WHEN DO EXTRA EVALUATIONS PER STEP PAY OFF IN FEW-STEP SAMPLING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step samplers for diffusion and flow models spend a fixed budget of network function evaluations (NFE). A multi-stage solver such as Heun spends some of them inside a step to correct it; a first-order solver spends each on a new step. We ask when an extra evaluation is worth more inside a step. Measured on the forward path of a reference sample, the error of one step is a term shared by all solvers plus a correction from the extra evaluations; for Euler the shared term is the step's share of the remaining time times the model's error in predicting the clean sample. An Euler step is a convex combination of the current state and this prediction, so it is non-expansive whenever the prediction is. The correction grows with the ratio h/k of the step length to the distance of its end to the clean image, where the sampling equation becomes stiff. Extra evaluations should therefore pay off only on short steps away from the image, and their cost inside long steps should shrink as the budget grows. A solver menu graph records the one-step error of eight solvers on every interval of a fine grid and plans the timestep and solver of every step for an exact budget by dynamic programming. On three undistilled models, one Heun step never beats two Euler steps once $h/k > 5$. On GenEval at 10 NFE, re-placing Euler steps raises the score by up to 0.107, while solvers that spend evaluations inside a step lose 0.031 on SDXL and 0.090 on Qwen-Image on average; at 15 NFE the loss falls to 0.010 and 0.028. On FLUX.1-dev and SD3.5-Large, five Heun steps score 0.05 below ten Euler steps, both placed by the native scheduler. An Euler schedule calibrated on a few prompts compares favourably with Align Your Steps and GITS on SDXL at 10 NFE, and paths planned on Qwen-Image carry over to Qwen-Image-Edit, where they bring edits closer to the 50-step result. On the distilled Z-Image-Turbo, whose graph has almost no window, two planned Euler steps score as high as its default eight. In few-step sampling, evaluations should thus go first to placing first-order steps, and into a step only where the step is short relative to its distance from the clean image.

1 INTRODUCTION

A diffusion or flow model is sampled by integrating an ordinary differential equation from noise to data, and the cost of a sample is the number of network function evaluations (NFE). A fixed budget can be spent in two ways. A first-order solver such as Euler (Hairer et al., 1993) spends one evaluation per step, so ten NFE buy ten steps. A multi-stage solver such as Heun spends several evaluations inside one step to correct it, so ten NFE buy five more accurate steps; multistep solvers such as DPM-Solver++(2M) raise the order by reusing the previous evaluation instead (Figure 1a). Current samplers fix one solver for the whole trajectory (Song et al., 2021a; Lu et al., 2022; 2025; Zhao et al., 2023), and a higher order is the default choice: EDM integrates with Heun (Karras et al., 2022). The timesteps come from a heuristic or a separate optimisation for the chosen solver (Sabour et al., 2024; Chen et al., 2024), and solver schedule search varies the solver along the trajectory by searching over sampled configurations (Liu et al., 2024).

The case for a higher order rests on the limit of small steps. At 5 to 15 NFE the steps are long, and it is not known where a second evaluation is worth more inside a step than as a new step. Solver papers compare whole trajectories, and schedule optimisers take the solver as given, so neither measures the two uses of an evaluation against each other on the same interval.

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

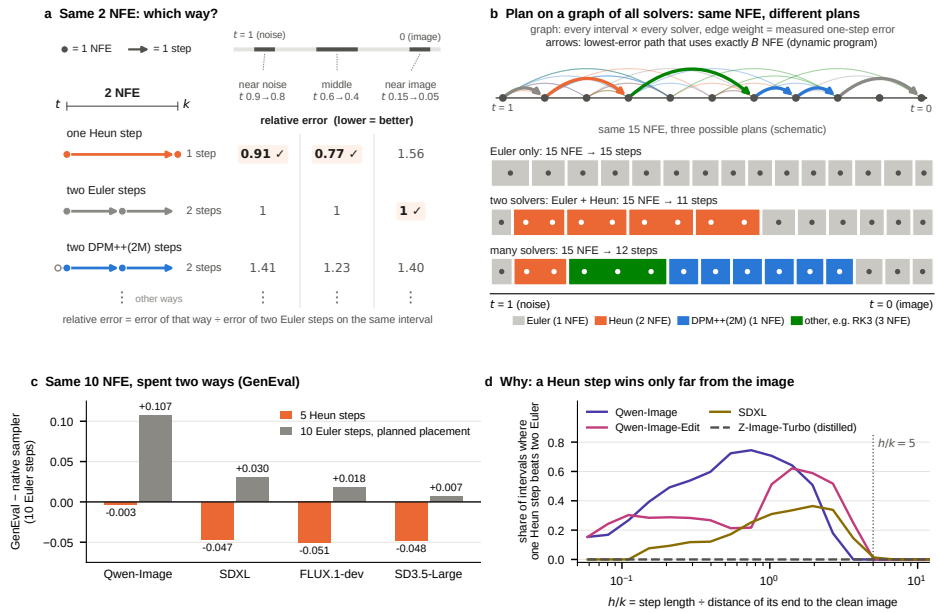


Figure 1: NFE = one evaluation of the model's velocity, including classifier-free guidance (CFG: two network passes on SDXL, the Qwen models, and SD3.5-Large; one on FLUX.1-dev and Z-Image-Turbo); a step = one update of the sample. (a) Two NFE on an interval $[t, k]$ buy one Heun step, two Euler steps, or two DPM-Solver++(2M) steps; error of each divided by that of two Euler steps on Qwen-Image (lower is better) near the noise, in the middle, and next to the clean image. (b) The graph has one edge per interval of a fine time grid and per solver, weighted by the measured one-step error; a dynamic program returns the lowest-error plan with exactly B NFE. (c) Ten NFE spent two ways, GenEval relative to the native sampler (ten Euler steps): five Heun steps (planned schedule on SDXL and Qwen-Image, native schedule on FLUX.1-dev and SD3.5-Large) and ten Euler steps on the graph-planned schedule. Table 1 gives all eight solvers at 10 and 15 NFE. (d) Share of intervals on which one Heun step has a smaller one-step error on the reference path than the best two Euler steps, against h/k, where $h = t - k$ is the length of a step from t to k and k its distance to the image.

We answer this question from the error of one step. Measured on the forward path of a reference sample, the error of one step of any consistent solver is the Euler error, which every solver shares, plus a correction built from the additional evaluations (Section 3). The Euler error is the step's share of the remaining time times the model's error in predicting the clean sample. An Euler step is a convex combination of the current state and this prediction, so it does not amplify an error inherited from earlier steps. The correction of a multi-stage step contains the Jacobian of the velocity, which grows as $1/k$ near the clean image, so its effect scales with h/k, the step length over the distance of the step's end to the clean image. Where the sampling equation becomes stiff, the correction loses accuracy. This predicts a window: an extra evaluation inside a step pays off only on steps that are short relative to their distance from the image, which the budget and the placement of the steps decide.

To measure the window we build a solver menu graph. For one calibration trajectory per model and every pair of times on a fine grid, it stores the error, NFE, and history requirement of one step of each of eight solvers (Euler, Heun, RK3, and RK4 (Hairer et al., 1993), DPM-Solver-2S and 3S (Lu et al., 2022), DPM-Solver++(2M) (Lu et al., 2025), and a closed UniPC predictor–corrector (Zhao et al., 2023)). A dynamic program over the graph selects the timestep and the solver of every step for an exact NFE budget, without training (Figure 1b). The graph shows the window on Qwen-Image and Qwen-Image-Edit (Wu et al., 2025) and SDXL (Podell et al., 2024): one Heun step never beats two Euler steps once $h/k > 5$ (Figure 1d), and on the distilled Z-Image-Turbo (Z-Image Team, 2025) it never wins. Image quality follows the same pattern. At 10 NFE, samplers that spend evaluations inside a step lose GenEval against Euler on the same budget, and the loss shrinks at 15 NFE (Table 1). On FLUX.1-dev (Black Forest Labs, 2024) and SD3.5-Large (Esser et al., 2024), five Heun steps score 0.05 below ten Euler steps, whereas placing the ten Euler steps by the graph raises the score on all four models (Figure 1c). A few-step sampler should therefore spend its evaluations first on

placing first-order steps, and inside a step only where the step is short relative to its distance from the clean image.

Our contributions are: (i) an analysis of where an extra evaluation pays off, from a decomposition of the one-step error into a shared term and a correction, with the Euler term in closed form, the non-expansiveness of Euler steps, and the h/k scaling of the correction (Section 3); (ii) measurements of the window on the one-step errors of four models and of its effect on the image quality of six (Sections 4 and 5); (iii) the solver menu graph and an exact-budget planner, whose Euler layer, measured with about 100 NFE per prompt, gives schedules that compare favourably with Align Your Steps and GITS on SDXL and transfer from Qwen-Image to Qwen-Image-Edit.

2 SETTING

All models are sampled by integrating $\mathrm{d}x/\mathrm{d}t = v_\theta(x,t)$ from $t=1$ (noise) to $t=0$ (clean image); for the flow models (Lipman et al., 2023) $x_t = (1-t)x_0 + t\varepsilon$, and SDXL uses $t = \sigma/\sigma_{\max}$ (Appendix C). One NFE is one evaluation of the guided velocity v_θ: two network passes under classifier-free guidance (Ho & Salimans, 2022), one on Z-Image-Turbo and FLUX.1-dev. All comparisons are at equal NFE. For a fixed noise ε, the native pipeline at its fine schedule (50 steps; 8 on Z-Image-Turbo) gives the *reference sample* x_0 and its forward path, the *reference path* $x_t^* = (1-t)x_0 + t\varepsilon$ (on SDXL $x_0 + t\sigma_{\max}\varepsilon$). This is the path on which the flow-matching loss supervises the model for the pair (x_0, ε), and the residual $r_t = v_\theta(x_t^*, t) - (\varepsilon - x_0)$ is the model's flow-matching residual on this sample. With the clean-sample prediction $\hat{x}_0(x) = x - t\,v_\theta(x,t)$, it equals $r_t = -\big(\hat{x}_0(x_t^*) - x_0\big)/t$. A *deployable step* of solver s from t to k starts at x_t^*, calls the network as in sampling, and lands at $\Phi_s(x_t^*; t \to k)$; its error $\mathrm{err}_s(t \to k)$ is the mean squared distance to x_k^*. The menu has eight solvers (Table 4, Appendix A): single-step multi-stage methods at one NFE per stage, the multistep DPM-Solver++(2M), which reuses the previous evaluation, and a closed UniPC-2 corrector at 2 NFE (1 NFE in the stateful implementation, which reuses the corrector evaluation in the next step; Appendix B).

3 WHERE AN EXTRA EVALUATION CAN PAY OFF

We compare solvers through the error of one deployable step. This error splits into a term shared by every solver and a solver-specific correction. Four consequences follow from this split, and summing the errors along a path gives the cost that the planner minimises. Derivations are in Appendices H and I.

3.1 A SHARED LEADING TERM AND A SOLVER-SPECIFIC CORRECTION

Let $h = t - k > 0$ be the step length. One Euler step from x_t^* lands at $x_k^* - h\,r_t$, so its error is

$$\mathrm{err}_E(t \to k) = h^2\,\|r_t\|^2. \tag{1}$$

In terms of the clean-sample prediction, $\mathrm{err}_E(t \to k) = (h/t)^2\,\|\hat{x}_0(x_t^*) - x_0\|^2$: the square of the step's share of the remaining time times the model's error in predicting the reference sample at t. Any consistent solver that starts from the same first evaluation advances by $-h$ times a convex combination of stage velocities, so its step is the Euler step plus a correction: $\Phi_s(x_t^*) = x_k^* - h\,r_t - h\,C_s(t,k)$, where C_s combines the differences between stage velocities and the first velocity. For Heun, $C = \frac{1}{2}\big(v_\theta(x_k^* - hr_t, k) - v_\theta(x_t^*, t)\big)$. Hence

$$\mathrm{err}_s(t \to k) = h^2\,\|r_t + C_s\|^2 = \mathrm{err}_E + h^2\big(2\langle r_t, C_s\rangle + \|C_s\|^2\big). \tag{2}$$

The first term is shared by every solver. The extra evaluations add only the second term, and they reduce the error only if $\langle r_t, C_s\rangle < -\frac{1}{2}\|C_s\|^2$: the correction must point against the residual without overshooting it.

3.2 FOUR CONSEQUENCES

Proposition 1 (Short intervals). *If v_θ is Lipschitz in (x,t) with constant L on the region traversed by the stages, then $\|C_s\| \le c_s L\,h\,(1 + \|v_\theta(x_t^*, t)\|)$ for a constant c_s depending only on the tableau of s, so* $\mathrm{err}_s/\mathrm{err}_E = 1 + O\big(hL(1+\|v_\theta\|)/\|r_t\|\big)$.

On the unit intervals of a fine grid every solver therefore nearly reproduces the Euler error.

Proposition 2 (Euler steps do not amplify errors)**.** *An Euler step from t to k is the convex combination $\Phi_E(x) = \frac{k}{t}\, x + \left(1 - \frac{k}{t}\right) \hat{x}_0(x)$. If the prediction $\hat{x}_0$ does not amplify perturbations, $\|\partial_x \hat{x}_0\| \le 1$, then Φ_E is 1-Lipschitz for every step length.*

An inherited error therefore passes through an Euler step of any length without growth.

Proposition 3 (Stiffness near the clean image)**.** *Expanding the Heun correction around the reference state at k gives $C = \frac{1}{2}\left(r_k - r_t\right) - \frac{1}{2} h\, J_k r_t + O(h^2)$ with $J_k = \partial_x v_\theta(x_k^*, k)$. Heun beats Euler on $[t, k]$ if and only if $\|r_t + r_k - h J_k r_t\| < 2\|r_t\|$ up to the $O(h^2)$ term.*

The intermediate evaluation is made at the Euler prediction $x_k^* - h r_t$, which lies off the reference path by $h r_t$. For flow and diffusion models J_k grows like $1/k$ as $k \to 0$, because the velocity is essentially $(x - \hat{x}_0(x))/k$, so $h J_k r_t$ scales like $(h/k)\, r_t$. As the step approaches the clean image this term dominates, and the correction amplifies the residual instead of cancelling it; for exponential integrators the same effect enters through $\lambda(k) \to \infty$. In numerical terms the sampling equation becomes stiff near the clean image. In a linear model of the denoiser, one Heun step is more accurate than one Euler step only for h/k below 2 to 3 and amplifies perturbations for h/k above 13, whereas the Euler step never does (Appendix I).

Proposition 4 (Window)**.** *Cover $[t, k]$ either by one Heun step (2 NFE) or by two Euler steps through an intermediate time m (2 NFE). If the residual varies slowly across the interval, the best subdivision costs about $\frac{1}{2}\mathrm{err}_E(t \to k)$ in the path cost of Section 3.3, and n Euler substeps divide the shared term by n. Heun wins only if its correction removes more than half of the shared term, $\|r_t + C\|^2 < \frac{1}{2}\|r_t\|^2$.*

Two conditions must hold at once. The step must be long enough for C to be comparable to r_t; on short steps C is negligible (Proposition 1) and Heun loses the factor two of subdivision. The step must also end far enough from the clean image for the correction to stay aligned (Proposition 3). Very long steps fail as well, because r_k and r_t decorrelate. A multi-stage step can therefore win only inside a window in the (h, k) plane that depends on the model through r and J, and a distilled model trained for a few Euler steps may have no window at all. Multistep methods follow the same equation with a correction built from the history velocity, $C = \alpha\,(r_h - r_t)$; they cost 1 NFE and are governed by alignment alone (Appendix A).

3.3 From local to global: an additive path cost

Every one-step error above is measured from the reference path, so the error of a step does not depend on how a path reached its start. The cost of a path is the sum of its step errors, and its minimum over paths follows exactly from the Bellman recursion $\mathrm{OPT}(t) = \min_{k,s}\left[\mathrm{err}_s(t \to k) + \mathrm{OPT}(k)\right]$ (Chen et al., 2026). Sampling does not return to the reference path after each step. To first order, its endpoint error is the sum of the one-step defects $d_i = \Phi_{s_i}(x_{t_i}^*) - x_{t_{i+1}}^*$, each carried to the end by the remaining steps, $x_{t_N} - x_0 = \sum_i P_i d_i$.

Proposition 5 (Path cost)**.** *If the remaining steps do not amplify a deviation, $\|P_i\| \le 1$, and the propagated defects are uncorrelated over the initial noise, $\mathbb{E}\langle P_i d_i, P_j d_j\rangle = 0$ for $i \ne j$, then $\mathbb{E}\|x_{t_N} - x_0\|^2 \le \sum_i \mathbb{E}\|d_i\|^2$, the expected path cost.*

The argument uses only that the reference path starts at the initial noise and ends at x_0; it does not need the reference path to be a trajectory of the sampling equation. Proposition 2 gives the first condition for Euler steps, and the second is an assumption, which we test through its consequence.

Hypothesis H. *Among paths of equal budget, the additive path cost $\sum_i \mathrm{err}_{s_i}(t_i \to t_{i+1})$ ranks paths in the direction of their image quality, so that its minimum is a good path, but it does not predict the size of quality differences.*

Predictions. The analysis predicts (P1) a window: an extra evaluation inside a step beats a further step only on steps that are short relative to their distance from the image; (P2) a budget effect: at small budgets the steps are long, so spending evaluations inside steps costs quality, and the cost shrinks as the budget grows; and (P3) placement: the steps should be placed where the model's clean-sample prediction is poor, which the Euler layer of the graph measures. Section 4 tests P1 on the one-step errors and Section 5 tests P2 and P3 on image quality.

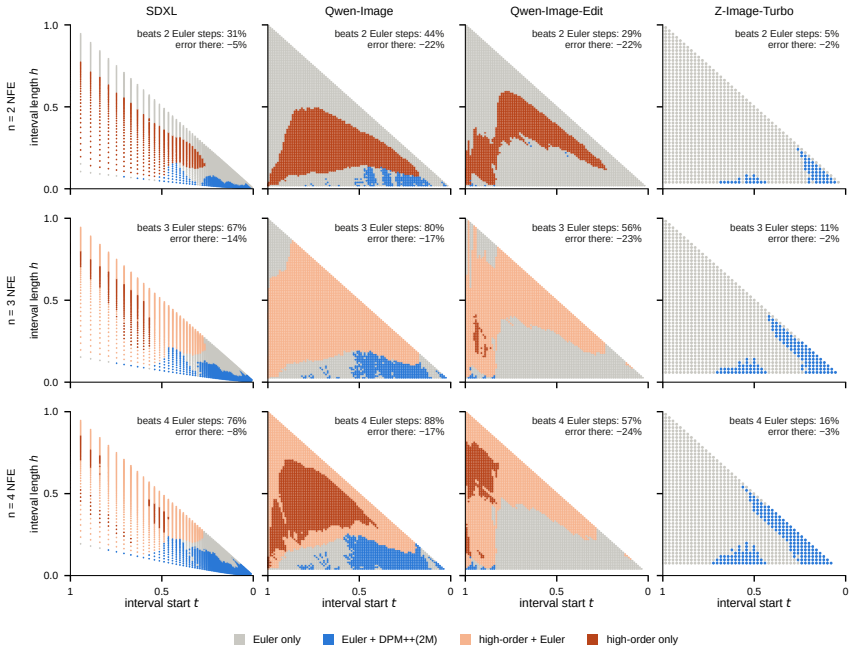


Figure 2: Best way to spend n NFE on one interval. Each dot is an interval of the time grid, placed at its start t (noise on the left) and its length h. For every interval and $n = 2, 3, 4$, all combinations of the eight solvers that use exactly n NFE inside the interval are compared, and the dot is coloured by the kind of combination with the lowest error. The interval starts without history, so a multistep step can only follow a step taken inside the interval; intervals shorter than n grid cells are excluded. Corner text: share of intervals on which the best combination beats n Euler steps, and the median error reduction on those intervals. With CFG one NFE is two network passes (one on Z-Image-Turbo).

4 The solver menu graph and the planner

4.1 Graph construction

The graph turns the one-step errors of Section 3 into a directed acyclic graph over timesteps, so that a sampler becomes a path. Nodes are grid times $1 = t_0 > t_1 > \cdots > t_N = 0$, with $N = 100$ ($N = 50$ on Z-Image-Turbo). From one calibration prompt and seed we form the reference path on the grid and cache its velocities; every pair $i < j$ and every solver s then gives an edge $(i \to j,\ s,\ \mathrm{NFE}_s,\ \mathrm{err}_s,\ \text{father})$ from one deployable step. A multistep edge depends on the previous node h, its father, and is legal only if the path reached i from h; single-step edges can follow any predecessor. Building all eight layers costs about 4.8×10^4 NFE per prompt (Table 5). The Euler layer is far cheaper: an Euler edge $i \to j$ weighs $h^2 \|r_{t_i}\|^2$ (Equation (1)), so the whole layer follows from the residuals at the 100 grid nodes, about 100 NFE beyond the reference run. On 25 431 prompts of the PickScore training set (Kirstain et al., 2023), the per-prompt residual profiles have a mean pairwise cosine similarity of 0.957 on SDXL and 0.955 on Qwen-Image (Appendix D). The Euler layer is thus a property of the model rather than of the prompt, and we build the full eight-solver graph from one calibration prompt; Sections 5.2 and 5.5 test the other layers. Averaged over 16 prompts, as many as the warm-up prompts of GITS, the Euler layer also serves as a schedule (Section 5.4).

4.2 The window on across models

Table 8 (Appendix E) lists the quantities that Propositions 1, 3, and 4 predict, measured as one-step errors on the reference path. On unit intervals, the median ratio of any solver's error to the Euler error is within 11% of one on all four models. Among intervals ending below $t = 0.05$, one Heun step has a larger error than one Euler step on 94 to 100%, and on 51 to 89% of those ending below $t = 0.2$. Proposition 3 identifies h/k as the variable of the loss of alignment, and the measured thresholds are close: on Qwen-Image, Qwen-Image-Edit, and SDXL, one Heun step never beats the best pair of Euler steps once h/k exceeds 3.1, 4.0, and 5.0, and 99% of its wins over one Euler step lie below

$h/k = 4.7$ to 6.4 (Figure 1d); on the true sampling trajectory the advantage of Heun also declines with h/k (Appendix M). With more evaluations the window narrows: RK3 and RK4 beat three and four Euler steps on at most 15% of intervals (Table 9, Appendix E). For the window we compare, on every interval, the best way to spend n NFE over all combinations of the eight solvers with n Euler steps (Figure 2). With two NFE, a combination other than two Euler steps wins on 44% of intervals on Qwen-Image, 29% on Qwen-Image-Edit, 31% on SDXL, and 5% on Z-Image-Turbo; with four NFE the shares are 88%, 57%, 76%, and 16%. Multi-stage steps win at intermediate lengths away from the clean image, and near the clean image only DPM-Solver++(2M) beats Euler.

4.3 The planner and the family of planned samplers

A sampler is a path from node 0 to node N with a solver on every edge. It is legal if every multistep edge's father is the actual predecessor; its cost is the sum of its edge errors and its budget the sum of its edge NFE. A dynamic program over states (h, i, b), the previous node, the current node, and the NFE used, finds the lowest-cost legal path for an exact budget B in $O(N^3 B)$ time, a few seconds for $N = 100$ and $B = 50$ (Appendix F).

Restricting the menu gives a family of samplers. **Euler** is the lowest-cost path over Euler edges; it optimises only the schedule. A member named after a solver S (**Heun**, **DPM-2S**, **RK3**, **DPM-3S**, **RK4**; **DPM++2M** and **UniPC-2C** for the multistep solvers) uses S on every step where S is defined. Euler serves as a bootstrap step where S needs an interior start or a history state, and as the terminal step where S has no terminal edge or the budget is not divisible; where this rule admits no path of exactly B NFE, the member takes the fewest Euler steps that do (Appendix F). **Mix** is the lowest-cost path over the full menu.

5 Experiments

5.1 Setup

We use the native pipelines of Qwen-Image (Wu et al., 2025) (1664×928, true guidance 4.0, bf16), SDXL base 1.0 (Podell et al., 2024) (1024^2, guidance 5.0, fp16), Qwen-Image-Edit (Wu et al., 2025) (true guidance 4.0), and Z-Image-Turbo (Z-Image Team, 2025) (1024^2, no guidance). On FLUX.1-dev and SD3.5-Large (1024^2, guidance 3.5), we compare ten Euler steps and five Heun steps on the native schedule with ten Euler steps on the planned schedule, all at 10 NFE (Appendix N). Text-to-image quality is measured with GenEval (Ghosh et al., 2023), 553 prompts with 4 seeds each, and with ImageReward-v1.0 (Xu et al., 2023) on the same images. Editing is measured on the 272 (source image, instruction) pairs of the FlowEdit benchmark (Kulikov et al., 2025): ImageReward against the benchmark's description of the edited image, and SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), L1, and PSNR against the 50-step output of the editing pipeline. All methods start from the same initial noise and use the same NFE; since a multi-stage step costs two to four NFE, samplers with equal budgets take different numbers of steps. **Native** is the pipeline's own scheduler with B steps, which is B NFE. Each graph is calibrated on one prompt and one seed (Section 4.1) and used unchanged on all prompts and edits.

5.2 Does the lowest path cost pick a good path?

For each budget of 5, 10, 15, and 25 NFE we draw 1000 legal paths from the full graph, generate a fixed prompt set with every path from the same initial noise, and score each path by its mean ImageReward (Figure 3, Table 13). Path cost and mean ImageReward are negatively correlated at every budget (Spearman -0.77 to -0.82 on SDXL, -0.25 to -0.34 on Qwen-Image). The lowest-cost path, which is the Mix path, scores above 94.1 to 99.9% of the random paths on SDXL and above 95.6 to 99.8% on Qwen-Image, with 50 prompts per path. Between 1 and 59 of the 999 random paths score higher, at most 14 of them significantly (paired t-test over prompts, 5% level). The lowest-cost path is thus a good path rather than the best one, as Hypothesis H states.

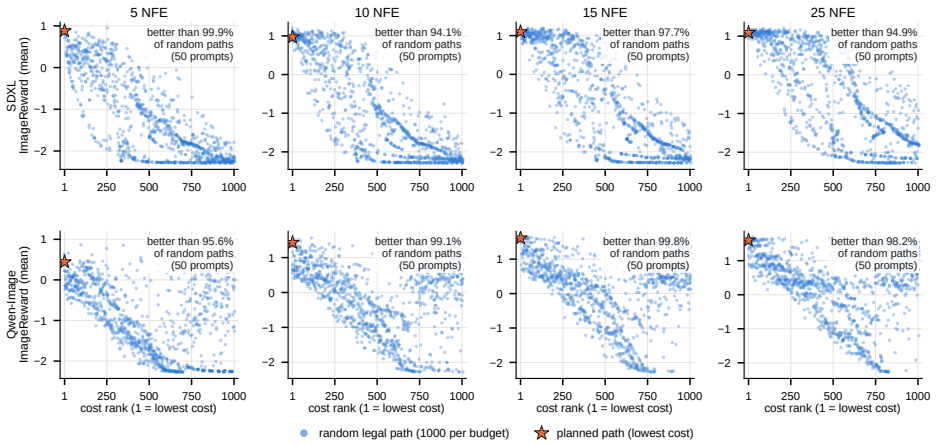


Figure 3: The lowest-cost path is a good path. For each budget (5, 10, 15, 25 NFE), 1000 random legal paths with exactly B NFE are drawn on the graph, and every prompt is generated with every path under the same seed. Each dot is one path: x = its rank by graph cost among the 1000 paths (1 = lowest cost), y = mean ImageReward over the prompts; star: the planned path. Corner: share of the 999 random paths that score below the planned path, and the number of prompts. With CFG, B NFE are $2B$ network passes.

5.3 Spending evaluations inside a step

Table 1 gives every planned sampler at 10 and 15 NFE; Figure 6 (Appendix K) shows their paths. At 10 NFE the six solvers that spend evaluations inside a step take four to seven steps and score below Euler on average, by 0.031 on SDXL and 0.090 on Qwen-Image. Heun, whose 10-NFE plan has no Euler step and must take a second-order step into the clean image, loses 0.077 and 0.110, as Proposition 3 predicts. At 15 NFE the steps are shorter, the average loss falls to 0.010 and 0.028, and Heun, whose last step is now Euler, scores 0.009 and 0.004 above Euler. The exception follows the window: on SDXL the DPM-2S plan places its four second-order steps between $t = 0.89$ and $t = 0.11$, inside the SDXL window (Table 8), and scores highest at 10 NFE (0.504); on Qwen-Image, whose window is confined to short steps, the same solver scores lowest. DPM-Solver++(2M), which raises the order without extra evaluations, stays within 0.007 of Euler at both budgets. On FLUX.1-dev and SD3.5-Large, five Heun steps on the native schedule score 0.575 and 0.628 against 0.626 and 0.676 for ten native Euler steps, with the largest losses on two-object prompts and attribute binding (Table 15).

5.4 Placing the steps

The Euler member takes as many steps as native and differs only in where they fall. At 10 NFE it raises GenEval over native by 0.107 on Qwen-Image and 0.030 on SDXL, and by 0.018 and 0.007 on FLUX.1-dev and SD3.5-Large (Figure 1c); at 7 NFE the relative gains are 24.8% and 22.7% on Qwen-Image and SDXL. At 15 NFE Euler and Mix both exceed native (0.846 and 0.850 against 0.818 on Qwen-Image, 0.514 and 0.521 against 0.501 on SDXL), and at 25 NFE Mix reaches 0.854 and 0.544. The native pipeline at 50 steps scores 0.849 and 0.544, so the planned sampler matches it with 30% of its NFE on Qwen-Image and half on SDXL. The full graph adds little over its own Euler layer; most of the gain over native comes from placement. The Euler layer of the graph is a schedule optimiser on its own, and Table 2 compares it with dedicated ones on SDXL at 10 NFE. Run with DPM-Solver++(2M), the planned schedule averaged over 16 prompts scores 0.533, against 0.516 for Align Your Steps (Sabour et al., 2024) and 0.485 for the Karras sigmas of diffusers (Karras et al., 2022; von Platen et al., 2022). With Euler, it scores 0.519 against 0.508 for Align Your Steps and 0.510 for GITS (Chen et al., 2024).

5.5 Transfer within a model family

Qwen-Image-Edit and Qwen-Image share a backbone family, and their graphs agree closely. Over the intervals they share, the edge errors of each solver have Spearman correlations above 0.97, the log

Table 1: GenEval of the planned samplers at 10 and 15 NFE: native scheduler, Euler on the planned schedule, the multistep DPM-Solver++(2M), which raises the order without extra evaluations, and the six solvers that spend evaluations inside a step, each planned on the model's graph (Section 4.3). Last row: mean difference of these six from Euler at the same NFE. With CFG one NFE is two network passes.

	NFE per	SDXL		Qwen-Image	
Sampler	step	10 NFE	15 NFE	10 NFE	15 NFE
native	1	0.467	0.501	0.723	0.818
Euler	1	0.497	0.514	**0.830**	0.846
DPM-Solver++(2M)	1	0.503	0.516	0.823	0.841
UniPC-2C	2	0.471	0.500	0.770	0.788
Heun	2	0.420	**0.523**	0.720	**0.850**
DPM-Solver-2S	2	**0.504**	0.507	0.688	0.828
RK3	3	0.453	0.486	0.730	0.825
DPM-Solver-3S	3	0.463	0.508	0.765	0.781
RK4	4	0.488	0.500	0.769	0.836
Mix (full menu)	–	0.499	0.521	0.792	**0.850**
mean of the six, minus Euler		−0.031	−0.010	−0.090	−0.028

Table 2: Timestep schedules on SDXL at 10 NFE (GenEval; one NFE per step, two network passes with classifier-free guidance). Planned, Euler layer: lowest-cost path of the Euler layer, run with the listed solver; DPM++(2M) member: planned on the DPM-Solver++(2M) layer (Table 1). Calibration: reference samples (planned), sampled trajectories (GITS, our re-implementation, Appendix D), or data samples (Align Your Steps, as reported by Sabour et al. (2024)). Native and Karras schedules are hand-designed; the 50-step rows are references.

Schedule	Solver	Calibration	NFE	GenEval
native	Euler	–	50	0.544
native	DPM-Solver++(2M)	–	50	0.520
native	Euler	–	10	0.467
planned, Euler layer	Euler	1 reference sample	10	0.497
Align Your Steps	Euler	8192 samples	10	0.508
GITS	Euler	16 trajectories	10	0.510
planned, Euler layer	Euler	16 reference samples	10	**0.519**
Karras sigmas (diffusers)	DPM-Solver++(2M)	–	10	0.485
Align Your Steps	DPM-Solver++(2M)	8192 samples	10	0.516
planned, DPM++(2M) member	DPM-Solver++(2M), Euler	1 reference sample	10	0.503
planned, Euler layer	DPM-Solver++(2M)	16 reference samples	10	**0.533**

residual profiles have a correlation of 0.84, 72% of the Heun window of Qwen-Image-Edit lies inside that of Qwen-Image, and the cheaper of one Heun step and one Euler step is the same on 96.6% of intervals. At 10 NFE the Mix path has four Heun steps on Qwen-Image and two Heun steps and one RK3 step on the editing model; both begin and end with Euler. We therefore run paths planned on the Qwen-Image graph on the editing model without recalibration, on the 272 FlowEdit edits at 10 NFE (Table 3). Both transferred paths land closer to the 50-step output than native (LPIPS 0.085 and 0.088 against 0.117) and score a higher ImageReward (1.007 and 1.038 against 0.981; 1.055 at 50 steps). Paths planned on the editing graph itself reach LPIPS 0.092 (Euler) and 0.072 (Mix). Figure 8 (Appendix Q) shows edits of four types.

5.6 The distilled model

Z-Image-Turbo is distilled to sample in 8 steps. Its window is almost empty: two Euler steps are the cheapest use of two NFE on 95% of intervals, and the other winners lower the error by 2 to 3%. Mix is pure Euler up to 13 NFE (Figure 5, Appendix J), so the local measurements alone recover that this model should be integrated with first-order steps. With two Euler steps at $t = 1 \to 0.76 \to 0$, the planned sampler scores GenEval 0.762 at 2 NFE, as high as the native scheduler at 4 steps (0.762)

Table 3: Qwen-Image-Edit, 272 FlowEdit edits at 10 NFE (20 network passes with classifier-free guidance). Paths planned on the Qwen-Image graph run on the editing model without recalibration. Path: the steps of the sampler from noise to image (E Euler, H Heun, RK3; $\times k$ marks k consecutive steps of the same solver). SSIM, LPIPS, L1, and PSNR are measured against the native 50-step edit.

Sampler	Path	ImageReward ↑	SSIM ↑	LPIPS ↓	L1 ↓	PSNR ↑
native	native schedule, E×10	0.981	0.862	0.117	0.041	23.6
Euler, Edit graph	E×10	1.015	0.885	0.092	0.030	25.7
Mix, Edit graph	E·H·RK3·E·H·E (6 steps)	1.014	**0.904**	**0.072**	**0.025**	**26.1**
Euler, Qwen-Image graph	E×10	1.007	0.892	0.085	0.028	**26.1**
Mix, Qwen-Image graph	E·H×4·E (6 steps)	**1.038**	0.888	0.088	0.033	24.8
native, 50 steps	E×50	1.055	–	–	–	–

and above it at its default 8 steps (0.757); at 4 and 8 NFE the planned and native schedules are within 0.003 (Table 16).

6 Related work

Fast samplers with a fixed solver. DDIM (Song et al., 2021a) and PNDM (Liu et al., 2022) showed that the probability-flow ODE (Song et al., 2021b) can be integrated in tens of steps; DPM-Solver and DPM-Solver++ (Lu et al., 2022; 2025), DEIS (Zhang & Chen, 2023), UniPC (Zhao et al., 2023), and DPM-Solver-v3 (Zheng et al., 2023) raised the order with exponential integrators and multistep formulas, and EDM (Karras et al., 2022) recommended Heun with a tuned schedule.

Graph formulations of sampling. Selecting timesteps by dynamic programming over a grid goes back to Watson et al. (2021), who choose the DDPM schedule that maximises the ELBO. BOSS (Nguyen et al., 2024) finds the step sizes of a flow-matching model by a Bellman recursion and then fine-tunes the model to them, and OSS (Pei et al., 2025) distils step sizes from a reference trajectory by dynamic programming. Graphs over timesteps have also been used to cast denoising as a shortest-path problem trained into the model (Chen et al., 2025), and to place feature caching (Gao et al., 2025). These works choose which timesteps to visit, or which computations to reuse, for a fixed update rule. The closest work, S^3 (Liu et al., 2024), searches a per-step solver strategy among exponential integrators, guided by a predictor of the FID of sampled configurations. We instead measure one-step errors on a reference path, which explains where an extra evaluation pays off, and plan by an exact-budget dynamic program without searching over generated samples.

Schedule optimisation and learned solvers. Align Your Steps (Sabour et al., 2024) and GITS (Chen et al., 2024), which also solves a dynamic program on a timestep grid, optimise the timesteps for a given solver. The Euler layer of our graph belongs to this line (Table 2), and the planner adds the choice of solver. Bespoke solvers (Shaul et al., 2024), AMED (Zhou et al., 2024), learned samplers (Watson et al., 2022), and distillation (Salimans & Ho, 2022; Song et al., 2023) learn coefficients, times, or a student model and reach smaller budgets at the price of training; our planner needs no training. Training-free methods that lower the cost of each evaluation, by reusing cached features (Gao et al., 2025) or by updating only a sparse set of tokens (Sun et al., 2026), are complementary.

7 Conclusion

Euler steps are non-expansive whenever the model's prediction is, while a multi-stage correction loses accuracy as the step grows relative to its distance from the clean image, where the sampling equation is stiff. On the graphs of the three undistilled models a Heun step beats two Euler steps only while $h/k < 5$. Evaluations spent inside steps cost GenEval at 10 NFE on four text-to-image models and less at 15 NFE on SDXL and Qwen-Image, while placing Euler steps by the model's own prediction error raises the score on all four. In few-step sampling, evaluations are thus best spent first on placing first-order steps, and inside a step only where h/k is small.

REPRODUCIBILITY STATEMENT

Section 2 and Appendices A–D define the reference path, the solvers and their NFE accounting, the grids, and the calibration settings; Appendix F gives the planner, and Section 5.1 and Appendix L the evaluation protocol. Graph files, planned paths, and code will be released.

AI USE STATEMENT

We used a large language model to polish the language of the manuscript, to help check derivations, to write the evaluation code, and to write the code that draws the figures. The authors checked all code and results. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Black Forest Labs. FLUX. `https://github.com/black-forest-labs/flux`, 2024.

Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ODE-based diffusion sampling. In *International Conference on Machine Learning*, 2024.

Ping Chen, Xingpeng Zhang, Zhaoxiang Liu, Huan Hu, Xiang Liu, Kai Wang, Min Wang, Yanlin Qian, and Shiguo Lian. Optimizing for the shortest path in denoising diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Ping Chen, Xiang Liu, Xingpeng Zhang, Fei Shen, Xun Gong, Zhaoxiang Liu, Zezhou Chen, Huan Hu, Kai Wang, and Shiguo Lian. Chain-of-trajectories: Unlocking the intrinsic generative optimality of diffusion models via graph-theoretic planning. *arXiv preprint arXiv:2603.14704*, 2026.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Huanlin Gao, Ping Chen, Fuyuan Shi, Chao Tan, Zhaoxiang Liu, Fang Zhao, Kai Wang, and Shiguo Lian. LeMiCa: Lexicographic minimax path caching for efficient diffusion-based video generation. In *Advances in Neural Information Processing Systems*, 2025.

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

Ernst Hairer, Syvert P. Nørsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, 2 edition, 1993.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-Pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *IEEE/CVF International Conference on Computer Vision*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Enshu Liu, Xuefei Ning, Huazhong Yang, and Yu Wang. A unified sampling framework for solver searching of diffusion probabilistic models. In *International Conference on Learning Representations*, 2024.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022.

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

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, 22 (4):730–751, 2025.

Bao Nguyen, Binh Nguyen, and Viet Anh Nguyen. Bellman optimal stepsize straightening of flow-matching models. In *International Conference on Learning Representations*, 2024.

Jianning Pei, Han Hu, and Shuyang Gu. Optimal stepsize for diffusion sampling. *arXiv preprint arXiv:2503.21774*, 2025.

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

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your steps: Optimizing sampling schedules in diffusion models. In *International Conference on Machine Learning*, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

Neta Shaul, Juan Perez, Ricky T. Q. Chen, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke solvers for generative flow models. In *International Conference on Learning Representations*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, 2023.

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

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. `https://github.com/huggingface/diffusers`, 2022.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *International Conference on Learning Representations*, 2022.

Chenfei Wu, Jiahao Li, Jingren Zhou, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023.

Z-Image Team. Z-Image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Wenliang Zhao, Lujia Bai, Yongming Rao, Jie Zhou, and Jiwen Lu. UniPC: A unified predictor-corrector framework for fast sampling of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. DPM-Solver-v3: Improved diffusion ODE solver with empirical model statistics. In *Advances in Neural Information Processing Systems*, 2023.

Zhenyu Zhou, Defang Chen, Can Wang, and Chun Chen. Fast ODE-based sampling for diffusion models in around 5 steps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

A SOLVER MENU, UPDATES, AND NFE ACCOUNTING

Table 4: The solver menu. NFE is the number of velocity evaluations of one deployable step (each is two network passes on the models that use classifier-free guidance: SDXL, Qwen-Image, Qwen-Image-Edit); history lists the additional state a step needs from the path.

Solver	Family	Order	NFE	History
Euler	single-step	1	1	none
Heun	single-step, 2 stages	2	2	none
DPM-Solver-2S	exponential, 2 stages	2	2	none
RK3	single-step, 3 stages	3	3	none
DPM-Solver-3S	exponential, 3 stages	3	3	none
RK4	single-step, 4 stages	4	4	none
DPM-Solver++(2M)	multistep	2	1	previous velocity
UniPC-2 (closed)[a]	predictor–corrector	2	2	previous velocity, equal step

[a]Corrector evaluation discarded after the step; the reference implementation reuses it and counts 1 NFE per step (Appendix B).

All solvers act on the graph coordinate t of Section 2 and start a step on $[t, k]$, $h = t - k$, from the cached first velocity $v_t = v_\theta(x_t, t)$, so that the cost of a step is one evaluation plus the extra stages listed here. Write $\Delta = k - t = -h$.

Euler. $x_k = x_t + \Delta v_t$; 0 extra evaluations.

Heun. $\tilde{x} = x_t + \Delta v_t$, $x_k = x_t + \frac{\Delta}{2}\big(v_t + v_\theta(\tilde{x}, k)\big)$; 1 extra.

RK3 (Kutta). $v_2 = v_\theta(x_t + \frac{\Delta}{2} v_t,\ t + \frac{\Delta}{2})$, $v_3 = v_\theta(x_t + \Delta(-v_t + 2v_2),\ k)$, $x_k = x_t + \frac{\Delta}{6}(v_t + 4v_2 + v_3)$; 2 extra.

RK4. The classical tableau with stages at $t + \Delta/2$ (twice) and k; 3 extra.

DPM-Solver-2S and 3S. The single-step exponential integrators of Lu et al. (2022) in the ε-parameterisation, with stages at the λ-midpoint (2S) or at one third and two thirds of the λ-interval (3S). On the flow path $\alpha_t = 1 - t$, $\sigma_t = t$, the first-order exponential step coincides with Euler, and we use that identity to form the stage states without cancellation between $O(1/\alpha_t)$ terms. For SDXL the same formulas are applied with the model's $\bar{\alpha}_t$. Because λ is singular at the endpoints, these edges are constructed only on interior intervals ($0 < k < t < 1$ for the flow models; $k > 0$ for SDXL, whose $t = 1$ has finite λ). 1 and 2 extra evaluations.

DPM-Solver++(2M). With history velocity v_h at the father time $h > t$, the x_0-prediction $\hat{x}_0(x, v, t) = x - tv$ (flow) and the ratio $\rho = (\lambda_h - \lambda_t)/(\lambda_k - \lambda_t)$, the update is

$$x_k = \frac{\sigma_k}{\sigma_t} x_t - \alpha_k \, \varphi_1(-H)\Big(\hat{x}_0(x_t, v_t, t) + \tfrac{1}{2\rho}\big(\hat{x}_0(x_h, v_h, h) - \hat{x}_0(x_t, v_t, t)\big)\Big),$$

with $H = \lambda_k - \lambda_t$ and $\varphi_1(z) = e^z - 1$, which is DPM-Solver++(2M) of Lu et al. (2025); the order-2 UniPC predictor with B(h) coefficients (Zhao et al., 2023) is algebraically the same update, so it is not listed separately. 0 extra evaluations. On the reference states the error of this update is a quadratic form in the residuals r_t and r_h with coefficients fixed by (h, t, k), which is why every father can be tabulated from the cached velocities at no cost.

UniPC-2, closed. Predict $\tilde{x}_k$ with the update above, evaluate $\tilde{v}_k = v_\theta(\tilde{x}_k, k)$, and correct with the order-2 UniC formula using $\hat{x}_0(\tilde{x}_k, \tilde{v}_k, k)$; the corrector evaluation is then discarded (Appendix B). 1 extra evaluation, and a father at equal step ratio.

In the first-order reading of Section 3, the multistep correction is $C = \alpha(r_h - r_t)$ with $\alpha = \alpha(\rho, H)$ from the coefficients above, so the multistep methods are governed by the alignment of the history residual with the current one and never by the equal-cost competition with subdivision.

B Cost model of the closed predictor–corrector

The reference UniPC implementation evaluates the network at the corrected state and reuses that evaluation as the first velocity of the next step, so it reports one evaluation per step. In that form the evaluation used to correct step n is made at a state that step n has already left, and the next step starts from a velocity that was computed at the uncorrected prediction. Measured as a deployable step from the reference state, that step is a different object on every interval, because its first velocity depends on the previous prediction. We therefore measure the closed form: the corrector evaluation belongs to the step it corrects and the next step evaluates afresh at its own start. This costs 2 NFE per step, one more than the reference implementation, and is the form used by the graph and by the UniPC-2C member of Table 10. The stateful form is a different sampler whose steps are not independent edges, so it is not in the graph.

C Setting and graph construction details

Generative ODE and graph time. All four models are sampled by integrating $\mathrm{d}x/\mathrm{d}t = v_\theta(x, t)$ from $t = 1$ (noise) to $t = 0$ (clean image). For the flow models (Qwen-Image, Qwen-Image-Edit, Z-Image-Turbo) the training path is $x_t = (1-t)x_0 + t\varepsilon$ and v_θ predicts $\varepsilon - x_0$. For SDXL, a variance-preserving model with an ε-prediction network, we use $x_t = x_0 + t\,\sigma_{\max}\varepsilon$ with $t = \sigma/\sigma_{\max} \in [0, 1]$, so the velocity is $\sigma_{\max}\varepsilon_\theta$. Exponential integrators (the DPM-Solver and UniPC families) use the half-log-SNR $\lambda(t)$ of each model, which diverges at both endpoints; they are therefore defined on interior intervals only, and Euler, Heun, and the Runge–Kutta methods cover the endpoints. Classifier-free guidance is applied inside v_θ, so one NFE runs the network with and without the condition: batched into one call on SDXL and as two calls on the Qwen models. Z-Image-Turbo is sampled without guidance.

Reference solution. For a fixed noise ε, every sampler approximates the exact solution $x(0)$ of the ODE. A consistent method of order p has global error $O(h^p)$ as the step length $h \to 0$ (Hairer et al., 1993), so the native pipeline at its fine schedule (50 steps; 8 on Z-Image-Turbo, its native setting) gives the closest available approximation, which we take as the reference sample; its forward path is

the reference path of Section 2. Because all deployable steps start on the same reference path, errors of different solvers on the same interval are comparable, and the cost of a path is the sum of the errors of its steps.

Graph construction. The grid has $N = 100$ intervals for Qwen-Image, Qwen-Image-Edit, and SDXL and $N = 50$ for Z-Image-Turbo. The flow grids are uniform in t. The SDXL grid is uniform in the training timestep and mapped to $t = \sigma/\sigma_{\max}$, so the network sees the timesteps of the native pipeline; it is dense near the clean image, and 3572 of its 4950 non-unit intervals end below $t = 0.1$. For each model the graph is built once, in four steps.

1. *Reference.* Run the native pipeline at its fine schedule from one calibration prompt and seed, keep the noise ε and the output x_0, and form the reference states $x^*_{t_i}$ on the grid (Section 2).
2. *Velocities.* Evaluate and cache $v_\theta(x^*_{t_i}, t_i)$ at all $N+1$ reference states.
3. *Edges.* For every pair $i < j$ and every solver s, apply one deployable step of s from $x^*_{t_i}$ to t_j and record the edge $(i \to j,\ s,\ \mathrm{NFE}_s,\ w,\ \text{father})$ with error $w = \|\Phi_s(x^*_{t_i}; t_i \to t_j) - x^*_{t_j}\|^2$ (mean over entries). Euler edges follow from the cached velocities through Equation (1); multi-stage edges run their extra stages from the cached first velocity; exponential integrators are built on interior intervals only.
4. *History.* A multistep edge depends on the previous node h, so DPM-Solver++(2M) gets one edge per father $h < i$, each computed from two cached velocities at no extra cost; closed UniPC edges are measured at equal step ratio and evaluate once more at the predicted endpoint.

Single-step edges have no father and can follow any predecessor. A multistep edge with father h is legal only if the path reached i from h, because its coefficients and history velocity belong to that step; a multistep solver therefore cannot be laid over an independently optimised schedule. On the 100-interval grids construction takes about 4.8×10^4 NFE, the cost of about a thousand 50-step samples (Appendix D).

D Calibration settings and graph cost

Table 6 lists the calibration trajectory of each graph and Table 7 the number of edges and network evaluations per solver. Euler edges and the DPM-Solver++(2M) edges cost nothing beyond the cached velocities. The reference of Z-Image-Turbo is its native 8-step sample. The closed UniPC layer is measured at equal step ratio only ($h = t + (t - k)$); the DPM-Solver++(2M) layer keeps every father, so the planner can pick the history freely. Equal step ratio is the setting in which multistep methods are deployed on a smoothly varying schedule.

Multi-prompt calibration of the Euler layer. The corpus consists of 25 431 prompts of the PickScore training set (Kirstain et al., 2023). Per prompt, the Euler layer needs the native 50-step reference (50 NFE) and the velocity at the 100 grid nodes where an Euler edge starts (100 NFE); the other seven layers are not recomputed. Each prompt gives a 100-dimensional vector $(\|r_{t_0}\|^2, \ldots, \|r_{t_{99}}\|^2)$, which determines its Euler layer; we report the mean cosine similarity over all pairs of prompts (SDXL 0.957, s.d. 0.03; Qwen-Image 0.955, s.d. 0.04). Cosine similarity is the matching measure because rescaling the vector leaves the lowest-cost Euler path unchanged. The schedule of Table 2 is the lowest-cost Euler path on the mean vector of 16 prompts drawn from the corpus, as many as the warm-up prompts of GITS.

GITS baseline. We re-implement GITS (Chen et al., 2024) with an Euler base solver: the native Euler sampler with 100 steps generates teacher trajectories for 16 warm-up prompts, the cost of a coarse step from node i to node j is the mean squared distance between the Euler step from the teacher state at i and the teacher state at j, and a dynamic program selects the K-step schedule of lowest total cost. Sampling uses the native Euler scheduler on the selected timesteps with the GenEval prompts and seeds of all other runs.

Compute. All experiments run on 8 NVIDIA H100 GPUs; graph edges are independent given the cached reference velocities and are measured in parallel across GPUs. Table 5 lists the cost of one

full graph in NFE and in H100 GPU hours. The time of one NFE is the slope of generation time against NFE over the random-path runs of Appendix L, which removes the fixed cost of text encoding and decoding per image.

Table 5: Cost of one full graph (all eight solver layers from one prompt, including the reference run). Time of one NFE on one H100, measured on the random-path runs; one NFE includes both classifier-free guidance passes. †Time of one image divided by its NFE (Qwen-Image-Edit: 16.92 s at 5 NFE; Z-Image-Turbo: 1.1 s at 2 NFE), which includes the fixed cost of encoding and decoding and bounds the time of one NFE from above. GPU hours on one H100; on 8 H100 GPUs the wall-clock time is one eighth.

Model	s / NFE	NFE	GPU h
SDXL	0.039	4.8×10^4	0.5
Qwen-Image	0.63	4.8×10^4	8.3
Qwen-Image-Edit	3.4†	4.8×10^4	45
Z-Image-Turbo	0.55†	1.2×10^4	1.8

Table 6: Calibration trajectory of each graph. One prompt and one seed per model; the graph is then used on all evaluation prompts without refitting. With classifier-free guidance one NFE is two network passes; Z-Image-Turbo runs without guidance, one pass per NFE.

Model	Grid	Reference	Resolution, guidance	Prompt / input
Qwen-Image	100 uniform	50 steps, seed 42	1664×928, true CFG 4.0	a fixed Chinese prompt (a researcher writing on a glass wall)
Qwen-Image-Edit	100 uniform	50 steps, seed 42	source size, true CFG 4.0	FlowEdit `gas_station`, "Change the sign text from 'CAFE' to 'ICLR'."
SDXL	101 training-timestep nodes	50 steps, seed 42	1024×1024, CFG 5.0	"A cinematic photo of a young researcher writing equations on a glass wall in a futuristic data center."
Z-Image-Turbo	50 uniform	8 steps, seed 42	1024×1024, no CFG	"a peony flower."

Table 7: Edges and velocity evaluations (NFE) per solver layer on a 100-interval grid (SDXL; Qwen-Image and Qwen-Image-Edit have the same counts up to the endpoint exclusions of the exponential solvers) and on the 50-interval Z-Image-Turbo grid. Evaluations exclude the cached first velocity of each edge. Each NFE is two network passes on SDXL and the Qwen models (classifier-free guidance) and one on Z-Image-Turbo.

	100 intervals (SDXL)		50 intervals (Z-Image-Turbo)	
Solver	edges	evaluations	edges	evaluations
Truth velocities	101	101	51	51
Euler	5050	0	1275	0
Heun	5050	5050	1275	1275
DPM-Solver-2S	4950	4950	1176	1176
RK3	5050	10100	1275	2550
DPM-Solver-3S	4950	9900	1176	2352
RK4	5050	15150	1275	3825
DPM-Solver++(2M) (all fathers)	161700	0	18424	0
UniPC-2 closed (equal ratio)	2450	2450	576	576
Total evaluations		47701		11805

E THE THREE EFFECTS IN NUMBERS

Table 8 collects the per-model quantities quoted in Section 4.2; the window columns are the corner numbers of Figure 2. Restricted to one Heun step against the best pair of Euler steps, the window

Table 8: The three predicted effects on the four graphs. Unit intervals: the multi-stage solver whose median error ratio to Euler on one-cell intervals is farthest from one. Near the image: share of intervals ending at $k \leq$ the threshold on which one Heun step has a larger error than one Euler step. Window: share of intervals on which the best way to spend n NFE beats n Euler steps (Figure 2). One NFE is one guided velocity evaluation: two network passes on SDXL, Qwen-Image, and Qwen-Image-Edit, which use classifier-free guidance, and one on Z-Image-Turbo.

	Unit intervals		Near the image: Heun worse			Window: beats n Euler steps		
Model	solver	ratio	$k \leq 0.05$	$k \leq 0.10$	$k \leq 0.20$	$n = 2$	$n = 3$	$n = 4$
Qwen-Image	Heun	0.92	94%	78%	51%	44%	80%	88%
Qwen-Image-Edit	RK4	0.92	100%	88%	53%	29%	56%	57%
SDXL	DPM-3S	1.01	100%	85%	69%	31%	67%	76%
Z-Image-Turbo	DPM-2S	1.02	99%	98%	89%	5%	11%	16%

covers 39%, 29%, 15%, and 0% of the non-unit intervals of Qwen-Image, Qwen-Image-Edit, SDXL, and Z-Image-Turbo, and only 6%, 2%, 1.5%, and 0% of those ending below $t = 0.1$. Table 9 extends this comparison to RK3 and RK4: at three and four NFE the multi-stage step wins on fewer intervals than Heun does at two.

Table 9: One multi-stage step against the best subdivision of the same interval into as many Euler steps as it uses NFE, on the one-step error measured on the reference path. Share: intervals of at least n grid cells on which the multi-stage step has the smaller error. Largest h/k: the largest ratio of step length to the distance of the step's end to the clean image at which it does. One NFE is one guided velocity evaluation: two network passes with classifier-free guidance, one on Z-Image-Turbo.

	Heun vs. 2 Euler		RK3 vs. 3 Euler		RK4 vs. 4 Euler	
Model	share	largest h/k	share	largest h/k	share	largest h/k
Qwen-Image	39.2%	3.1	4.1%	2.0	0.0%	–
Qwen-Image-Edit	28.9%	4.0	5.3%	3.6	4.8%	7.0
SDXL	15.3%	5.0	14.4%	8.7	7.4%	7.5
Z-Image-Turbo	0.0%	–	0.0%	–	0.0%	–

F PLANNER

Algorithm 1 gives the exact-budget dynamic program. Every member of the planned family runs the same recursion with a restricted menu and a phase attached to the state. For a member named after a solver S the path has three phases: a start phase in which Euler steps may be taken as bootstrap (at most one, or two for the multistep solvers on the flow graphs, whose multistep edges exclude a father at $t = 1$) or S may start directly where it is defined; a body phase in which only S is admissible; and a single terminal Euler step into $t = 0$, admissible once at least one step of S has been taken. For Heun and the Runge–Kutta methods, which are defined on every interval, Euler is admitted only when the budget is not a multiple of the solver's NFE per step. The Euler counts of Table 10 follow from this rule and the budget: at $B = 10$ on SDXL, DPM-Solver-2S and closed UniPC take four steps and two Euler steps, DPM-Solver-3S and RK3 take three steps and one Euler step, RK4 takes two steps and two Euler steps, DPM-Solver++(2M) takes eight steps after one bootstrap and before one terminal Euler step, and Heun divides 10 exactly and is pure.

On the flow graphs the exponential and multistep edges exclude both $t = 1$ and $k = 0$, so under this rule a DPM-Solver-3S path needs $2 + 3m$ evaluations and a closed UniPC path an odd number. Where the rule admits no path of exactly B NFE we relax it: the member takes the fewest Euler steps that make the budget exact, placed anywhere on the path, and the planner minimises cost over those placements. On Qwen-Image at 10 NFE this gives three Euler steps, two DPM-Solver-3S steps, and a terminal Euler step (6 steps, 4 of them Euler) for DPM-3S, and three Euler steps, three UniPC-2C steps, and a terminal Euler step (7 steps, 4 of them Euler) for UniPC-2C. At 15 NFE the relaxed rule is used for UniPC-2C, DPM-Solver-3S, and RK4 on SDXL and for DPM-Solver-2S, DPM-Solver-3S, and RK4 on Qwen-Image. The planner then minimises graph cost over all schedules compatible with these counts; Figure 6 shows the resulting paths at 15 NFE on Qwen-Image and SDXL.

Algorithm 1 Exact-NFE shortest path with father constraint

```
cost[(−1, 0, 0)] ← 0; all other states ∞
for i = 0, . . . , N − 1 do
    for each state (h, i, b) with finite cost do
        for each edge e = (i → j, s, NFE_s, w, father) with father ∈ {−1, h} and b + NFE_s ≤ B do
            relax cost[(i, j, b + NFE_s)] ← min(·, cost[(h, i, b)] + w) and record the parent
        end for
    end for
end for
return the cheapest state (·, N, B) and its path
```

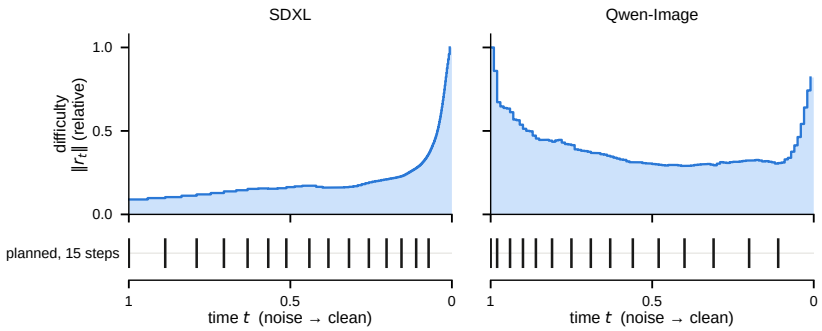


Figure 4: How hard each moment of the trajectory is, and where 15 steps are placed. Top: residual $\|r_t\|$, the distance between the model's velocity and the slope of the straight noise-to-reference path, normalised by its maximum; one Euler step of length h costs $h^2\|r_t\|^2$. Bottom: start times of the 15 steps of the lowest-cost Euler path on the graph.

G Residual profile and step placement

For Euler steps the cost of a step is $h^2\|r_t\|^2$ (Equation (1)), so an Euler schedule of fixed length should place its steps densely where $\|r_t\|$ is large. Figure 4 shows the residual profile and the 15 steps of the lowest-cost Euler path.

H Derivations for Section 3

Decomposition. On the reference path x_t^* the exact velocity is $v^* = \varepsilon - x_0$ (flow) or $\sigma_{\max}\varepsilon$ (SDXL), constant in t, and $x_k^* = x_t^* + \Delta v^*$. With $v_\theta(x_t^*, t) = v^* + r_t$, Euler lands at $x_t^* + \Delta(v^* + r_t) = x_k^* + \Delta r_t$, which gives Equation (1). A consistent single-step method advances by $\Delta\sum_i b_i v_i$ with $\sum_i b_i = 1$ and $v_1 = v_\theta(x_t^*, t)$, so it lands at $x_k^* + \Delta r_t + \Delta\sum_i b_i(v_i - v_1)$, and $C_s = \sum_i b_i(v_i - v_1)$. Equation (2) follows by expanding the square.

Proposition 1. Each stage state is $x_t^* + \Delta\sum_j a_{ij}v_j$ at time $t + c_i\Delta$. By induction over the stages, $\|x_i - x_t^*\| \le c\,h\,(\|v_1\| + \max_j \|v_j - v_1\|)$ and hence, using the Lipschitz bound in (x, t), $\|v_i - v_1\| \le L\,(\|x_i - x_t^*\| + c_i h) \le c_s L h(1 + \|v_1\|)$ for h below a threshold set by L and the tableau. The ratio bound follows from Equation (2).

Proposition 3. The Heun stage is evaluated at $x_k^* + \Delta r_t = x_k^* - hr_t$ and time k. Expanding v_θ around (x_k^*, k) in the state, $v_\theta(x_k^* - hr_t, k) = v^* + r_k - hJ_k r_t + O(h^2)$, so $C = \frac{1}{2}(r_k - r_t) - \frac{1}{2}hJ_k r_t + O(h^2)$ and $r_t + C = \frac{1}{2}(r_t + r_k - hJ_k r_t) + O(h^2)$. Comparing $\mathrm{err}_{\mathrm{Heun}} = h^2\|r_t + C\|^2$ with $\mathrm{err}_E = h^2\|r_t\|^2$ gives the stated condition. For the scaling of J_k: writing the velocity through the clean-image prediction, $v_\theta(x, k) = (x - \hat{x}_0(x, k))/k$ on the flow path (up to the sign convention), $\partial_x v_\theta = (I - \partial_x \hat{x}_0)/k$, and $\partial_x \hat{x}_0$ is bounded near the data, so $\|J_k\|$ grows like $1/k$. The same growth appears for exponential integrators through the coefficients $\varphi_1(\lambda_k - \lambda_t)$, since $\lambda_k \to \infty$ as $k \to 0$.

Proposition 4. Two Euler steps through m cost $(t-m)^2\|r_t\|^2 + (m-k)^2\|r_m\|^2$. If $\|r_m\| \approx \|r_t\|$ on the interval the minimum over m is at the midpoint and equals $\frac{1}{2}h^2\|r_t\|^2$; with n substeps of equal length the cost is $h^2\|r\|^2/n$. Heun wins against the best pair only if $h^2\|r_t + C\|^2 < \frac{1}{2}h^2\|r_t\|^2$. Since $\|C\| = O(h)$ by Proposition 1, this fails for small h; since $\|r_t + C\|$ contains the term $\frac{1}{2}hJ_kr_t$ of size $O(h/k)\|r_t\|$ by Proposition 3, it fails for k small relative to h; and for large h the residuals at the two ends decorrelate, so $\|r_t + r_k\|$ approaches $\sqrt{2}\|r_t\|$ and the condition fails again. The winning set is therefore bounded in h from both sides and bounded away from $k = 0$.

I THEORY OF THE PATH COST AND OF THE WINDOW

Euler error and the clean-sample prediction. On a flow model $v_\theta(x,t) = \big(x - \hat{x}_0(x)\big)/t$, and on the reference path $\varepsilon - x_0 = (x_t^* - x_0)/t$. Hence $r_t = -\big(\hat{x}_0(x_t^*) - x_0\big)/t$, and an Euler step of length h from x_t^* lands at $x_k^* - h\,r_t$, so that $\mathrm{err}_E(t \to k) = (h/t)^2\|\hat{x}_0(x_t^*) - x_0\|^2$. The same holds on SDXL with $x_t^* = x_0 + t\sigma_{\max}\varepsilon$ and $v_\theta = \sigma_{\max}\varepsilon_\theta$.

Proof of Proposition 2. $\Phi_E(x) = x - (t-k)\,v_\theta(x,t) = x - \frac{t-k}{t}\big(x - \hat{x}_0(x)\big) = \frac{k}{t}x + \big(1 - \frac{k}{t}\big)\hat{x}_0(x)$. Its Jacobian is $\frac{k}{t}I + \big(1 - \frac{k}{t}\big)\partial_x\hat{x}_0$, whose norm is at most $\frac{k}{t} + \big(1 - \frac{k}{t}\big)\|\partial_x\hat{x}_0\| \le 1$. Along the exact flow, a perturbation δ obeys $\dot{\delta} = (I - \partial_x\hat{x}_0)\,\delta/t$; in a direction where $\partial_x\hat{x}_0$ has eigenvalue λ, $\delta(t) = \delta(k)\,(t/k)^{1-\lambda}$ for $t < k$, so components that the denoiser removes ($\lambda < 1$, off the data manifold) decay and components that it keeps ($\lambda \approx 1$) are preserved.

A linear model of the window. Let $\hat{x}_0(x) = \lambda x$ in one direction and $a = 1 - \lambda \in [0,1]$, so that $v = a\,x/t$ and the exact flow maps $x(t)$ to $(k/t)^a x(t)$. One Euler step multiplies by $1 - a\,h/t \in [k/t, 1]$. One Heun step multiplies by $1 - \frac{a}{2}\frac{h}{t} - \frac{a}{2}\frac{h}{k}\big(1 - a\frac{h}{t}\big)$, which depends only on h/k. Compared with the exact factor, Heun is more accurate than Euler for h/k below 3.0, 2.3, and 1.9 at $a = 0.25$, 0.5, and 0.75, and less accurate above; at $a = 0$ and $a = 1$ both are exact. For $h/k > 12.9$ the Heun factor exceeds one in absolute value for some a (worst near $a = 0.58$), so the step amplifies perturbations, while the Euler factor never does. The correction of a multi-stage step thus first loses accuracy and then stability as h/k grows, the behaviour of an explicit method on a stiff equation.

Proof of Proposition 5. Let x_i be the sample at t_i and $e_i = x_i - x_{t_i}^*$, with $e_0 = 0$ because the reference path starts at the initial noise. Then $e_{i+1} = \big[\Phi_{s_i}(x_i) - \Phi_{s_i}(x_{t_i}^*)\big] + d_i$, and to first order $e_N = \sum_i P_i d_i$, where P_i is the product of the step Jacobians after step i. Because $x_{t_N}^* = x_0$, e_N is the endpoint error. Expanding $\mathbb{E}\|e_N\|^2$, the cross terms vanish by the second condition and $\mathbb{E}\|P_i d_i\|^2 \le \mathbb{E}\|d_i\|^2$ by the first. With norms instead of squares, $\|e_N\| \le \sum_i \|d_i\|$ follows from the first condition alone.

The reference path gives the best entry points on average. Let x_0 and ε be independent and $t > k$. Writing $x_t = \frac{1-t}{1-k}x_k + c\,\varepsilon'$ with $c^2 = t^2 - \big(\frac{(1-t)k}{1-k}\big)^2 \ge 0$ and ε' independent reproduces the joint law of (x_0, x_t), so x_t carries no more information about x_0 than x_k does, and $\mathbb{E}\|x_0 - \mathbb{E}[x_0 \mid x_t]\|^2 \ge \mathbb{E}\|x_0 - \mathbb{E}[x_0 \mid x_k]\|^2$. Any state reached from x_t is a function of x_t, so on average no step can arrive at time k with more information about x_0 than the forward state x_k. The one-step errors therefore measure the distance of a step's landing point from the best entry point at its end time.

J READING THE GRAPH AND THE PLANNER

On SDXL the residual $\|r_t\|^2$ is lowest at the noise end and rises about 125-fold towards the clean image; on the flow models it is lowest mid-trajectory (Figure 4). Figure 5 shows the Mix path at every budget from 2 to 50 NFE. The smallest budgets go entirely to Euler. The first multi-stage step appears at 3 NFE on SDXL and at 5 on the two Qwen models; DPM-Solver++(2M) enters at 12 NFE on SDXL, 17 on Qwen-Image, and 30 on Qwen-Image-Edit, and takes three quarters of the budget at 50 NFE on SDXL. The share of multi-stage steps follows the width of the window: at 10 NFE Heun takes 80% of the budget on Qwen-Image, whose window is widest, and 20% on SDXL.

Z-Image-Turbo is pure Euler up to 13 NFE and has no multi-stage step below 38 NFE. On all four models the last step into the clean image is Euler at every budget.

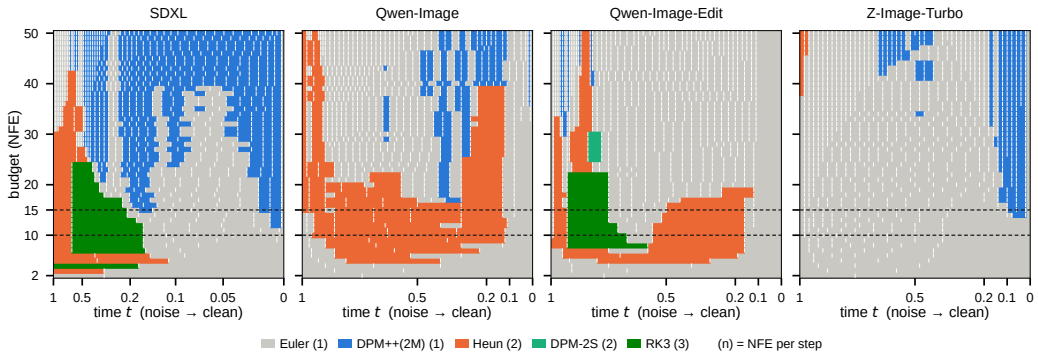


Figure 5: Where the planner spends its NFE, budget by budget. Each row is the lowest-cost path on the graph for one budget (2 to 50 NFE), read from noise (left) to the clean image (right); the colour is the solver used at that moment and thin white ticks separate steps. Grey: Euler step; blue: DPM-Solver++(2M), which reuses the previous NFE; other colours: extra NFE spent inside a step. Dashed lines mark the budgets tested on GenEval (10 and 15). Of the eight solvers, the planner uses only five at any budget on any model: Euler, DPM-Solver++(2M), Heun, RK3 (SDXL and Qwen-Image-Edit) and DPM-Solver-2S (Qwen-Image-Edit, 25–30 NFE). UniPC-2C, DPM-Solver-3S and RK4 are never on a lowest-cost path. With CFG one NFE is two network passes (one on Z-Image-Turbo).

K Full GenEval and ImageReward results

Table 1 gives GenEval and Table 10 the steps and ImageReward of every member of the planned family at 10 and 15 NFE; Figure 6 shows their paths. Tables 11 and 12 give the six GenEval task scores of every run; the rows named after a solver are the members of the planned family of Section 4.3. The `Mix, K steps` rows are Mix paths planned for exactly K steps; their NFE is given in parentheses.

ImageReward at 7 and 25 NFE. At 7 NFE the mean ImageReward of native, Euler, and Mix is 0.249, 0.815, and 0.811 on Qwen-Image and -0.600, 0.288, and 0.244 on SDXL; at 25 NFE Mix reaches 1.299 and 0.578.

L Random-path experiments

For each model and each budget of 5, 10, 15, and 25 NFE we draw 1000 legal paths from the eight-solver graph, stratified over the number of steps, and include the Mix path, which is the lowest-cost path. Each path generates the same 50 prompts, unrelated to the calibration prompt, from the same initial noise (seed 1000 plus prompt index). Every image is scored with ImageReward-v1.0 and the score of a path is its mean over the prompts. A random path is counted as significantly better than the lowest-cost path when its mean is higher and a paired t-test over prompts rejects equality at the 5% level. Table 13 gives the numbers behind Figure 3.

M The window on the true trajectory

The one-step errors of the graph are measured from the reference path. To test the window without it, we integrate the sampling equation of FLUX.1-dev and SD3.5-Large from the initial noise with Heun steps of length 0.0025 and keep the states X_i on the grid $t_i = 1 - i/100$, $t_i \geq 0.01$. On the first prompt, integrating with half that step length changes the states by a relative mean squared difference of at most 5.4×10^{-5} on SD3.5-Large and 2.0×10^{-3} on FLUX.1-dev. From every state X_i we take one Euler step, one Heun step, and two consecutive Euler steps through every intermediate grid time, the second starting where the first landed, keep the pair of Euler steps that lands closest, and compare

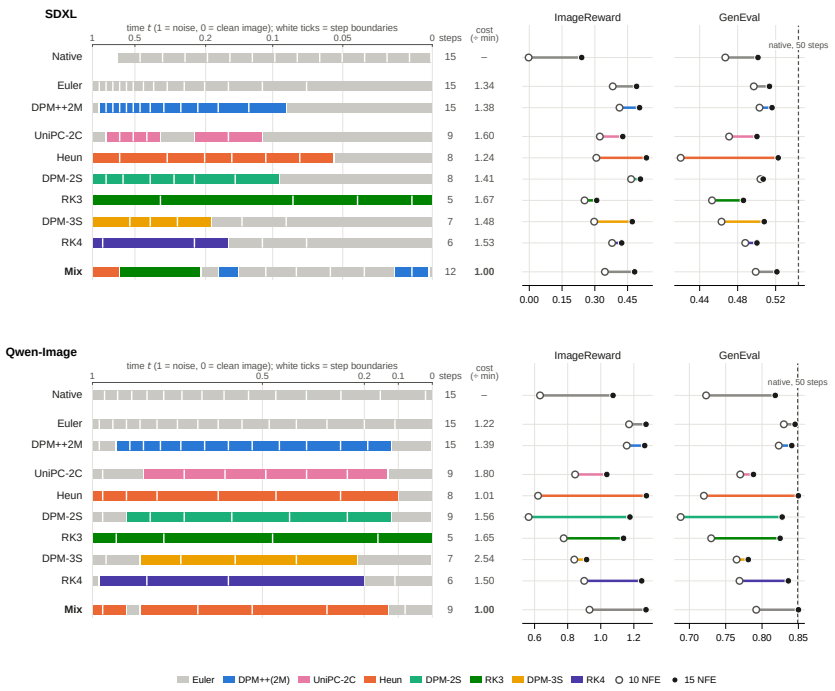


Figure 6: Every planned sampler at the same budget of 15 NFE. Left: the path of each sampler on the time axis; colour = solver, white ticks = step boundaries; steps = number of updates the 15 NFE buy; cost = graph path cost relative to the lowest-cost path (Mix). Native: the model's own scheduler. Right: ImageReward and GenEval at 10 (open) and 15 (filled) NFE; the connecting line takes the colour of the row's main solver; the dashed line is the native sampler at 50 steps. With CFG, B NFE are $2B$ network passes.

each landing point with the trajectory state at the end of the interval. Table 14 pools all intervals of at least two grid cells over 5 prompts on SD3.5-Large and 7 on FLUX.1-dev, none of them the calibration prompt.

On both models the advantage of the Heun step declines with h/k. Against one Euler step it wins on almost every interval below $h/k = 3$ on FLUX.1-dev and below $h/k = 7$ on SD3.5-Large, and on 11% and 82% of the intervals beyond $h/k = 10$. Against two Euler steps the median ratio of the Heun error to the two-Euler error rises with h/k on both models, from 0.65 to 2.9 on FLUX.1-dev and from 0.88 to 1.39 on SD3.5-Large, and the share of intervals on which the Heun step wins falls to zero beyond $h/k = 4$ on FLUX.1-dev and to 16% beyond $h/k = 10$ on SD3.5-Large. The decline holds on each prompt separately; beyond $h/k = 10$ on SD3.5-Large the Heun step beats two Euler steps on 9 to 24% of the intervals of every prompt.

N MODELS NOT USED IN THE ANALYSIS

FLUX.1-dev (Black Forest Labs, 2024) and SD3.5-Large (Esser et al., 2024) were evaluated after the analysis of Sections 3–4 and with no change to it. Both are flow models sampled at 1024×1024 with the native flow-matching Euler scheduler of diffusers and its time shift, guidance 3.5 (classifier-free guidance on SD3.5-Large, two network passes per NFE; the distilled guidance input on FLUX.1-dev, one pass per NFE). For each we build the Euler and Heun layers of the graph on the 100-interval uniform grid from the calibration prompt and seed of SDXL (Table 6), with the native 50-step sample as reference. At 10 NFE we compare the native scheduler with ten Euler steps, Heun on the native scheduler's five steps, and Euler on the lowest-cost ten-step path of the Euler layer. All three are evaluated on the 553 GenEval prompts with 4 seeds (Table 15).

Table 10: The whole planned family against the native scheduler (the ImageReward behind Figure 6; GenEval is in Table 1). Steps: number of steps the budget buys; Euler: how many of those steps are Euler steps (a member named after a solver uses it wherever it is defined and Euler elsewhere). IR: ImageReward. Bold: best at that budget. Both models use classifier-free guidance, so B NFE are $2B$ network passes.

	10 NFE			15 NFE		
Sampler	steps	Euler	IR	steps	Euler	IR
SDXL						
native	10	10	-0.003	15	15	0.240
Euler	10	10	0.382	15	15	0.491
DPM++(2M)	10	2	0.413	15	2	0.505
UniPC-2C	6	2	0.323	9	3	0.428
Heun	5	0	0.307	8	1	**0.536**
DPM-2S	6	2	**0.466**	8	1	0.509
RK3	4	1	0.253	5	0	0.308
DPM-3S	4	1	0.297	7	3	0.471
RK4	4	2	0.379	6	3	0.423
Mix	7	5	0.346	12	7	0.482
Qwen-Image						
native	10	10	0.633	15	15	1.074
Euler	10	10	**1.171**	15	15	1.274
DPM++(2M)	10	3	1.157	15	3	1.266
UniPC-2C	7	4	0.845	9	3	1.037
Heun	5	0	0.621	8	1	**1.276**
DPM-2S	6	2	0.564	9	3	1.176
RK3	4	1	0.776	5	0	1.138
DPM-3S	6	4	0.840	7	3	0.914
RK4	4	2	0.900	6	3	1.247
Mix	6	2	0.932	9	3	1.274

Table 11: GenEval task scores on Qwen-Image (553 prompts $\times$ 4 seeds). With classifier-free guidance, B NFE are $2B$ network passes.

Run (budget in NFE)	single	two	counting	colors	position	color attr	overall
native 7 NFE	91.25	60.61	63.12	76.60	43.75	36.50	0.620
native 10 NFE	95.62	70.20	75.62	84.57	53.50	54.25	0.723
native 15 NFE	98.44	89.65	85.00	89.36	63.25	65.25	0.818
Euler 7 NFE	97.81	81.57	81.56	86.17	58.75	58.50	0.774
Euler 10 NFE	98.44	93.43	82.81	89.89	63.00	70.25	0.830
Euler 15 NFE	99.06	96.21	86.56	90.69	63.75	71.00	0.846
Mix 7 NFE	97.19	82.83	80.31	85.64	59.00	55.50	0.767
Mix 10 NFE	97.19	84.34	83.75	87.77	61.50	60.75	0.792
Mix 15 NFE	98.75	96.97	88.12	91.49	64.50	70.25	0.850
Mix 25 NFE	99.06	97.22	89.06	91.76	64.25	70.75	0.854
Mix, 7 steps (16 NFE)	98.75	95.96	89.69	91.76	63.75	71.50	0.852
Mix, 10 steps (22 NFE)	99.06	96.72	89.38	90.69	63.50	74.00	0.856
Mix, 15 steps (31 NFE)	99.06	96.97	89.06	90.69	62.50	72.75	0.852
Heun 10 NFE	95.00	72.47	78.44	86.70	50.50	49.00	0.720
DPM-Solver-2S 10 NFE	93.12	68.43	74.06	80.59	48.25	48.25	0.688
DPM-Solver++(2M) 10 NFE	98.75	93.69	81.88	89.36	63.50	66.50	0.823
UniPC-2C 10 NFE	97.19	81.06	81.56	86.17	58.25	57.50	0.770
RK3 10 NFE	96.56	73.23	75.62	84.04	55.25	53.25	0.730
DPM-Solver-3S 10 NFE	97.81	81.06	81.56	84.04	59.00	55.50	0.765
RK4 10 NFE	97.19	80.30	82.19	85.37	59.75	56.75	0.769

O Z-IMAGE-TURBO

On the graph of the distilled model, one Heun step is worse than one Euler step on 89% of the intervals ending at or below $k = 0.2$ (Table 8). Because the Mix plan is pure Euler up to 13 NFE (Appendix J), the planned sampler and the native scheduler differ only in where they place their steps. Table 16 gives GenEval at 2, 4, and 8 NFE.

Table 12: GenEval task scores on SDXL (553 prompts $\times$ 4 seeds). With classifier-free guidance, B NFE are $2B$ network passes.

Run (budget in NFE)	single	two	counting	colors	position	color attr	overall
native 7 NFE	87.81	46.46	23.44	61.70	6.00	7.50	0.388
native 10 NFE	94.06	58.08	30.00	73.94	8.25	16.00	0.467
native 15 NFE	96.56	63.89	34.69	77.93	9.75	18.00	0.501
Euler 7 NFE	96.25	53.54	35.62	77.93	6.75	15.75	0.476
Euler 10 NFE	97.19	58.08	39.69	78.72	8.00	16.50	0.497
Euler 15 NFE	97.19	62.37	43.12	77.93	8.75	18.75	0.514
Mix 7 NFE	96.56	50.25	34.69	75.00	7.25	12.50	0.460
Mix 10 NFE	96.88	58.33	37.19	82.45	8.00	16.50	0.499
Mix 15 NFE	98.75	66.16	40.00	81.12	10.25	16.50	0.521
Mix 25 NFE	98.44	70.45	44.69	84.57	11.00	17.25	0.544
Mix, 7 steps (11 NFE)	97.50	62.12	38.75	82.98	9.25	14.50	0.509
Mix, 10 steps (15 NFE)	98.12	66.92	38.12	81.65	10.50	17.75	0.522
Mix, 15 steps (24 NFE)	98.75	68.18	42.50	84.84	8.25	19.25	0.536
Heun 10 NFE	88.44	48.23	26.88	70.48	6.50	11.50	0.420
DPM-Solver-2S 10 NFE	96.25	61.62	38.75	76.33	9.50	20.00	0.504
DPM-Solver++(2M) 10 NFE	96.25	59.34	42.50	78.19	7.00	18.25	0.503
UniPC-2C 10 NFE	97.50	50.76	38.12	75.27	8.25	12.50	0.471
RK3 10 NFE	95.62	49.49	32.50	74.47	8.00	12.00	0.453
DPM-Solver-3S 10 NFE	95.00	49.75	38.12	73.67	6.75	14.75	0.463
RK4 10 NFE	97.81	58.08	36.56	77.93	7.00	15.50	0.488

Table 13: Numbers behind Figure 3. Spearman: rank correlation between path cost and mean ImageReward over the 1000 paths. Better than: share of the 999 random paths whose mean ImageReward is below that of the planned (lowest-cost) path. Above: number of random paths that score higher. Both models use classifier-free guidance, so B NFE are $2B$ network passes.

Model	NFE	Spearman	better than	above
SDXL	5	-0.794	99.9%	1
SDXL	10	-0.813	94.1%	59
SDXL	15	-0.817	97.7%	23
SDXL	25	-0.772	94.9%	51
Qwen-Image	5	-0.335	95.6%	44
Qwen-Image	10	-0.255	99.1%	9
Qwen-Image	15	-0.252	99.8%	2
Qwen-Image	25	-0.322	98.2%	18

P Distance to the source image in editing

Measured against the source image instead of the 50-step edit, the five 10-NFE samplers of Table 3 have SSIM 0.678–0.692, LPIPS 0.324–0.336, L1 0.093–0.099, and PSNR 15.9–16.2; these distances measure the extent of the edit and are nearly identical across samplers. The unedited source scores ImageReward -0.60 under the edited-image descriptions, so the reward is sensitive to whether the edit was performed.

Q Samples

Figures 7 and 8 compare the native scheduler and the Mix path at the same NFE, starting from the same initial noise; each Mix path is the lowest-cost path on the model's own graph and is written next to the images as its ordered sequence of steps. The native samplers take one Euler step per NFE. On SDXL the multi-NFE steps (Heun, RK3) cover the trajectory from $t = 1$ down to $t = 0.11$–0.21, and the remaining steps are Euler steps, with three DPM-Solver++(2M) steps at 15 NFE. On Qwen-Image the Heun steps lie between $t = 0.97$ and $t = 0.12$–0.40, and the last one or two steps are Euler steps.

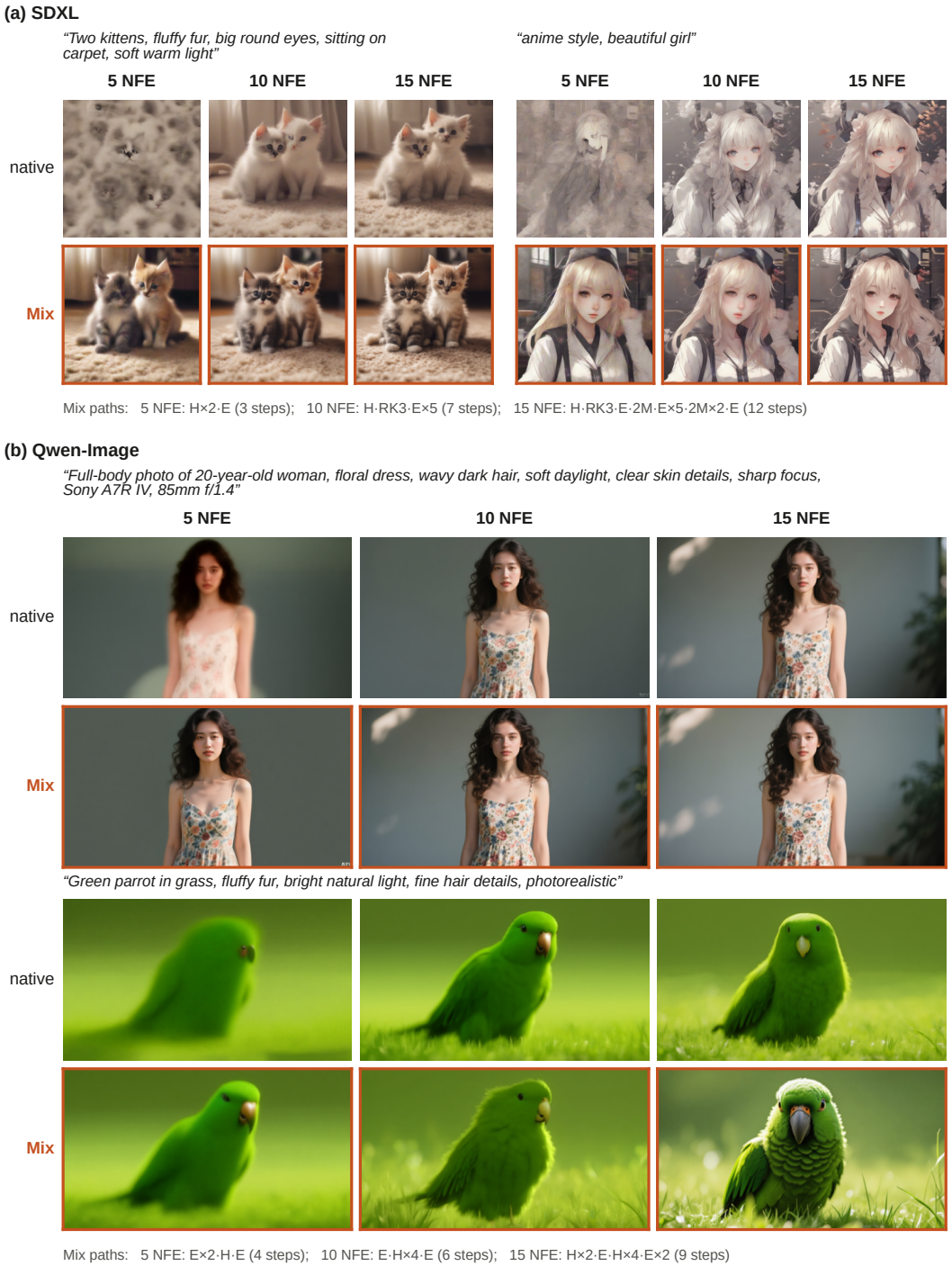


Figure 7: Text-to-image samples, native scheduler and Mix at 5, 10, and 15 NFE, same seed per prompt. (a) SDXL and (b) Qwen-Image, two prompts each. Both models use classifier-free guidance, so 5, 10, and 15 NFE are 10, 20, and 30 network passes. The line under each group gives the Mix path at each budget, read from noise to image (E Euler, H Heun, RK3, 2M DPM-Solver++(2M); $\times k$ marks k consecutive steps of the same solver), with its number of steps; the native samplers take one Euler step per NFE. Mix images are framed.

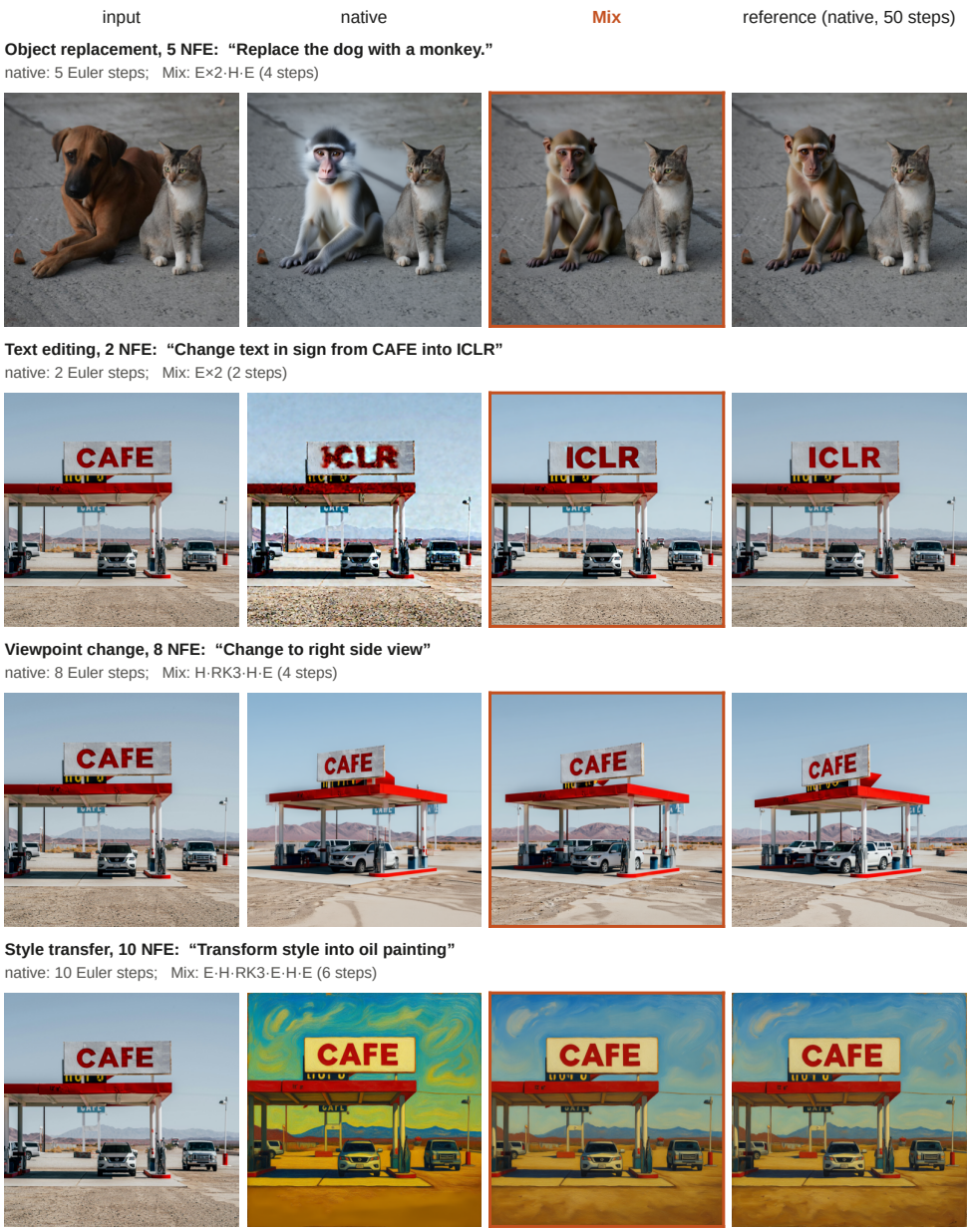


Figure 8: Qwen-Image-Edit, one type of edit per row: input image, native scheduler, Mix planned on the editing model’s graph, and a reference from the native scheduler at 50 steps. Qwen-Image-Edit uses classifier-free guidance, so B NFE are $2B$ network passes. The line under each instruction gives the Mix path (E Euler, H Heun, RK3; $\times k$ marks k consecutive steps of the same solver). Mix images are framed.

Table 14: One Heun step against two Euler steps and against one Euler step on the true trajectory, by h/k: share of intervals on which the Heun step lands closer to the trajectory state. n: number of intervals, pooled over prompts. SD3.5-Large uses classifier-free guidance (two network passes per NFE); FLUX.1-dev takes one pass per NFE.

	FLUX.1-dev (7 prompts)			SD3.5-Large (5 prompts)		
h/k	n	vs. 2 Euler	vs. 1 Euler	n	vs. 2 Euler	vs. 1 Euler
$[0, 0.5)$	10717	68.9%	99.5%	7655	61.5%	100%
$[0.5, 1)$	5817	48.3%	100%	4155	54.8%	100%
$[1, 2)$	5957	22.2%	100%	4255	53.6%	100%
$[2, 3)$	2954	0.9%	99.1%	2110	53.6%	100%
$[3, 4)$	1792	0.3%	89.6%	1280	50.8%	100%
$[4, 5)$	1176	0.0%	72.5%	840	51.2%	100%
$[5, 7)$	1470	0.0%	42.1%	1050	50.0%	100%
$[7, 10)$	1218	0.0%	24.3%	870	43.4%	99.5%
$[10, \infty)$	2856	0.0%	10.9%	2040	16.4%	81.6%

Table 15: GenEval task scores at 10 NFE on the two models that were not used in the analysis (553 prompts $\times$ 4 seeds). SD3.5-Large uses classifier-free guidance, so 10 NFE are 20 network passes; FLUX.1-dev takes one pass per NFE.

Sampler (10 NFE)	single	two	counting	colors	position	color attr	overall
FLUX.1-dev							
native, 10 Euler steps	96.88	**80.56**	68.75	74.73	19.00	35.50	0.626
native, 5 Heun steps	**98.12**	67.68	65.00	71.28	19.00	23.75	0.575
planned, 10 Euler steps	97.50	79.55	**69.06**	**77.39**	**19.25**	**43.75**	**0.644**
SD3.5-Large							
native, 10 Euler steps	98.12	83.33	**64.38**	**84.31**	**24.75**	56.75	0.676
native, 5 Heun steps	97.19	70.45	61.56	80.59	22.50	44.75	0.628
planned, 10 Euler steps	**98.44**	**83.59**	63.12	83.51	23.75	**57.25**	**0.683**

Table 16: GenEval task scores on Z-Image-Turbo (553 prompts $\times$ 4 seeds; no guidance, one network pass per NFE). The planned sampler is pure Euler at these budgets, so Euler and Mix coincide.

Sampler	single	two	counting	colors	position	color attr	overall
planned, 2 NFE	99.38	87.37	71.25	**89.10**	**49.00**	61.00	**0.762**
native, 4 steps	99.69	89.39	71.88	86.70	48.25	61.00	**0.762**
planned, 4 NFE	99.69	89.39	**72.50**	86.97	46.50	**61.50**	0.761
native, 8 steps (default)	**100.00**	**91.16**	70.94	86.44	47.00	58.75	0.757
planned, 8 NFE	**100.00**	90.15	71.25	86.70	45.50	59.00	0.754