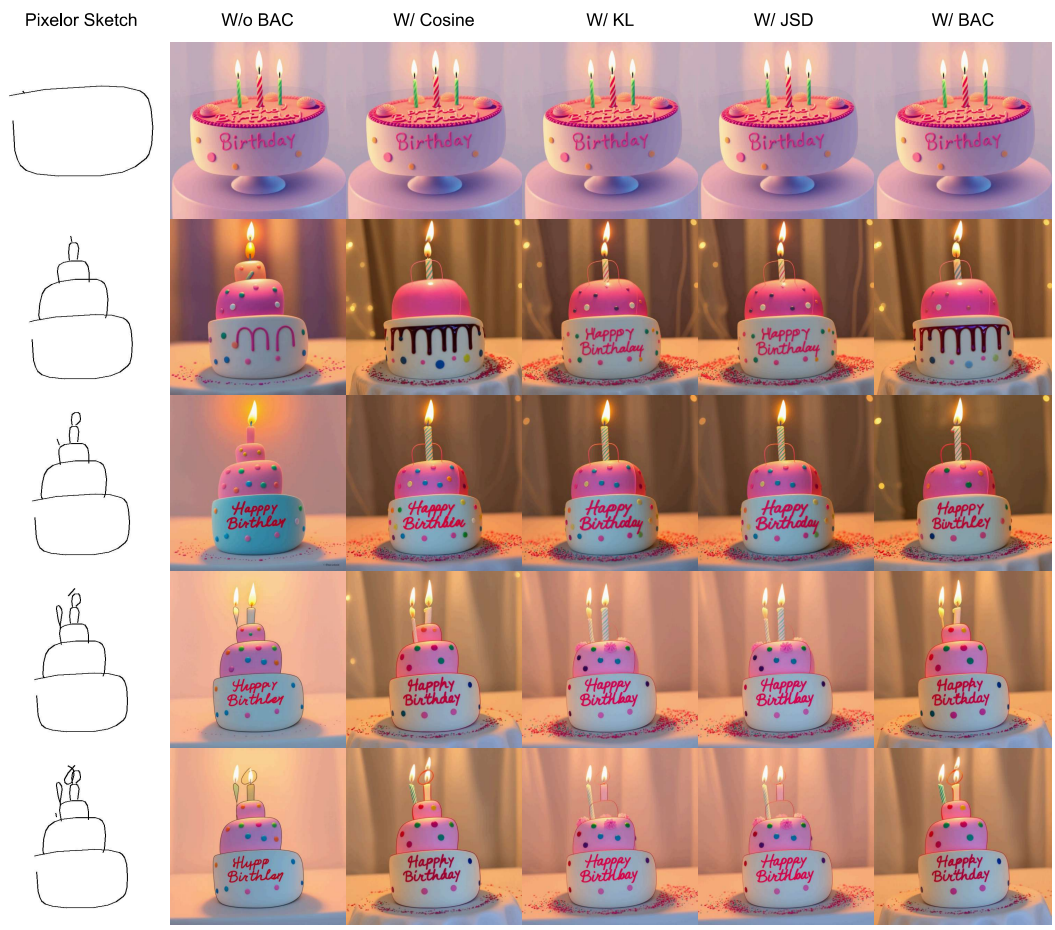


Figure 12: Comparison of BAC correction-factor variants on an out-of-distribution sketch from Pixelor (Bhunia et al., 2020). We evaluate cosine similarity, KL divergence, Jensen–Shannon divergence (JSD), and channel-wise JSD. The first column (W/o BAC) applies no correction factor. KL and JSD introduce visible artifacts at full completion, while disabling correction altogether retains a solid background with dark sketching artifacts. Compared to cosine similarity, our channel-wise JSD comparison produces smoother modulation without imprinting strong sketch-edge details into the generated image.

Table 9: Late/early JSD ratio across FLUX and SD3.5. Ratios above one indicate stronger redistribution in later ControlNet layers.

Transition	FLUX (5 layers)	SD3.5 Canny (19)	SD3.5 Blur (19)
10→20%	3.2×	3.3×	2.7×
20→30%	2.0×	2.0×	2.7×
30→50%	2.2×	2.2×	2.3×
50→70%	2.3×	2.0×	2.4×
80→100%	2.2×	1.8×	2.6×

Table 10: Step asymmetry at the 10→20% transition. SD3.5 exhibits progressive amplification across its four solver steps.

Control branch	Step 0	Step 1	Step 2	Step 3	Last/First	Last higher
SD3.5 Canny	0.005	0.007	0.009	0.021	3.97×	99.3%
SD3.5 Blur	0.118	0.147	0.168	0.212	1.79×	100%
FLUX (2-step)	0.134	0.206	–	–	1.54×	97.0%

Table 11: Late/early JSD ratio at solver step 0 across four sketch domains. The same band asymmetry persists at every evaluated transition.

Dataset	10→20	20→30	30→50	50→70	70→80	80→100
FS-COCO	3.2×	2.0×	2.2×	2.3×	2.3×	2.2×
FrISS	3.4×	2.0×	2.1×	2.1×	2.1×	2.1×
Photo-Sketching	3.2×	2.0×	2.1×	2.1×	2.1×	2.2×
Sketchy	3.3×	1.8×	2.2×	2.0×	1.9×	2.0×

Table 12: Step-1/step-0 JSD amplification at the 10→20% transition. The existence of step asymmetry is stable across domains, while its magnitude remains content-dependent.

Dataset	Step 1 / Step 0	Samples with step 1 higher
FS-COCO	1.54×	97.0%
FrISS	1.59×	97.7%
Photo-Sketching	1.44×	93.8%
Sketchy	2.15×	96.3%

Table 13: Transition ranking by mean JSD across datasets (rank 1 indicates the largest redistribution). The ordering is preserved across all four domains.

Rank	FS-COCO	FrISS	Photo-Sketching	Sketchy
1	20→30 (.362)	20→30 (.365)	20→30 (.371)	20→30 (.390)
2	80→100 (.247)	80→100 (.241)	80→100 (.241)	80→100 (.256)
3	30→50 (.238)	30→50 (.228)	30→50 (.232)	30→50 (.256)
4	50→70 (.225)	50→70 (.217)	50→70 (.214)	50→70 (.234)
5	70→80 (.193)	70→80 (.203)	70→80 (.198)	70→80 (.212)
6	10→20 (.134)	10→20 (.118)	10→20 (.128)	10→20 (.104)

C.3 Cross-dataset BAC validation

We next test whether the same redistribution pattern depends on FS-COCO. The BAC diagnostic is run on FLUX.1-schnell over four sketch domains totaling 52,368 samples: FS-COCO (Chowdhury et al., 2022) (10,000), FrISS (Kütük & Sezgin, 2025) (1,010), Photo-Sketching (Li et al., 2019) (5,000), and Sketchy (Sangkloy et al., 2016) (36,358). These datasets span freehand scenes, abstract text-associated scenes, edge-aligned contour tracings, and single-object freehand drawings.

Together, the SD3.5 and cross-dataset analyses show that the qualitative BAC signature used in the main paper, stronger late-band redistribution and solver-step asymmetry—persists beyond one backbone and one sketch domain.

D User Study Details

We recruited participants from open communities through an online Gradio interface. Both studies were conducted double-blind: method identities were hidden, presentation order was randomized per trial, and participants were not told which system produced an output.

Study 1: overall generation quality. Participants evaluated six methods (Ours, Flux+ControlNet (Black Forest Labs, 2024), SDXS (Song et al., 2024), Sketch-a-Sketch (Sarukkai et al., 2024), StableFlow (Avrahami et al., 2025), and Conditional Balance (Cohen et al., 2025)) across 10 diverse samples. For each sample, participants (i) ranked the six methods by overall visual preference, (ii) rated temporal coherency using a continuous 0–1 slider while viewing progressive-generation videos, and (iii) rated generation quality on a five-point Likert scale from Very Poor to Excellent. These correspond to the *Pref.*, *Coher.*, and *Qual.* columns in Table 3. The reported values aggregate responses across participants and evaluated samples and are presented as normalized criterion-wise scores, with higher values indicating stronger user preference.

Study 2: operation-level capabilities. Participants first selected the better output for 12 erasure samples and three editing-pipeline samples, evaluating stroke-removal and sketch-modification fi-

delity. They then viewed side-by-side videos of complete progressive sessions and judged prompt-guided progression, fine-grained drawing quality, fine-grained erasure quality, and content preservation across edits. These map to the *Erase*, *Edit*, *Prog.*, *F-Draw*, *F-Erase*, and *Preserv.* columns in Table 3. As in Study 1, responses are aggregated across participants and evaluated samples and reported as normalized criterion-wise scores, with higher values indicating stronger outcomes. Scores are normalized independently within each criterion across methods.

E Extended Related Work

Controllable Image Generation. ControlNet (Zhang et al., 2023) established spatial conditioning through auxiliary branches attached to frozen diffusion backbones, and subsequent works expanded controllability through unified multi-condition models (Zhao et al., 2023; Qin et al., 2023), lightweight conditioning adapters (Mou et al., 2024; Ye et al., 2023), feedback-based control architectures (Li et al., 2024; Zavadski et al., 2024), grounded and regional generation (Li et al., 2023b; Xie et al., 2023; Bar-Tal et al., 2023), and training-free feature manipulation (Mo et al., 2024; Tumanyan et al., 2023). Sketch-based conditioning has similarly progressed through dedicated sketch-to-image and interactive systems (Sarukkai et al., 2024; Song et al., 2024) toward increasingly general conditional generators. These developments largely assume a condition that is already sufficiently specified before generation begins.

The transition from UNet diffusion to DiT and rectified-flow architectures (Peebles & Xie, 2023; Esser et al., 2024; Black Forest Labs, 2024) has further changed how spatial conditions are injected. Ctrl-Adapter (Lin et al., 2025) bridges pretrained ControlNets to newer diffusion backbones, OminiControl (Tan et al., 2025) introduces lightweight universal control for FLUX, and Semantic-Draw (Lee et al., 2025) supports real-time multi-region interaction. ControlNet-XS (Zavadski et al., 2024) and ControlNet++ (Li et al., 2024) likewise revisit the interaction between the conditioning pathway and the generator. While these methods considerably expand *what* can be controlled, their conditioning remains predominantly generation-centric rather than progression-centric: the model receives the current condition, but does not explicitly reason about what should remain stable from the preceding interaction turn. ProgressNet instead treats the previous concept as part of the state of generation and explicitly distinguishes stroke addition, removal, and prompt reinterpretation.

Real-Time and Iterative Image Editing. Diffusion editing has been extensively studied through inversion and attention manipulation (Hertz et al., 2023; Meng et al., 2022; Couairon et al., 2023; Kawar et al., 2023; Mokady et al., 2023), inpainting (Avrahami et al., 2023; Lugmayr et al., 2022; Suvorov et al., 2022), instruction-based editing (Brooks et al., 2023), and personalization (Hu et al., 2022; Ruiz et al., 2023; Gal et al., 2023; Kumari et al., 2023). Such methods enable semantic or regional modification but generally solve each edit as a fresh optimization, inversion, or generation problem rather than maintaining a persistent state across a long interaction sequence.

Few-step and distilled generation substantially reduce this cost. Latent Consistency Models (Luo et al., 2023), SDXS (Song et al., 2024), NitroFusion (Chen et al., 2025a), SDXL-Turbo (Sauer et al., 2023), and StreamDiffusion (Kodaira et al., 2025) push inference toward interactive rates. However, low latency alone does not imply temporal consistency: regenerating from a changed sketch can alter previously established identity, appearance, geometry, or background even when the newly introduced stroke is local. Among interactive sketch systems, Block-and-Detail (Sarukkai et al., 2024) stabilizes generation through coarse-to-fine re-noising, whereas SDXS (Song et al., 2024) provides one-step image-conditioned synthesis; both remain vulnerable to drift when the conditioning evolves progressively.

Recent training-free editing methods are particularly related to our temporal component. StableFlow (Avrahami et al., 2025) identifies vital transformer layers and reuses internal features to preserve content during editing, while FluxSpace (Dalva et al., 2025) exposes semantic directions in rectified-flow representations and FLUX.1 Kontext (Black Forest Labs et al., 2025) enables iterative in-context editing. Related feature-reuse mechanisms have also been explored for consistency through cross-image or temporal attention (Alaluf et al., 2024; Cao et al., 2023; Qi et al., 2023). These works demonstrate that internal attention features provide a strong substrate for consistency, but progressive sketching requires a different balance: preserving all influential layers too strongly can suppress newly drawn structure. This motivates our Layer-Selective K/V Injection, which pre-

serves temporal information only through non-vital layers while leaving structurally important layers free to adapt.

Condition Misalignment and Adaptive Control. A separate literature studies the failure of text and visual conditions to remain semantically aligned. Text-to-image models exhibit semantic neglect, incorrect attribute binding, and word-to-concept mismatch (Rassin et al., 2022; Wang et al., 2024), with such failures analyzed through cross-attention (Tang et al., 2023) and latent-space representations (Ramos et al., 2025). Structured Diffusion (Feng et al., 2023) decomposes prompts for improved compositionality, while Attend-and-Excite (Chefer et al., 2023), Linguistic Binding (Rassin et al., 2023), and Attention Refocusing (Phung et al., 2024) intervene on attention to improve semantic correspondence.

More directly related to ProgressNet, SmartControl (Liu et al., 2024) adapts ControlNet behavior under rough or unreliable visual conditions, while Text-Anchored Score Composition (Wang et al., 2024) explicitly addresses conflict between textual and spatial conditions. Conditional Balance (Cohen et al., 2025) controls competing conditioning signals through selective feature manipulation, and training-free alternatives such as Ctrl-X (Lin et al., 2024) and FreeControl (Mo et al., 2024) regulate structure and appearance without conventional task-specific training. Collectively, these methods show that a single fixed global control scale is insufficient when text and structural evidence disagree.

Progressive sketching makes this conflict substantially more difficult because the disagreement itself changes over time. A sparse early contour may be highly ambiguous or even contradict the prompt, whereas later strokes can progressively resolve the intended geometry. The appropriate strength of control can therefore vary across interaction turns, architectural layers, and denoising steps. Our pilot analysis further shows that ControlNet perturbations are step- and band-asymmetric and that large inter-turn feature redistribution accompanies structurally disruptive stroke updates. ProgressNet exploits this behavior through Banded Adaptive Control (BAC), deriving step- and band-specific correction factors from retrospective comparisons of cached tile-memory ControlNet features across preceding turns. Unlike learned control policies (Liu et al., 2024) or manually selected control locations (Cohen et al., 2025), BAC derives its modulation directly from the evolving control representation and requires no additional training.

F Nano Banana Pro

Nano Banana Pro, a commercial image-generation and editing model by Google DeepMind, is included as a qualitative reference in Fig. 8 of the main paper. For fine-grained spatial-control comparisons, the same sketch and text instruction were supplied through its public chat interface, with an explicit request to follow the sketch structure. Because the system is proprietary and does not expose the controlled inference protocol required by our completion-level evaluation, we do not include it in the quantitative tables. The comparison is therefore intended only to illustrate differences in interactive sketch adherence under the available interface, not to establish a controlled benchmark ranking.

G Additional Qualitative Examples

We provide additional qualitative comparisons against the strongest competing methods, with all examples drawn from FS-COCO (Chowdhury et al., 2022), which offers a diverse set of single-object, multi-object, and scene-level sketches paired with reference photographs. This diversity allows a thorough evaluation of temporal coherence, sketch faithfulness, and semantic fidelity across several completion levels. For readability, these extended figures focus on SDXS, FLUX+ControlNet, and StableFlow, which represent one-step image-conditioned synthesis, stateless structural control, and strong temporal feature locking, respectively; the full six-method comparison is reported quantitatively in the main paper.

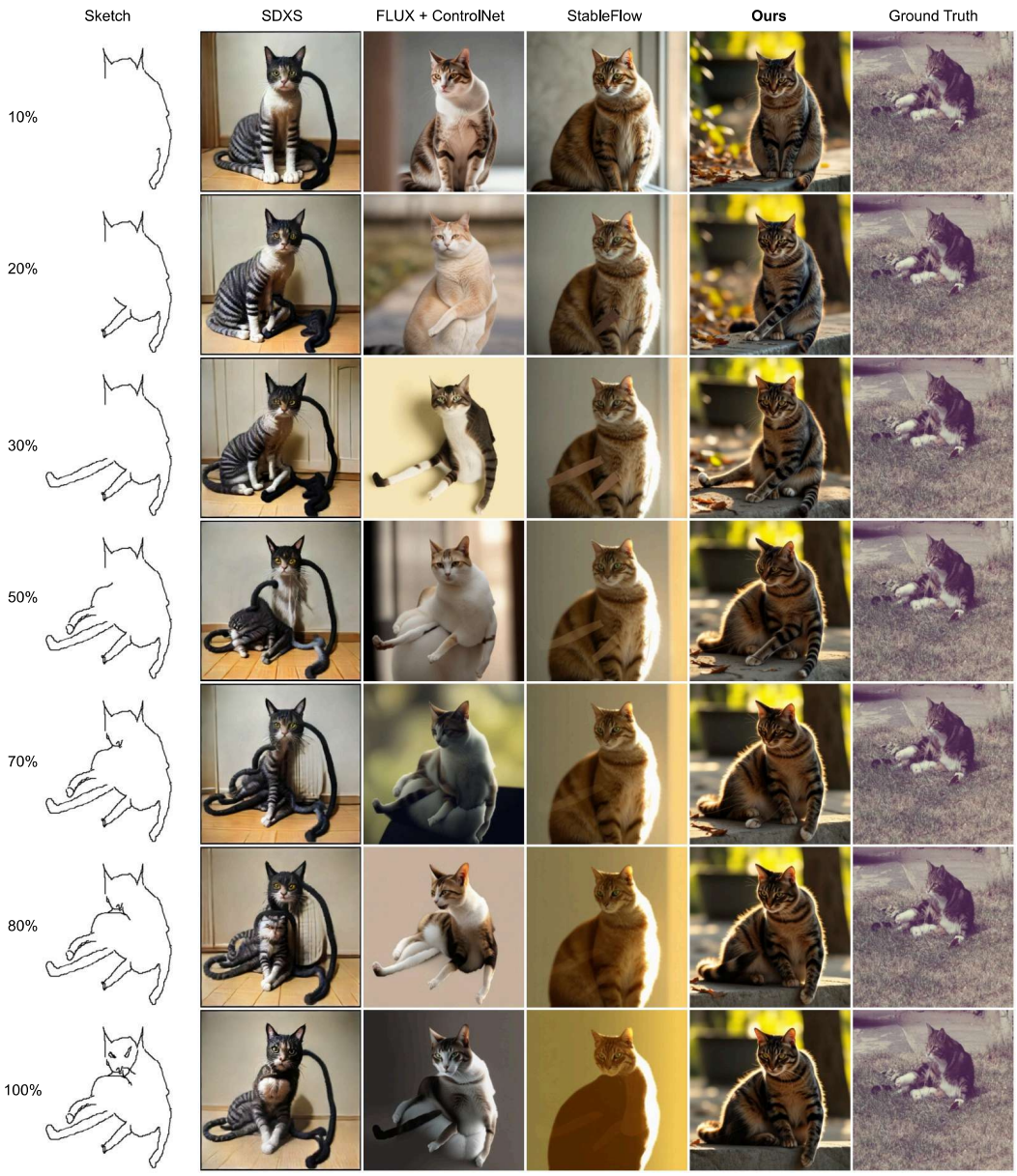


Figure 13: Progressive sketch-to-image generation, extension of Fig. 7. Prompt: `In this scene the cat sitting.` StableFlow shows stronger stroke-region amplification, while SDXS contains visible artifacts. The baseline forms a cat but changes noticeably in appearance and structure across turns. Our method remains responsive to stroke additions while preserving the sketch structure and overall appearance.

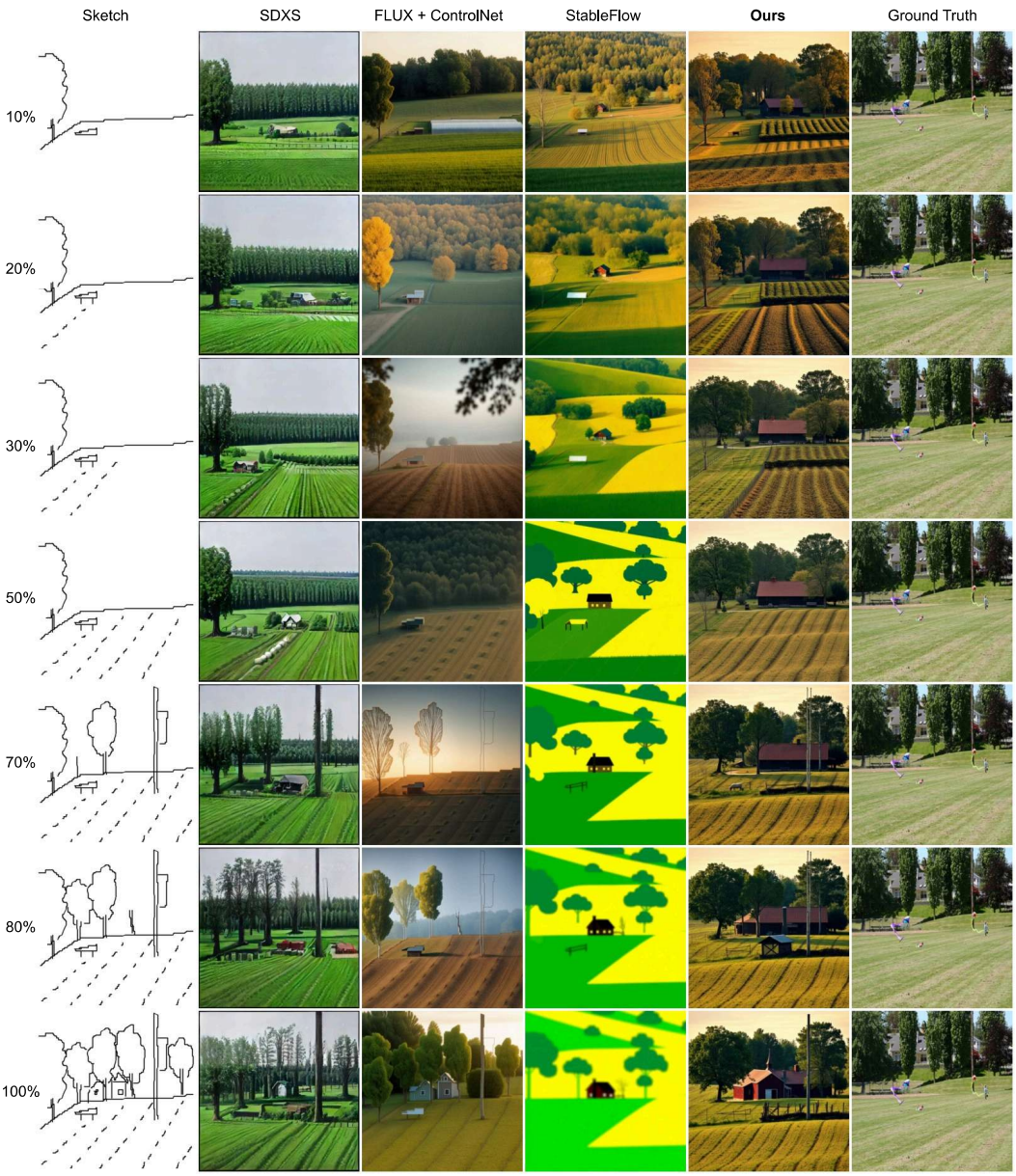


Figure 14: Progressive sketch-to-image generation. Prompt: `farm with trees in surrounding`. StableFlow shows stronger green saturation, while the baseline contains visible artifacts. SDXS captures the overall structure but retains less fine detail. Our approach adapts to sketch changes while preserving scene appearance and details.

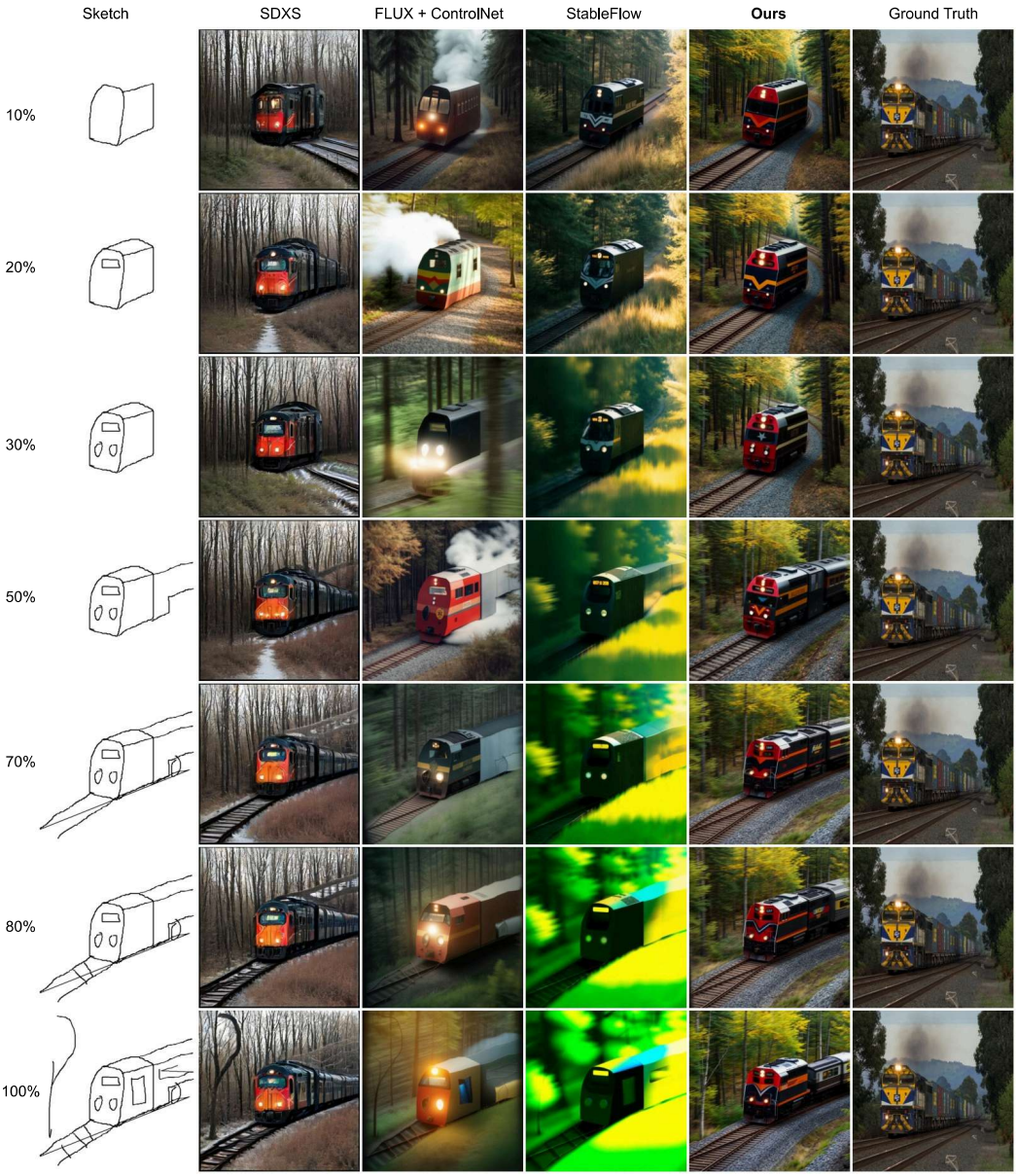


Figure 15: Progressive sketch-to-image generation. Prompt: `a train moving through the woods`. Both the baseline and SDXS contain visible artifacts, while StableFlow shows stronger green saturation. Our method adapts to the final tree-branch updates while retaining a consistent scene appearance.

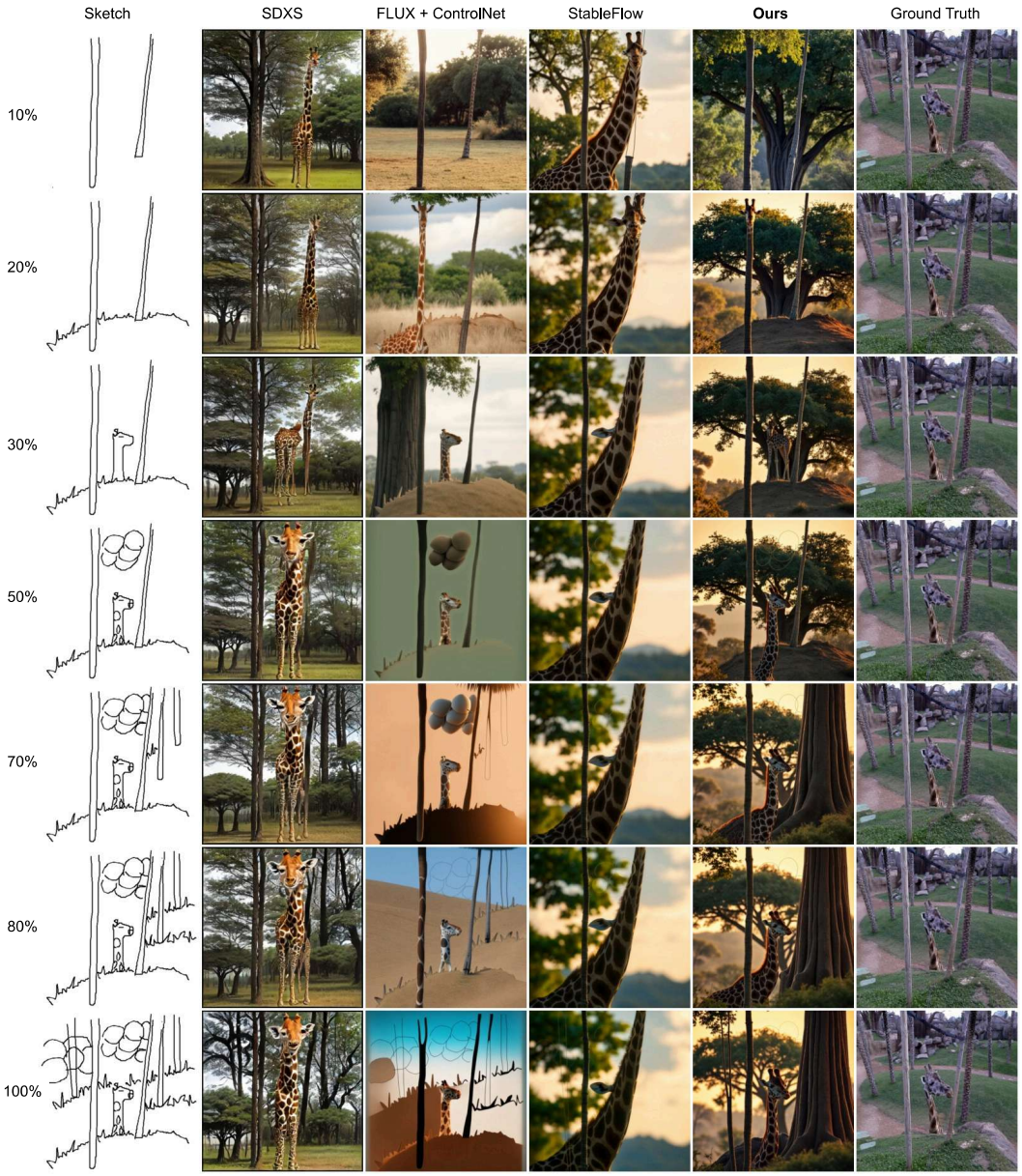


Figure 16: Progressive sketch-to-image generation. Prompt: `a giraffe is standing near a tree`. This is a challenging scenario where the evolving sketch initially differs from the prompt's general subject. StableFlow remains close to the initial structure, while SDXS shows limited adherence to later sketch updates and the baseline contains visible artifacts. Our model progressively adapts to the structural changes, following the initial tree strokes before forming the giraffe structure as additional evidence arrives.

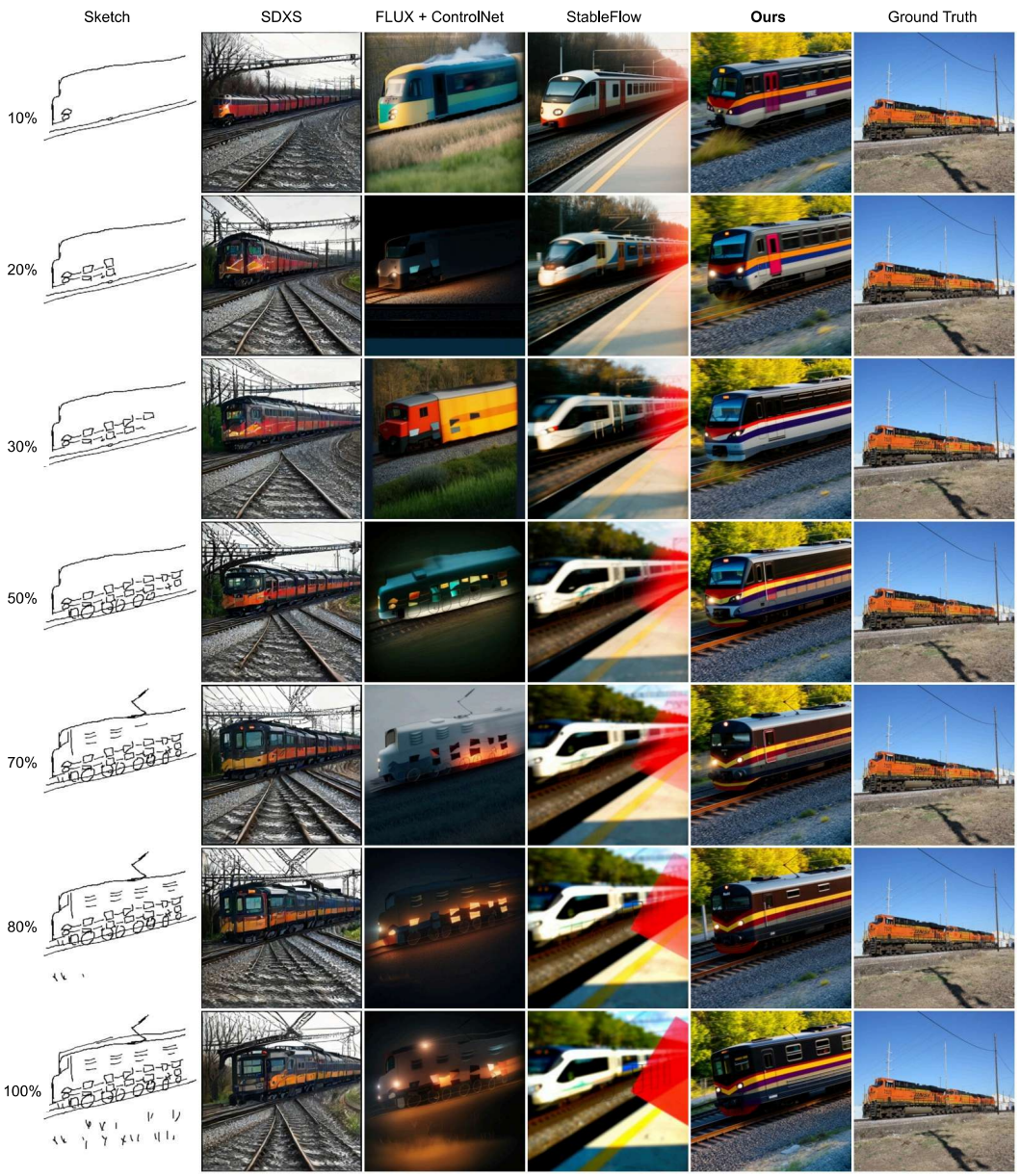


Figure 17: Progressive sketch-to-image generation. Prompt: `A train on a rail track`. StableFlow shows a reddish triangular artifact together with strong persistence of the initial structure and color palette. The baseline follows the sketch closely but produces a flat appearance, while SDXS preserves the main structure with limited detail. Our method progressively integrates specific details (e.g., the pantograph at 70% completion) and preserves them across subsequent steps.

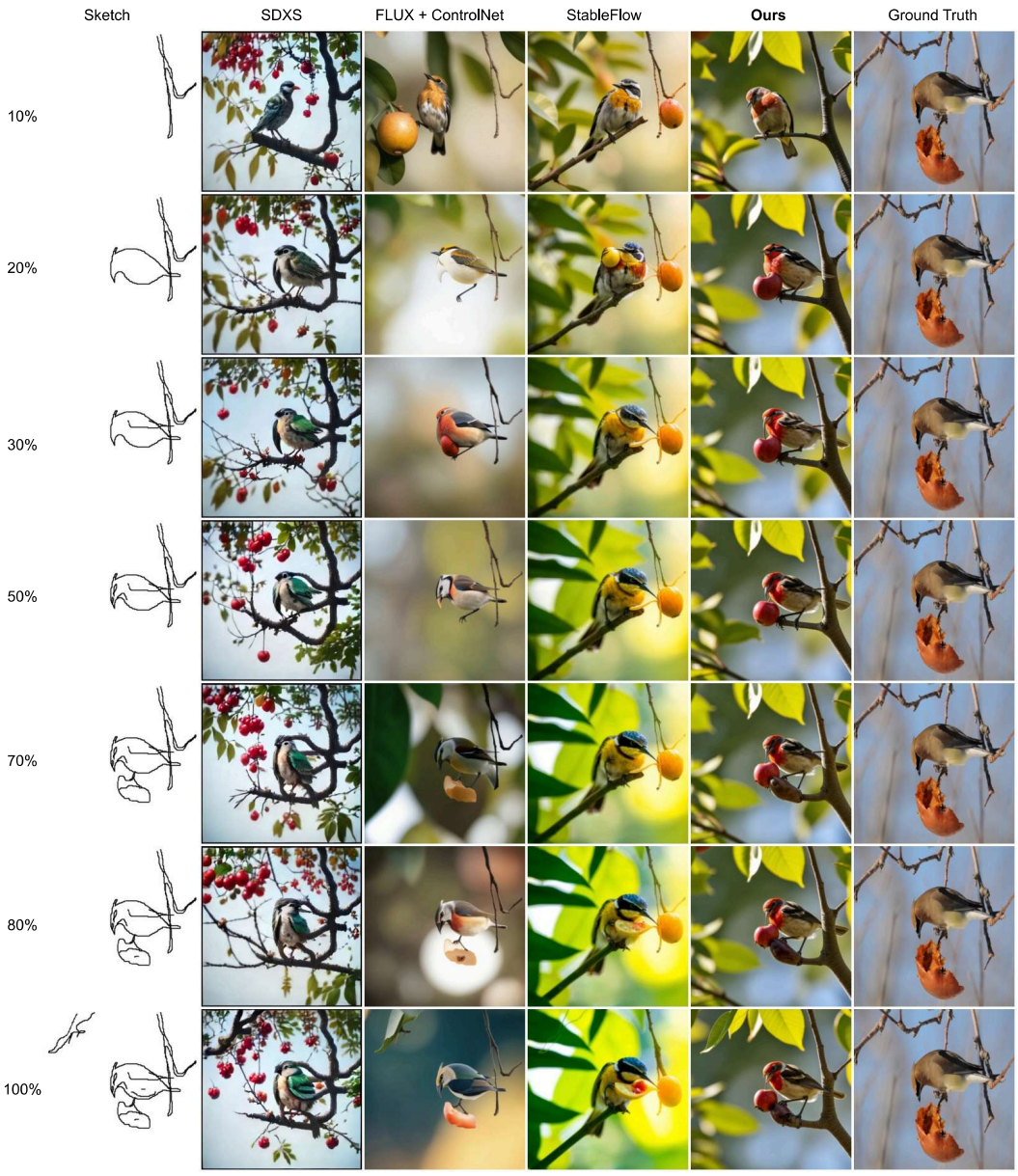


Figure 18: Progressive sketch-to-image generation. Prompt: `A bird is sitting on the branch of a tree eating a fruit.` The baseline shows visible artifacts around imperfect strokes, while SDXS introduces extraneous elements. StableFlow exhibits stronger visual amplification, whereas our method balances the evolving sketch structure and prompt semantics while retaining the intended visual content.

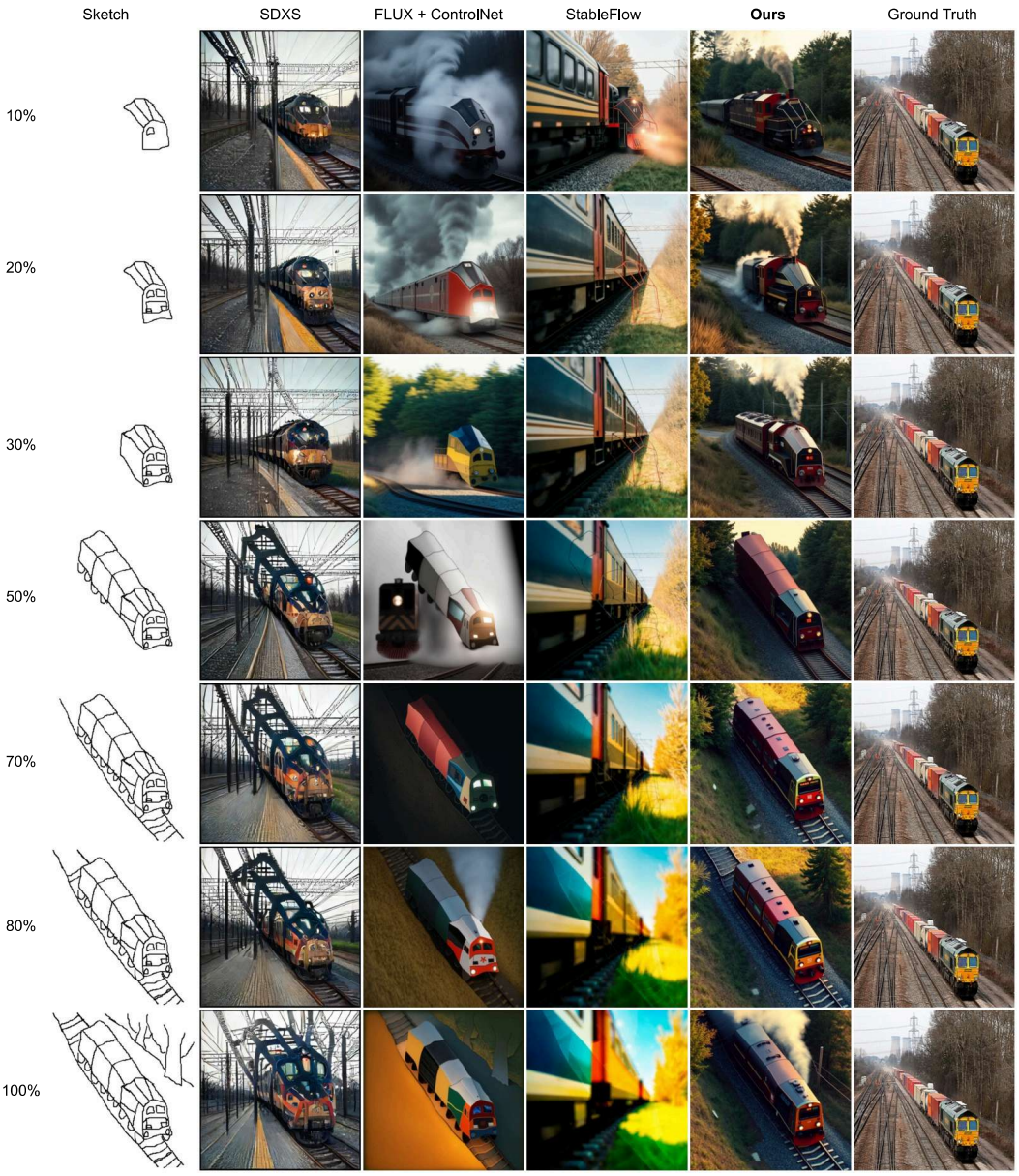


Figure 19: Progressive sketch-to-image generation. Prompt: `a train on the tracks`. Our method maintains a coherent progression across sketch completion levels while adapting the top-down view to later strokes. SDXS and the baseline show more visible artifacts, while StableFlow preserves the initial structure strongly and retains the saturated colors established at the 10% stage.

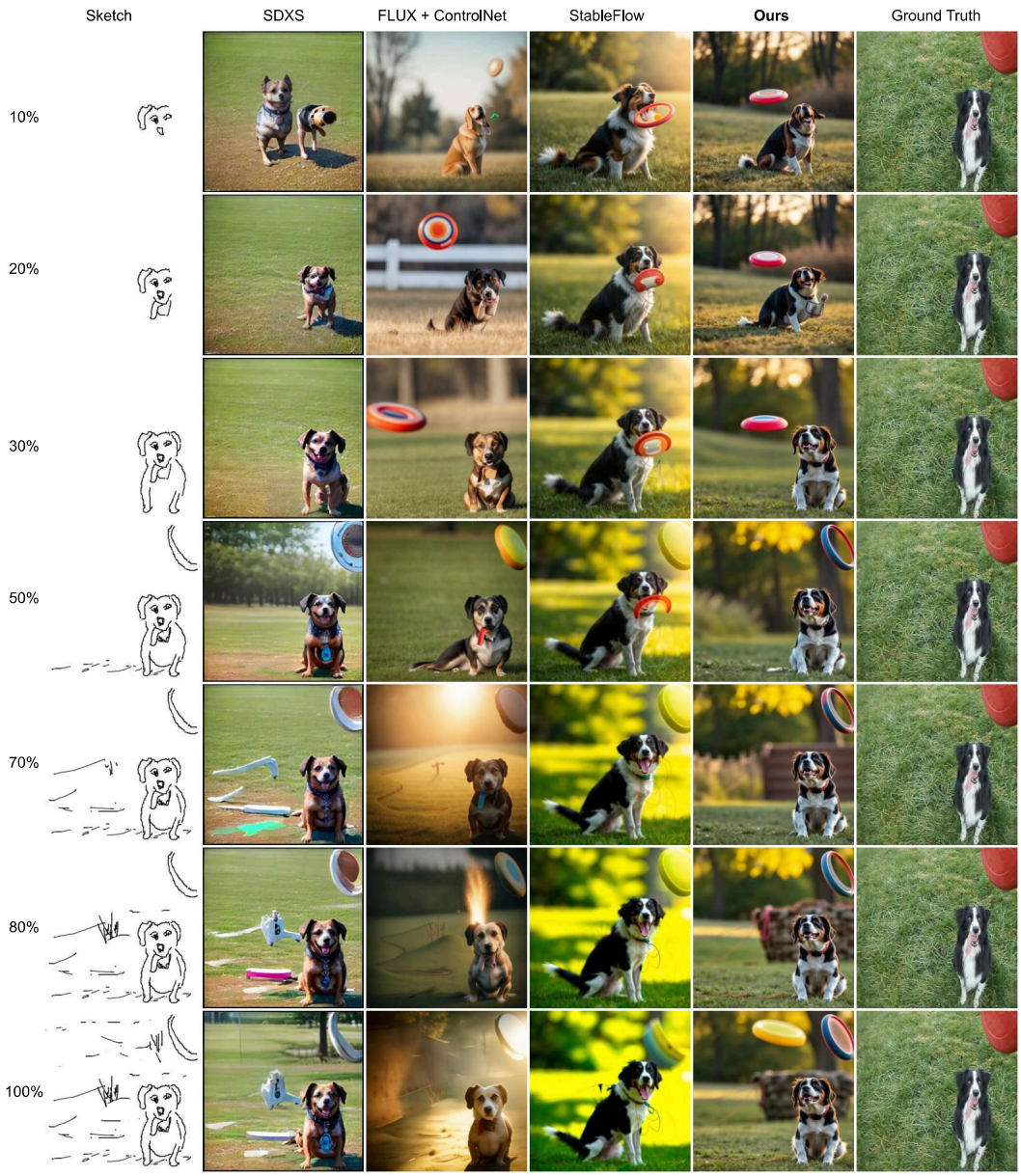


Figure 20: Progressive sketch-to-image generation. Prompt: `A dog sitting and eagerly waiting to catch the frisbee`. StableFlow shows stronger background saturation and visible artifacts at 80% and 100% progression. SDXS produces a high-quality image with some flawed details. Compared to the baseline, our method shows stronger sketch adherence while preserving the dog's facial structure.

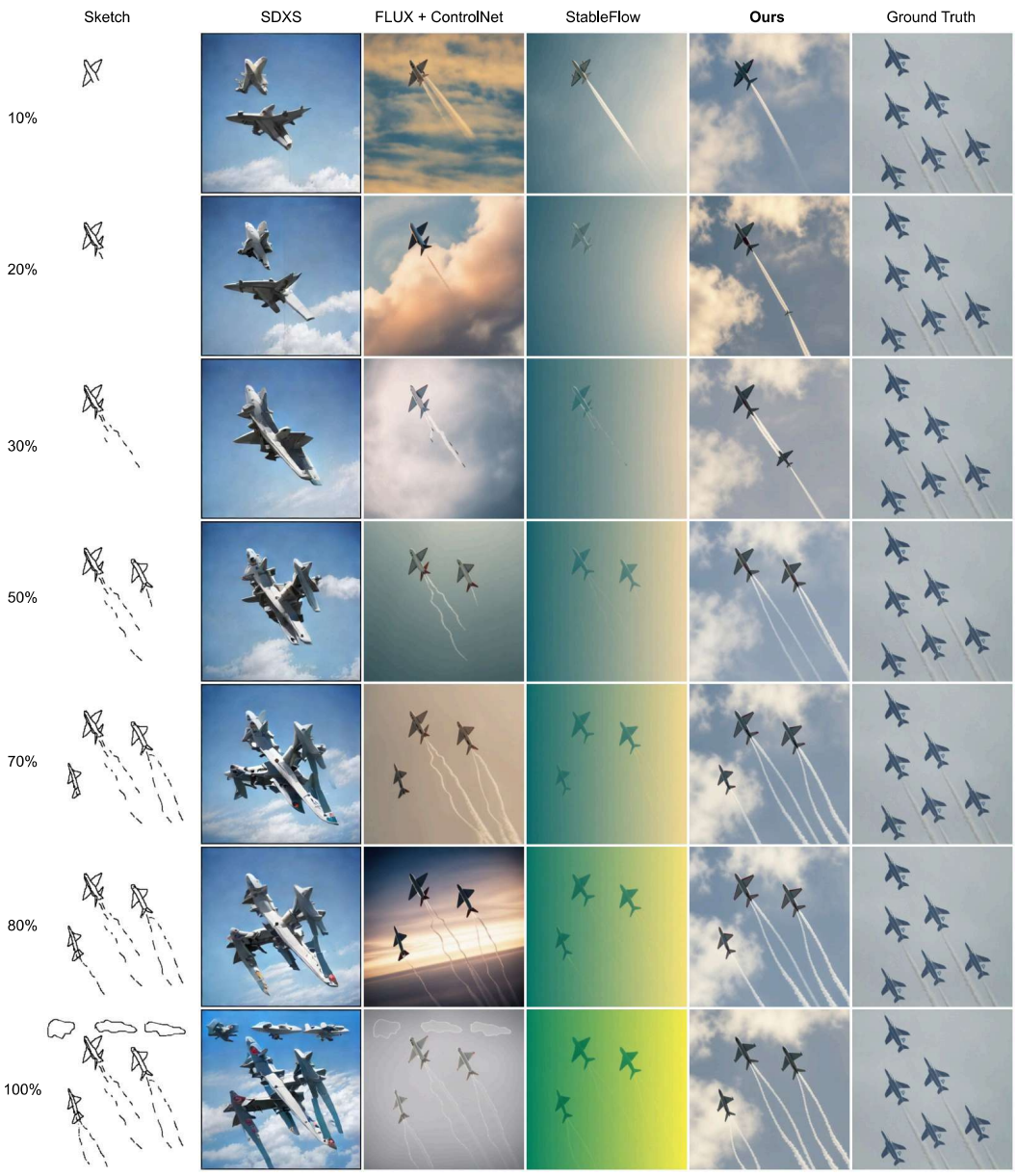


Figure 21: Progressive sketch-to-image generation. Prompt: `Jetplanes flying in the sky`. StableFlow shows strong persistence of the initial structure together with color distortion and weaker adherence to later strokes. SDXS and the baseline contain visible artifacts under the evolving sketch condition. Our method adapts to the evolving sketch while preserving the emerging contrails and reducing visible stroke artifacts.

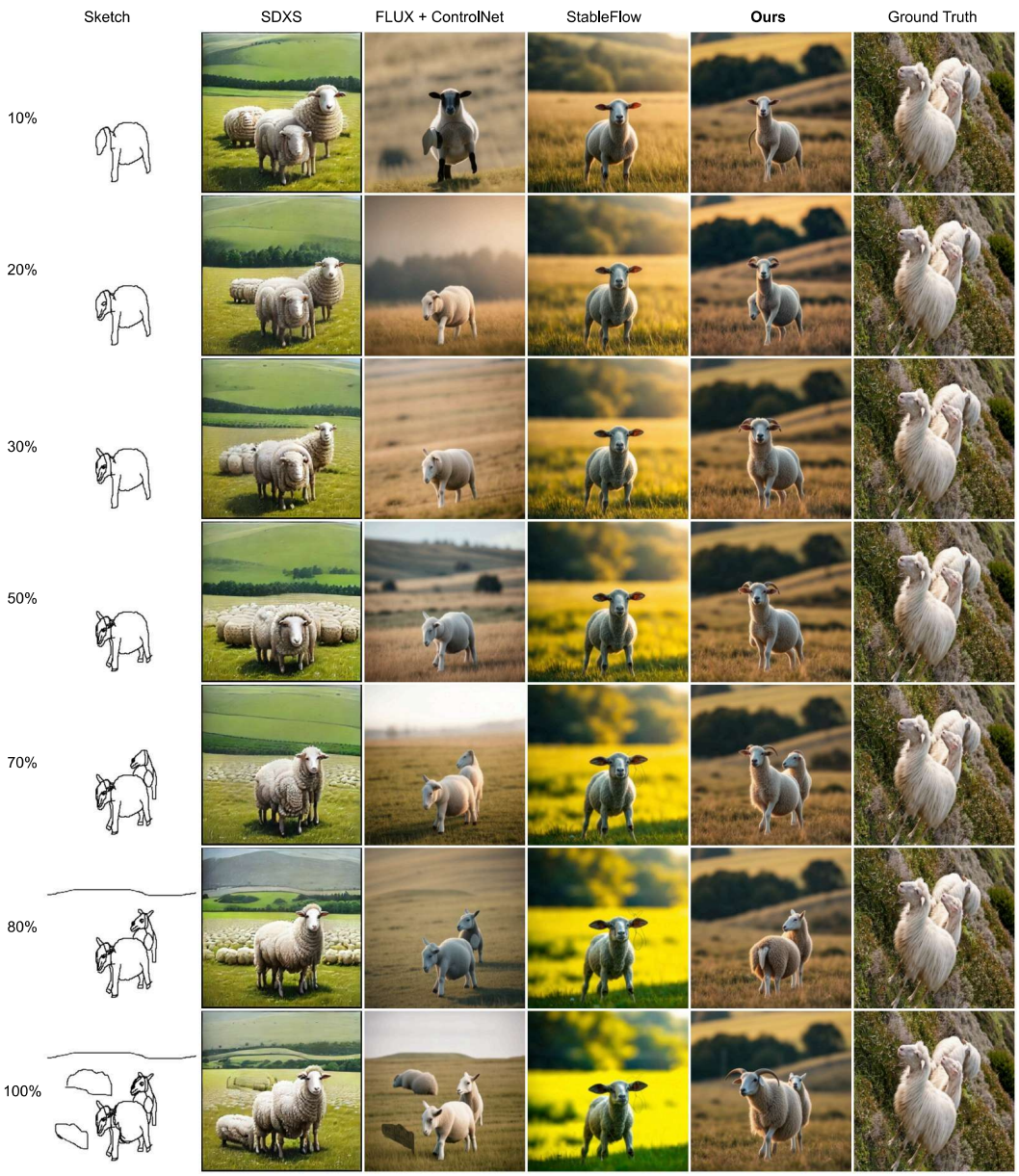


Figure 22: Progressive sketch-to-image generation. Prompt: `sheep standing in a field.` StableFlow remains focused on a single, greenish sheep, while the baseline follows the rough sketch structure but produces an unrealistic animal. SDXS produces a high-quality image with weaker preservation of the underlying sketch structure. Our method maintains sketch fidelity, introducing the second sheep at 70% completion and refining the scene across later stages.



Figure 23: Progressive sketch-to-image generation. Prompt: `player playing a tennis game`. StableFlow shows strong persistence of structure and color together with visible artifacts, while SDXS follows the sketch more loosely and contains image-level artifacts. The baseline also shows substantial artifacts. Our method deviates slightly from the exact pose in the ground truth but retains the overall sketch structure with fewer visible artifacts.

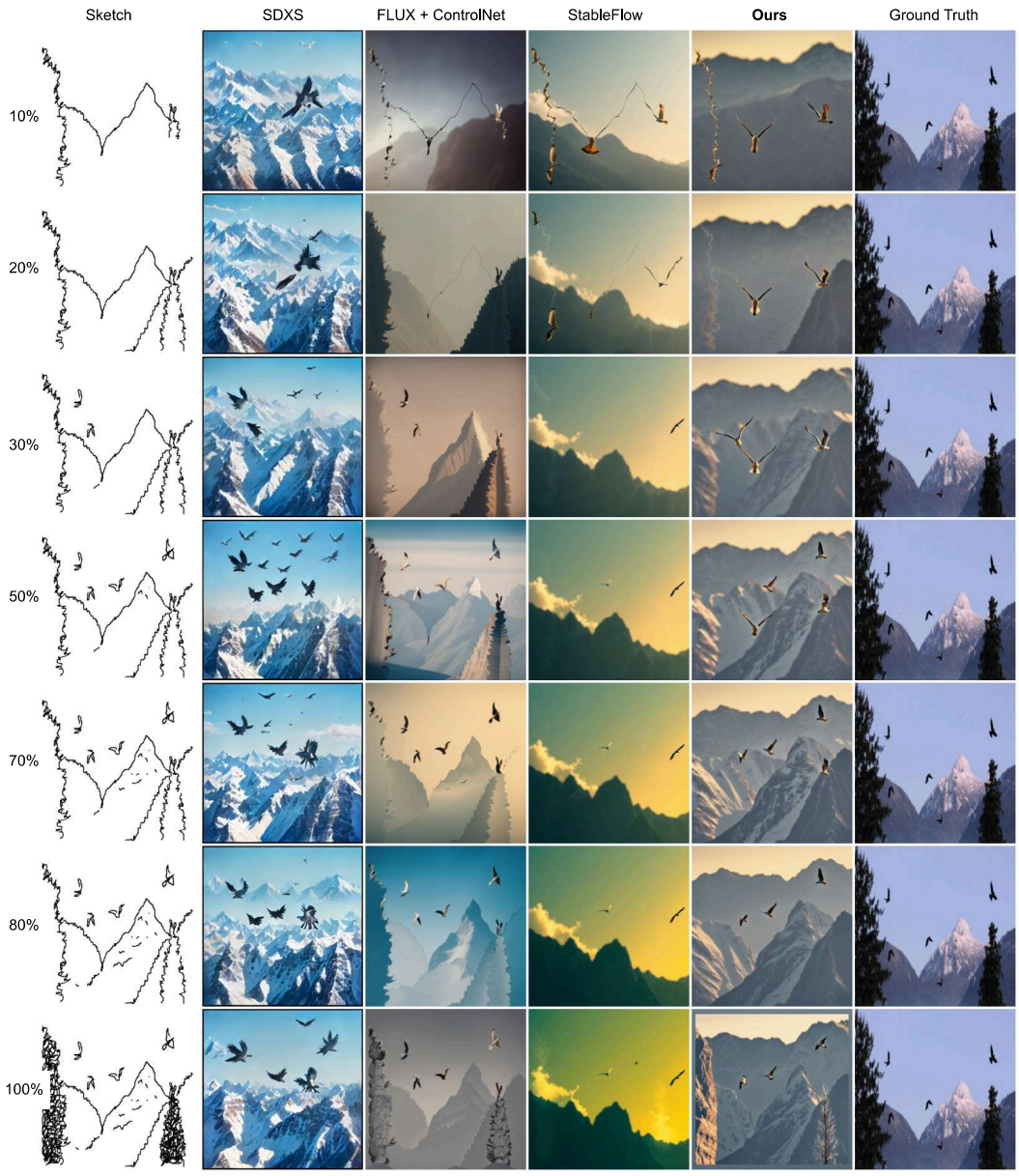


Figure 24: Progressive sketch-to-image generation. Prompt: `Few birds are flying over the mountains`. Our method forms birds from rough strokes while suppressing many of the initial stroke artifacts. The baseline reflects the sketch structure but yields a less photorealistic interpretation. StableFlow shows stronger color saturation and weaker agreement with the final sketch, while SDXS contains substantial image-level artifacts.

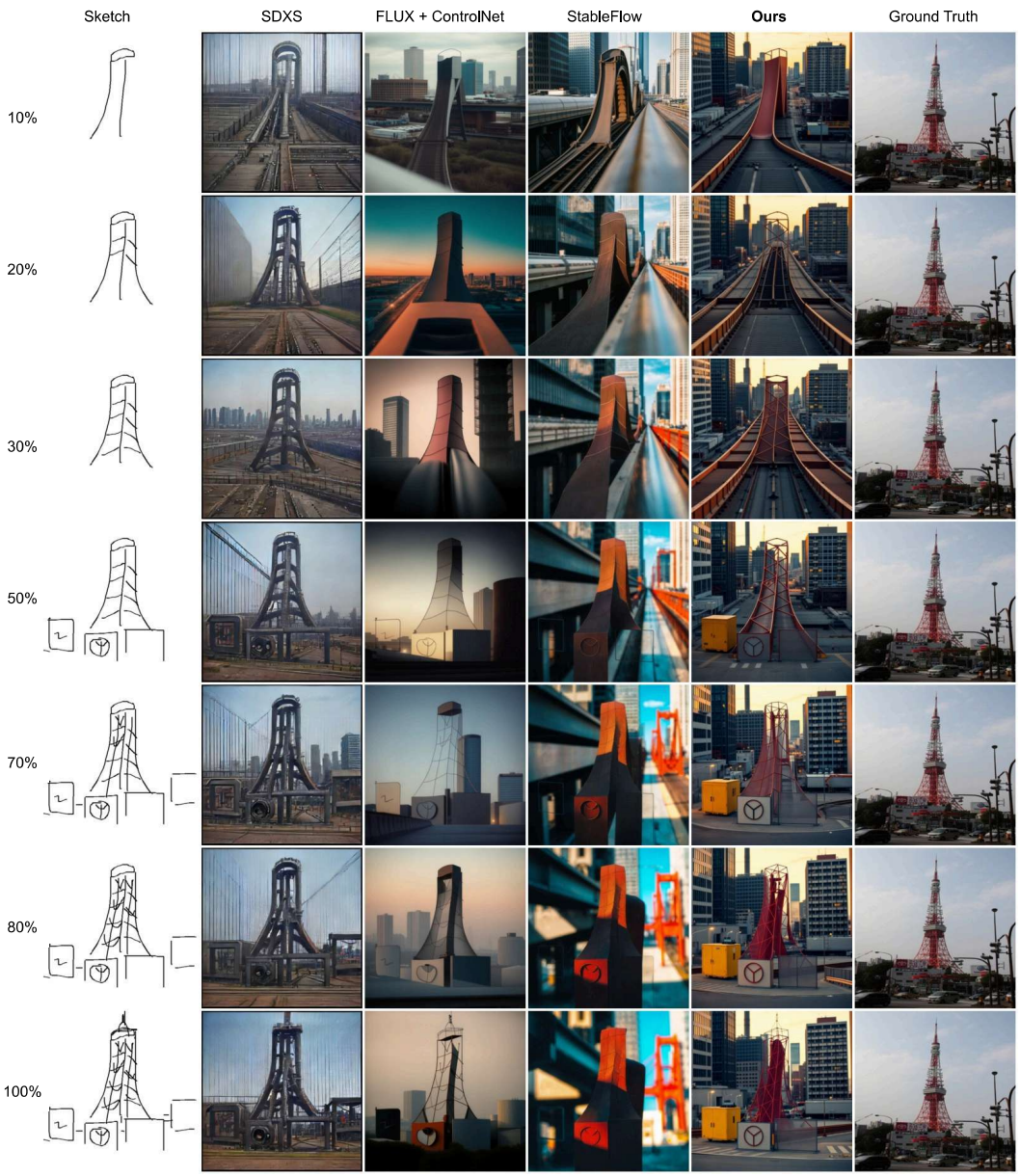


Figure 25: Progressive sketch-to-image generation. Prompt: `Industrial belt in the city`. This represents a challenging scenario where the prompt's open-ended nature makes interpretation difficult. Our method interprets the circular symbol and box-like structures consistently across the progression. The competing outputs show different limitations, including stronger saturation for StableFlow and visible artifacts for the baseline and SDXS.

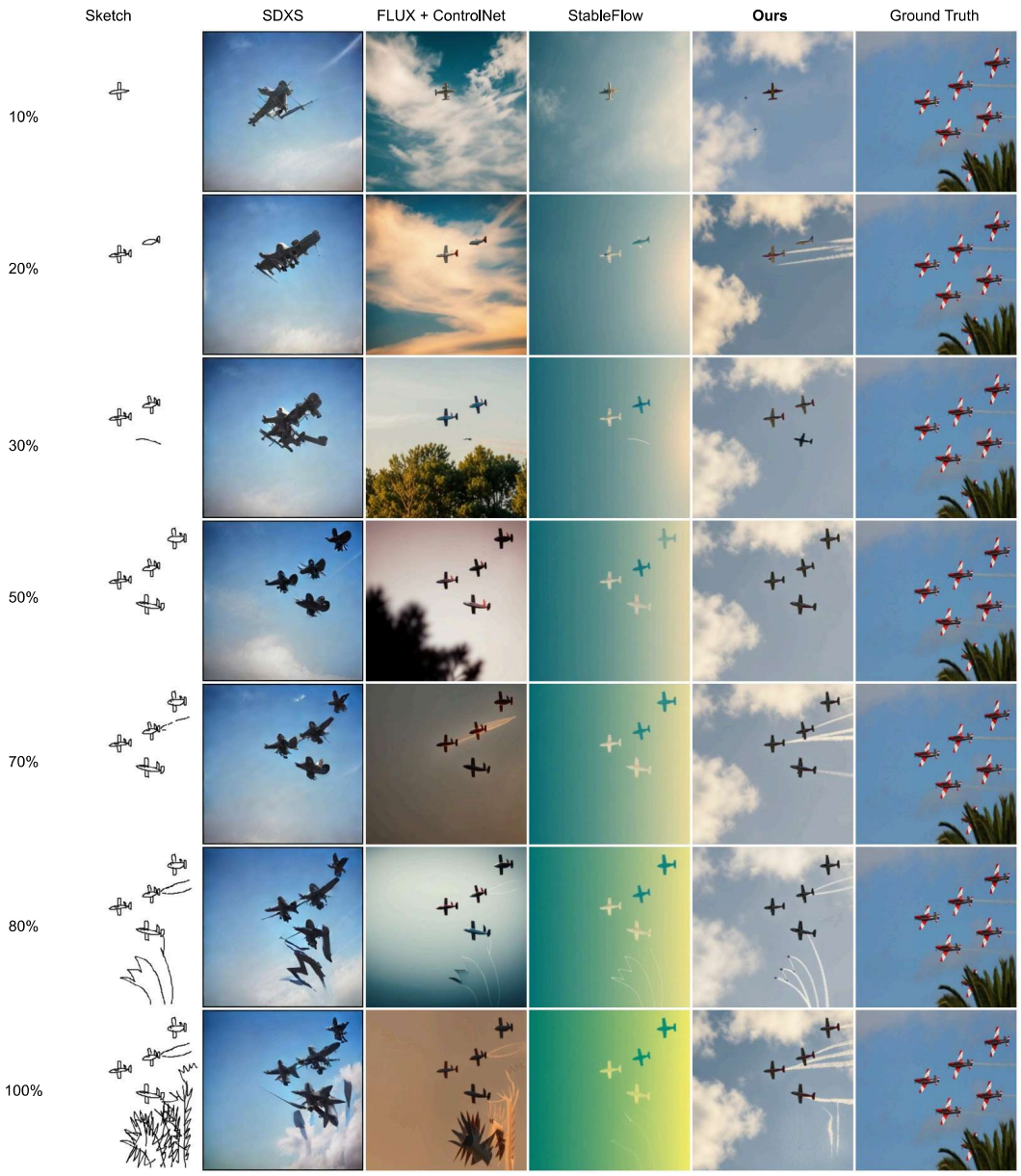


Figure 26: Progressive sketch-to-image generation. Prompt: `aeroplanes are flying in the sky`. StableFlow shows stronger color saturation, while the baseline contains visible imperfections around sketch-aligned regions. SDXS changes noticeably across turns and retains tree-like artifacts. Our method translates the evolving strokes into jets while gradually reducing the early tree-like structures.

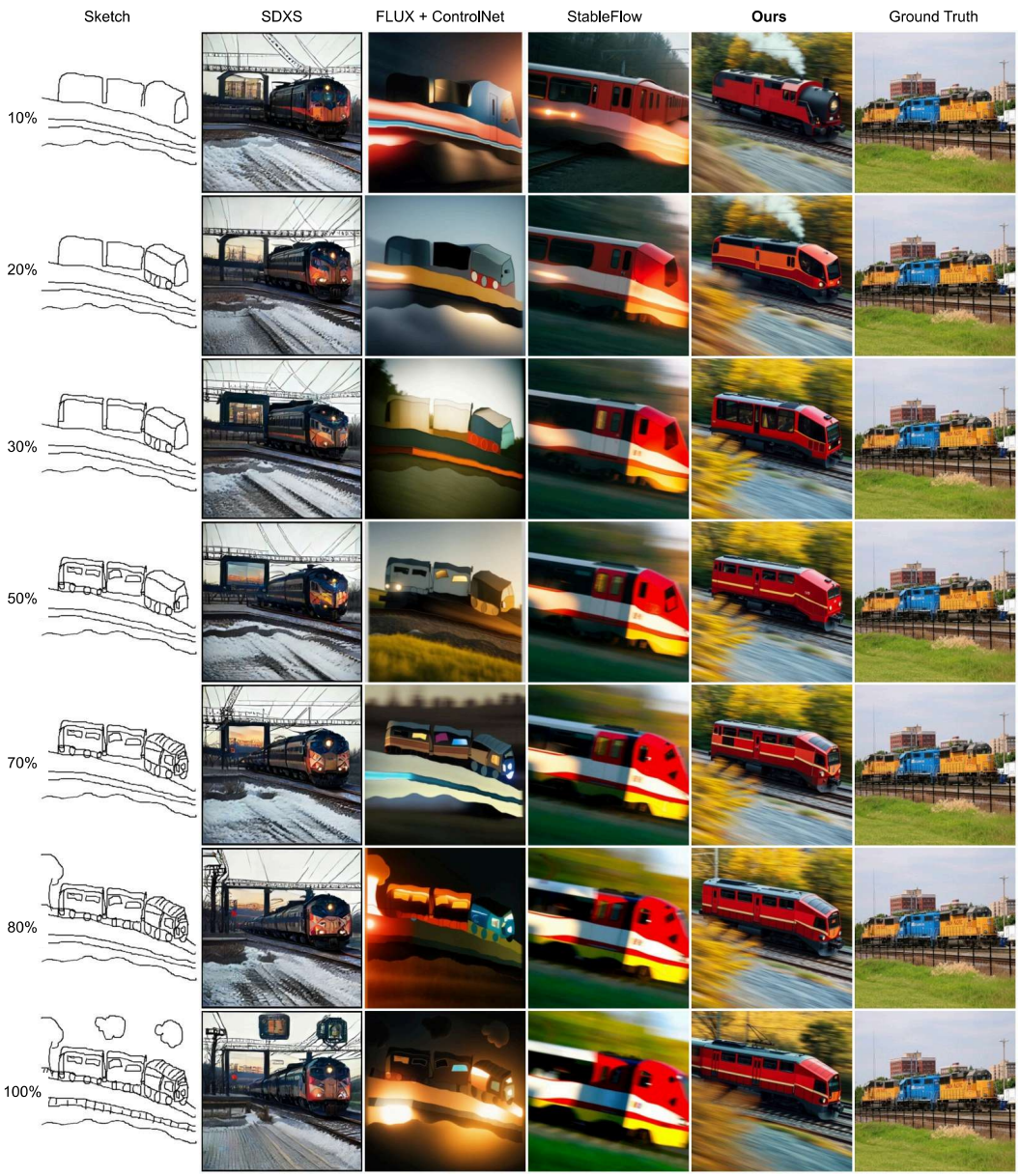


Figure 27: Progressive sketch-to-image generation. Prompt: `a train moving on tracks.` The baseline contains cartoon-like artifacts, while StableFlow strongly preserves the initial structure and shows limited later adaptation together with color distortion. SDXS shows weak sketch adherence. Our method provides smoother progression while balancing the evolving sketch with scene realism.

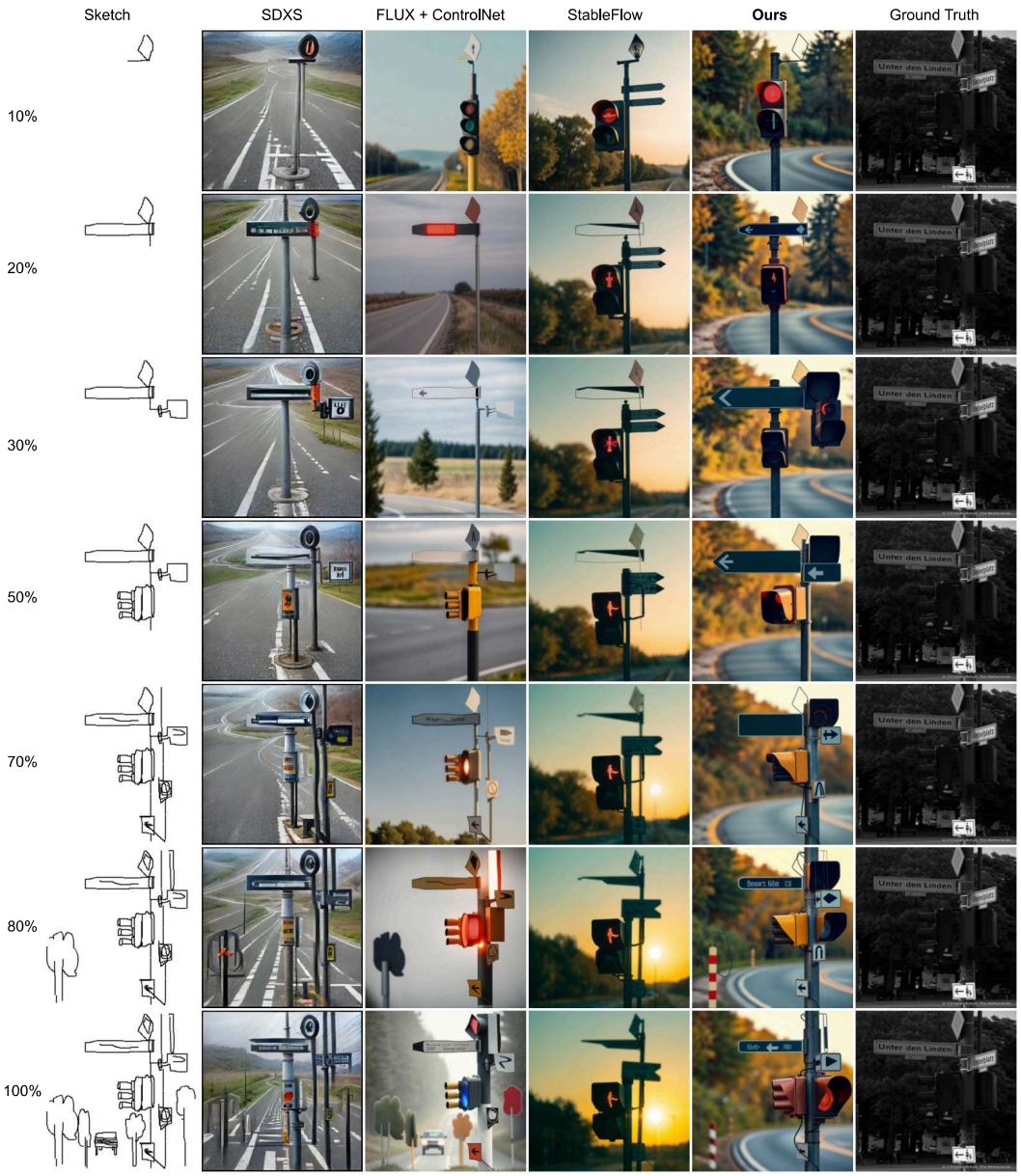


Figure 28: Progressive sketch-to-image generation. Prompt: `signal and direction pole on road`. In this scene-level generation, StableFlow strongly preserves the initial structure and shows color distortion as details are added. The baseline reflects several sketch imperfections in the final image, while SDXS contains limited fine detail. Our method incorporates progressive details while maintaining a coherent road scene with fewer visible artifacts.



Figure 29: Progressive sketch-to-image generation. Prompt: `People are crossing the busy streets.` In this complex, multi-element scene, the baseline becomes more cartoon-like and contains visible artifacts under rough strokes. SDXS shows distorted figures, while StableFlow has stronger color saturation and adapts less to later strokes. Our method maintains a coherent traffic scene with central figures that remain aligned with the evolving strokes.



Figure 30: Progressive sketch-to-image generation. Prompt: `Two planes flying in sky where a coconut tree is also in the frame.` StableFlow exhibits color distortion while maintaining the broad sketch structure, the baseline shows artifacts in the tree leaves, and SDXS shows weaker sketch adherence. Our method progresses from a single airplane to multiple planes as the corresponding strokes are introduced.



Figure 31: Progressive sketch-to-image generation. Prompt: `a park with clock tower and a streetlamp nearby` This example features a vague description of a park; as the user draws trees (despite the prompt focusing on a park and clock), competing methods drift and generate visible artifacts under the evolving sketch condition. Our method demonstrates flexibility by interpreting rough tree strokes as lamp-like structures appropriate for a park scene, while smoothly incorporating user details added at the 70% stage.

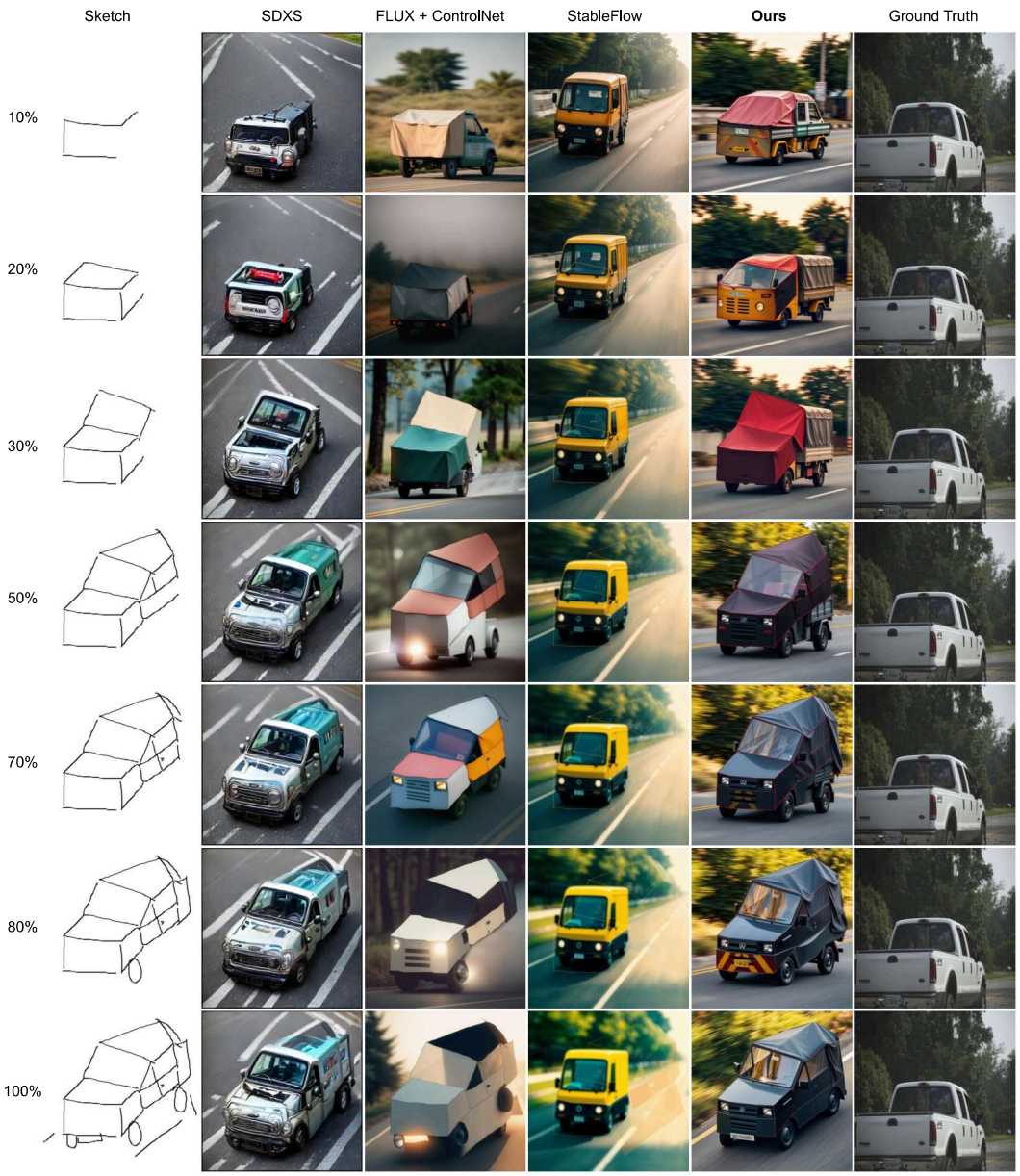


Figure 32: Progressive sketch-to-image generation. Prompt: `A mini truck on road`. This is a challenging task because the upper section of the truck is only partially specified. Our method interprets the incomplete strokes as a plausible vehicle structure and updates it as more evidence arrives. The competing outputs show different behaviors: the baseline contains substantial artifacts, StableFlow strongly preserves the initial structure, and SDXS preserves the overall intent with less fine detail. Our approach updates the structure as new strokes are provided.



Figure 33: Progressive sketch-to-image generation. Prompt: `a train on a track.` StableFlow strongly preserves the initial structure and exhibits substantial color amplification. SDXS retains the broad structure but shows limited response to small corrections, while the baseline contains heavy artifacts. Our method forms a coherent train structure and extends the sky when additional cloud strokes are introduced.

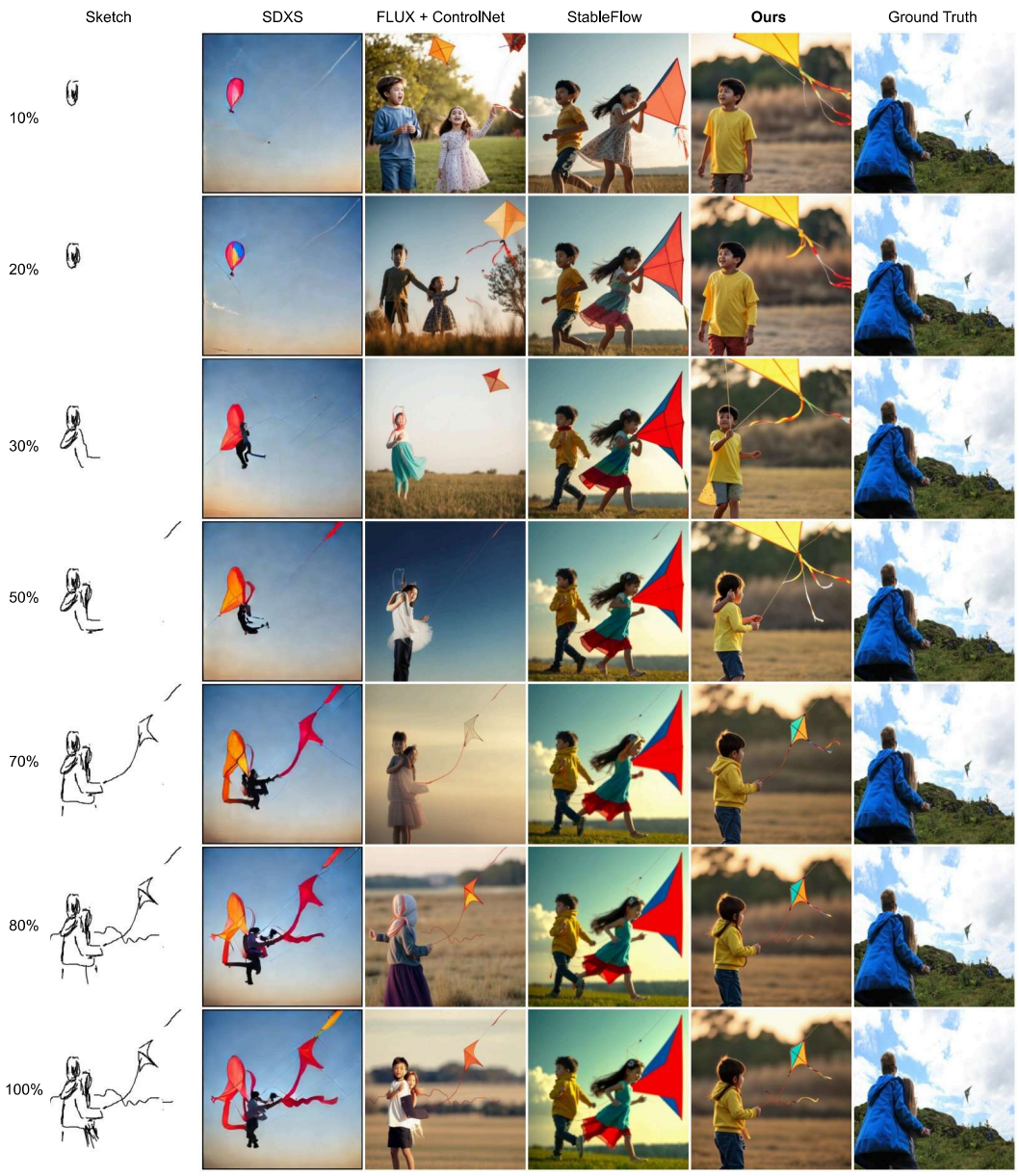


Figure 34: Progressive sketch-to-image generation. Prompt: `A boy and a girl flying a kite`. This case is challenging because the sketch strokes overlap heavily. Heavy stroke overlap remains challenging at the final stages, although the overall scene composition remains photorealistic and temporally stable. SDXS, the baseline, and StableFlow show more drift and visible artifacts, while our method remains comparatively stable in overall composition.

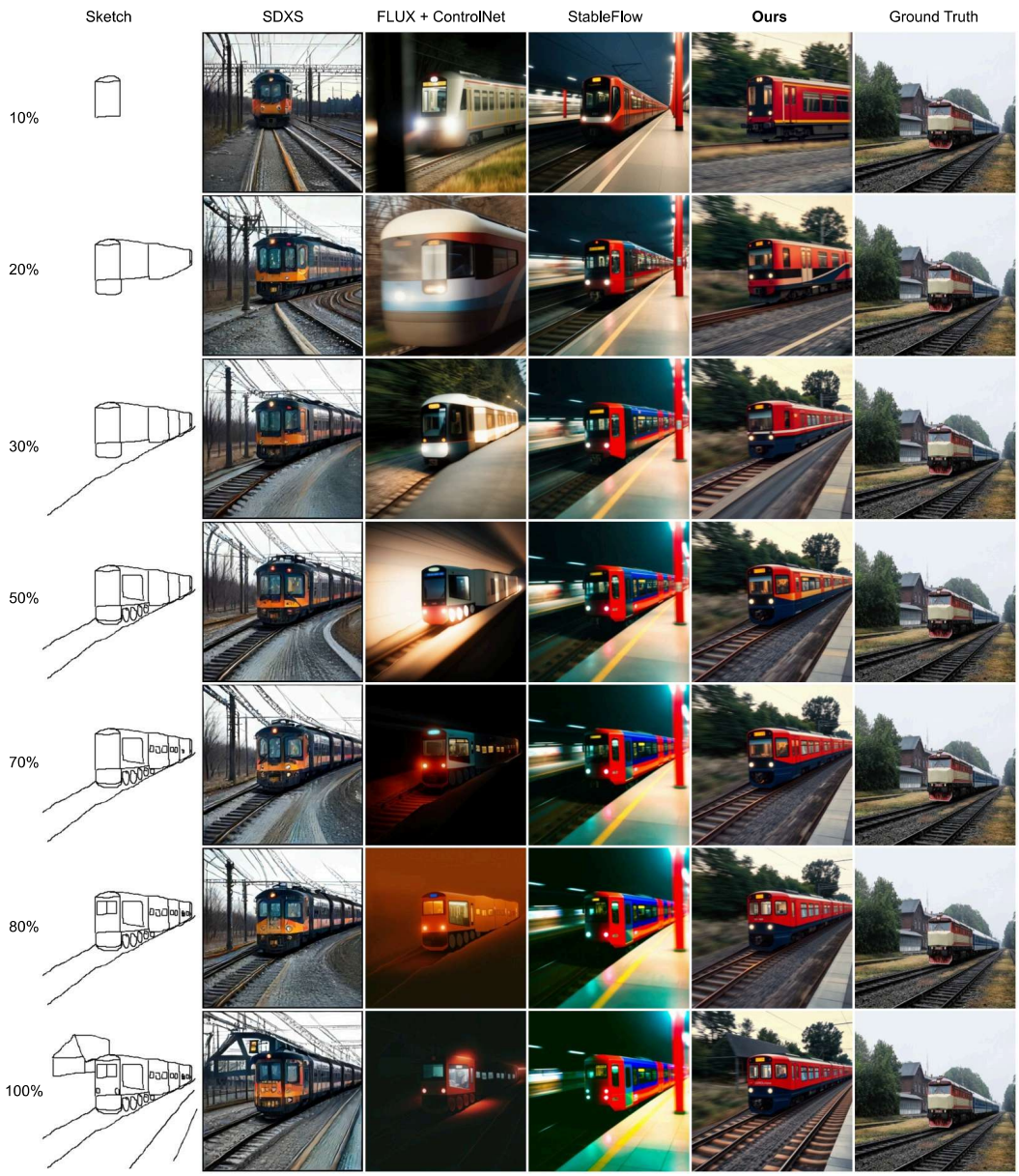


Figure 35: Progressive sketch-to-image generation. Prompt: `A train moving on track`. This demonstrates faithful generation where our method progressively incorporates an extra line intended by the user at 100% completion. SDXS also reacts to the added line but loses some fine detail, whereas our method integrates the new information while preserving more of the established scene.