REPLAY THE SEEDS, STORE THE REMAINDER: COMPRESSING PERSISTENT DIFFUSION INVERSION STATE

Anonymous authors
Paper under double-blind review

ABSTRACT

Inversion-based diffusion editors retain dense stochastic controls to support repeated text-guided edits from a single inversion, imposing severe storage and bandwidth bottlenecks. We observe that a substantial component of each materialized control is deterministically governed by the inversion solver and its generating random draws. Leveraging this insight, we introduce *replay factorization*, a denoiser-free representation that reconstructs an analytical predictor directly from recorded provenance, projects the remainder onto solver-prescribed stochastic carrier directions, and compactly codes the residual defect alongside endpoint and metadata. We formally establish the *stencil-consistency condition* under which explicit source-latent terms vanish, derive exact replay coordinates for first- and second-order stochastic solvers, and generalize the construction to shifted paths and optimized noise. Evaluated on all 700 PIE-Bench tasks across four U-Net configurations (Edit-Friendly and LEDITS++), replay factorization achieves 35.77–$47.04\,\%$ median complete-file savings over optimized generic codecs while preserving edit fidelity (median LPIPS 0.00023–0.0023 to raw-state edits). Without per-source fitting, parameter-free analytical predictors retain the majority of these gains. On the PixArt-α diffusion Transformer, our lossy format saves $45.25\,\%$ storage, while an exact-state variant achieves bitwise input restoration with $14.21\,\%$ median savings over native-lossless baselines. The exact framework similarly compresses four-step ReNoise and TurboEdit states without loss. By transforming generating provenance into reproducible structure at the decoder, replay factorization substantially reduces persistent editing state without evaluating neural denoisers.

1 INTRODUCTION

Diffusion editors support repeated edits of the same source image. Edit-Friendly (EF) and LEDITS++ retain an endpoint latent and per-step controls so later requests can reuse the source inversion. Each source becomes a persistent editing asset whose size determines how many assets fit a storage budget and the volume of state transferred between storage and editing workers. Exploratory editing motivates LEDITS++ (Brack et al., 2024), while NIRVANA illustrates cross-request reuse in production text-to-image serving (Agarwal et al., 2024). We study how to store these assets compactly while preserving the outputs of the editor that created them.

Each retained state is large because its per-step controls are dense tensors. For 140 EF/SD 1.5 sources, median complete files, including the endpoint and decoder metadata, are 4.3 MB raw and 1.46 MB with the per-source best generic codec. Yet the solver produced a substantial portion of each control from random draws that a decoder can regenerate. We therefore ask which parts of a materialized state can be replayed from its generating provenance, and how the remainder should be coded to preserve compatibility with the unmodified downstream editor. Our representation stores these states in 0.77 MB at the same nominal tolerance, nearly doubling the number of states a fixed storage budget can accommodate.

The stored controls contain reproducible random terms as well as source-dependent information. Generic codecs encode these terms as realized tensor values. Under the recorded random-number generator (RNG) contract, we regenerate the solver-weighted draws, project the remainder onto

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.

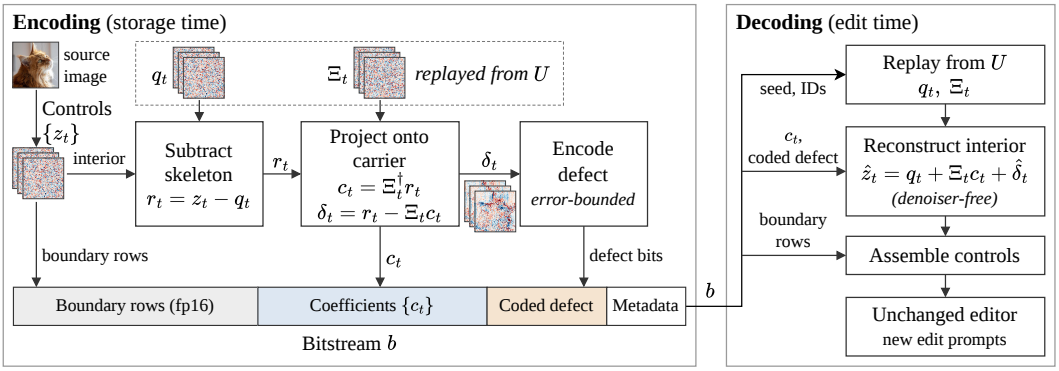

Figure 1: Lossy storage for repeated edits from one inversion. Both sides replay q_t, Ξ_t from provenance U (Section 3.1); the bitstream stores interior coefficients and coded defects, fp16 boundaries, and source seed/IDs. EF/SD 1.5 thumbnails: PIE-Bench source 421000000003, latent channel 1 (zero-based), interior rows $t = 631, 601, 581$ (front to back), independently scaled to ± 2.5 s.d. Native endpoint and shared model/runtime inputs are omitted; block widths are schematic.

solver-prescribed draws, and encode the residual (Figure 1). Source-specific coefficients and residuals remain in storage, together with boundary rows, the endpoint, and metadata.

The solver's latent-state weights determine whether the explicit source-latent term cancels. We derive this condition and identify the projection draws: the current draw for EF, and draws at the current and preceding solver points for second-order LEDITS++. For TurboEdit's shifted four-step path, the same analysis constructs a predictor using the source latent and the noise target at the model timestep. These coordinates provide an analytical predictor before fitting; fitted coefficients adapt it to each materialized control.

Let z_t denote a control retained by the editor, distinct from the original random draws ξ_i. The *seed skeleton* q_t is a solver-weighted combination of these draws, and the *carrier* Ξ_t contains the prescribed draws as columns. With fitted coefficients c_t and a stored residual, the *defect* δ_t, each interior control has the representation

$$z_t = q_t + \Xi_t c_t + \delta_t. \tag{1}$$

Boundary rows and metadata complete the bitstream. The unchanged editor consumes the reconstructed controls.

Our contributions are:

- **A provenance-derived representation of inversion state.** We establish the stencil-consistency condition under which the explicit source-latent term cancels from a stochastic control row (Lemma 1), derive replay coordinates for the EF update and the second-order LEDITS++ stencil, and construct predictors for TurboEdit's shifted path and ReNoise's optimized noise (Appendix A.3). Coefficients, boundary rows, and a full-dimensional coded remainder complete each state; decoding requires no denoiser evaluation (Section 4).

- **Predictive information from generating draws.** Generating provenance supplies predictive information; solver analysis specifies draws, weights, and source terms before source-wise fitting. Analytical prediction retains most fitted savings. Original-draw least squares achieves comparable file sizes, exhaustive support search recovers the prescribed draws, and unrelated draws consistently increase exact-state file sizes (Section 5.3).

- **Compact persistent states and the object to retain.** We measure complete-file savings and editing fidelity for lossy U-Net and PixArt-α states on all 700 PIE-Bench tasks. Exact-state and four-step studies compare retained objects and warm-query costs (Sections 5.2, 5.4, and 5.6).

2 Related Work

Inversion and reusable state. CycleDiffusion, EF, and LEDITS++ expose per-step stochastic controls (Wu & De la Torre, 2023; Huberman-Spiegelglas et al., 2024; Brack et al., 2024), as do few-step TurboEdit, Cora, and ReNoise's stochastic-sampler inversion (Deutch et al., 2024; Alimohammadi et al., 2025; Garibi et al., 2025). PnP Inversion (formerly Direct Inversion), DNAEdit, and DirectEdit retain trajectory-aligned corrections (Ju et al., 2024; Xie et al., 2025; Yang & Ye, 2026). Null-text inversion optimizes conditioning; ResEdit learns source-specific residual embeddings (Mokady et al., 2023; Baykal et al., 2026). DICE and UniEdit-Flow extend editable-state construction to discrete and flow settings (He et al., 2026; Jiao et al., 2026). These objects motivate our coding objective to evaluate storage representations while keeping the editor and its materialized state strictly fixed. Appendix H.3 groups persistent objects by their generating computations.

Reuse and tensor compression. FISEdit and RegionCache reuse features across edits, NIRVANA caches diffusion states across generation requests, and IF-V2V reuses source denoising vectors within a run (Yu et al., 2024; Li et al., 2026; Agarwal et al., 2024; Kong et al., 2025). SVD-Cache predicts principal feature components and reuses residual components to accelerate denoising (Chen et al., 2026). Our storage format uses predictive residual coding (Liang et al., 2023), with a predictor derived from the inversion state's original draws and solver stencil. Fixed error-bounded codecs on raw controls and defects isolate the representation's storage effect; learned codecs (Ballé et al., 2018; Yang et al., 2023) could use either interface.

Seed-based representations. SeedLM codes static weight blocks and AWSRC codes quantization residuals through seed-generated bases (Shafipour et al., 2025; Liu et al., 2026); PRANC and NOLA reparameterize weights with seed-generated random bases and learned coefficients (Nooralinejad et al., 2023; Abbasi Koohpayegani et al., 2024). DDCM and Turbo-DDCM select atoms from frozen Gaussian codebooks for image representations, with DDCM decoding stored codes under an edit prompt (Ohayon et al., 2025; Vaisman et al., 2026). Our bases are the draws that generated the stored editor state, identified by its provenance and solver stencil.

Noise-Anchored Reverse Correction (NARC) stores a forward-noise anchor and blends it into reverse-time noise predictions (Park et al., 2026). Its storage evaluation retains quantized anchors; deterministic seeds provide reproducibility and pairing. The same work also compresses materialized EF and LEDITS++ caches element-wise (int8, int4) and through subspace summaries (DCT, random projection), reports element-wise versus subspace compression asymmetry for both editors, and argues that inversion state is trajectory-bound. Our representation keeps the editor, its reverse process, and its materialized state fixed: it predicts each control from the draws that generated it and codes the full-dimensional remainder with an error-bounded or lossless codec, so no subspace is truncated and the exact format preserves native state bits. DeepCache reuses intermediate features during denoising (Ma et al., 2024); here the objective is persistent-state storage.

3 Persistent Inversion State as a Coding Problem

3.1 State and decoder-side provenance

Let S be the source-specific state an editor materializes for later edits. Versioned replay provenance U specifies model and VAE revisions, scheduler parameters, solver stencil, seed, RNG contract, software environment, and tensor conventions. Shared model/runtime inputs are supplied externally; source-specific replay fields are stored and counted in the bitstream. The coding interface is

$$b = \mathrm{Enc}(S; U), \qquad \widehat{S} = \mathrm{Dec}(b; U). \tag{2}$$

For each target prompt, the same editor can operate on either S or $\widehat{S}$ interchangeably. Matching provenance is required to reproduce the indexed draws and solver coefficients used during encoding; fitted coefficients c_t are read from the bitstream.

Scope. We evaluate EF and LEDITS++ on SD 1.5/SDXL, an EF port to PixArt-α, and four-step ReNoise and TurboEdit. EF/LEDITS++ satisfy interior source cancellation; TurboEdit uses a shifted path, and ReNoise optimizes its original draws. Fidelity is measured against each editor's raw-state output. Appendix H.3 describes other persistent objects.

3.2 STORAGE AND FIDELITY OBJECTIVE

Payload bytes B comprise the encoded controls (coefficients, boundary rows, defect bitstream, and metadata) plus the native endpoint: the inverted endpoint latent retained by the editor (65.536 kB for EF on SD 1.5). Complete-file size S^{file}, the standalone size, also includes the integrity and provenance envelope required for an independent reload. The count suffix k denotes 10^3; storage units are decimal (1 kB $= 10^3$ B, 1 MB $= 10^6$ B). Model and VAE revisions are shared decoder inputs. The no-store path recomputes the native source processing when queried. We compare payload and complete-file size separately under the same specification for generic and factorized representations.

We minimize stored bytes under a state-error criterion:

$$\min_b |b| \quad \text{s.t.} \quad d_{\text{state}}(S, \widehat{S}) \le \varepsilon_s. \tag{3}$$

Here d_{state} is maximum absolute fp32 reconstruction error. At nominal codec tolerance ε, ε_s is 1.25 times the best generic candidate's achieved error; Section 4.3 gives the full admission rule. Output fidelity $d_{\text{edit}}(\mathcal{E}(S), \mathcal{E}(\widehat{S}))$ is measured with the unchanged editor $\mathcal{E}$ at a fixed target prompt. Generating structure determines what can be replayed; the consuming editor determines the required reconstruction accuracy.

4 REPLAY FACTORIZATION

4.1 AFFINE STOCHASTIC SOLVERS AND SOURCE CANCELLATION

Each auxiliary latent combines the same source latent, scaled at each point, with a reproducible random draw. When the solver's signal weights match this path, the explicit source-latent terms cancel in the extracted control, leaving replayable noise terms and a model-dependent remainder.

A row is one timestep's control tensor; Ξ_t contains flattened draws as columns. Let u denote the produced solver point and J_u its *stencil*, the points supplying latents and predictions. Adjacent indices denote consecutive inference-grid points, not unit increments in training timestep labels.

Before post-inversion modifications, the exact-arithmetic solver control is z_u^{sol}. The auxiliary path is $x_\ell = a_\ell x_0 + b_\ell \xi_\ell$: x_0 is the source latent, a_ℓ, b_ℓ are signal and noise scales, and ξ_ℓ is an indexed draw reproducible under the RNG contract. For data predictions on this path, Appendix A.1 gives the conversion when $a_\ell \neq 0$. The *noise-prediction normal form* uses latent weights $A_{u,\ell}$, prediction weights $B_{u,\ell}$, and injected-control scale ς_u:

$$x_u = \sum_{\ell \in J_u} A_{u,\ell}\, x_\ell + \sum_{\ell \in J_u} B_{u,\ell}\, \varepsilon_\theta(x_\ell) + \varsigma_u z_u^{\text{sol}}. \tag{4}$$

Lemma 1 (Source cancellation)**.** *For a row with $\varsigma_u \neq 0$, substituting the auxiliary path into Equation 4 expresses z_u^{sol} as the sum of x_0 scaled by $(a_u - \sum_{\ell \in J_u} A_{u,\ell} a_\ell)/\varsigma_u$, a skeleton $q_u = (b_u \xi_u - \sum_{\ell \in J_u} A_{u,\ell} b_\ell \xi_\ell)/\varsigma_u$ of weighted draws, and a model term $m_u = -\sum_{\ell \in J_u} B_{u,\ell} \varepsilon_\theta(x_\ell)/\varsigma_u$. The explicit source-latent coefficient vanishes if and only if the stencil condition holds:*

$$a_u = \sum_{\ell \in J_u} A_{u,\ell}\, a_\ell \tag{5}$$

Applying Equation 5 to the sampler's noise-prediction normal form identifies the rows on which the predictor can omit x_0; the model predictions still carry source information. Appendix A.1 gives the conversion for noise- and data-prediction solvers. For EF and LEDITS++, boundary rows target the source latent itself; cancellation fails or the injected scale vanishes (Corollaries 1 and 2). They remain outside the factorization: lossy formats store them separately in fp16, and the exact-state format preserves their native bits in its whole-array XOR stream.

For an interior EF row, we use its departure point t: $u = t - 1$ and $J_u = \{t\}$ in Equation 4. Its update is $x_{t-1} = A_t x_t + B_{t,0} f_t + \sigma_t z_t^{\text{sol}}$, where σ_t is the injected-control scale, distinct from b_t. With $A_t = \sqrt{\bar{\alpha}_{t-1}/\bar{\alpha}_t}$, $a_t = \sqrt{\bar{\alpha}_t}$, and $f_t = \varepsilon_\theta(x_t)$, we have $a_{t-1} = A_t a_t$. For $\sigma_t \neq 0$, the row-labelled

skeleton and model term are

$$z_t^{\mathrm{sol}} = \underbrace{\frac{b_{t-1}\xi_{t-1} - A_t b_t \xi_t}{\sigma_t}}_{q_t} + \underbrace{\left(-\frac{B_{t,0} f_t}{\sigma_t}\right)}_{m_t}. \tag{6}$$

Let $\mathcal{I}_t$ index the draws at solver points contributing model predictions, and set $\Xi_t = [\xi_{i_1}, \ldots, \xi_{i_k}]$. For coding, z_t denotes the materialized control and we form the remainder $r_t := z_t - q_t$. On unmodified solver-produced rows in exact arithmetic, $r_t = m_t$. This definition also covers finite-precision discrepancies and post-inversion modifications; on the two interior LEDITS++/SDXL rows overwritten with zero, $r_t = -q_t$ (Appendix A.7). Using the Moore–Penrose pseudoinverse †, we fit

$$c_t = \Xi_t^{\dagger} r_t, \qquad \delta_t = r_t - \Xi_t c_t, \tag{7}$$

which yields Equation 1. The solver and recorded provenance determine q_t and the columns of Ξ_t; ordinary least squares (OLS) fits one or two coefficients per interior row. It removes the component of r_t in the carrier span, and a standard error-bounded tensor codec compresses the defect. Appendix A derives this projection.

4.2 Structure of the coded defect

For EF, $A_t = a_{t-1}/a_t$ and $B_{t,0} = \nu_t - A_t b_t$, where $\nu_t = \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}$. Define $e_t = \varepsilon_\theta(x_t) - \xi_t$ and $\kappa_t = -B_{t,0}/\sigma_t$. Then an unmodified interior control satisfies

$$z_t^{\mathrm{sol}} = \underbrace{\frac{b_{t-1}}{\sigma_t}\xi_{t-1} - \frac{\nu_t}{\sigma_t}\xi_t}_{\widetilde{q}_t = q_t + \kappa_t \xi_t} + \kappa_t e_t. \tag{8}$$

Thus the analytical predictor $\widetilde{q}_t$ is available before fitting. With $P_t = \Xi_t \Xi_t^{\dagger}$, the fitted defect is $\kappa_t(I - P_t)e_t$. When the injected variance is small relative to the path noise variance and adjacent noise scales are close, nearly equal weights on consecutive draws give the analytical predictor lag-one correlation approaching $-1/2$; consecutive predictors share one draw with opposite signs. The EF/SD 1.5 controls exhibit a measured correlation of -0.499 (Appendix A).

For a second-order LEDITS++ row, the analytical correction uses both draws associated with its noise predictions:

$$\widetilde{q}_j = q_j - \frac{B_{j,0}\xi_j + B_{j,1}\xi_{j+1}}{\varsigma}, \quad B_{j,0} = -\Big(\frac{K b_s}{a_s} + g\Big), \quad B_{j,1} = g' = \frac{K b_{s'}}{2 r_s a_{s'}}. \tag{9}$$

Corollary 2 defines K, g, r_s, and ς at the target, current, and preceding solver points u, s, s'. Replacing either draw changes this analytical prediction; local OLS fits their coefficients from each stored row instead.

For a general decomposition $z_t = d_t + u_t$, Equation 7 gives $\delta_t = (I - P_t)(d_t + u_t - q_t)$. The projected-error form follows when $d_t - q_t$ lies in the carrier span. This separates the structure supplied by the solver from finite-precision and post-inversion changes in the materialized state (Appendix A). Appendix A.6 relates this energy reduction to coded rate under a high-rate approximation; within each configuration, selected bytes track ρ_δ (Appendix A.7).

Solver instances. EF uses $\Xi_t = [\xi_t]$ on interior rows, with the first stored row as boundary. LEDITS++ follows the second-order SDE-DPM-Solver++ midpoint rule (Lu et al., 2025). In increasing-noise indexing, $[\xi_j, \xi_{j+1}]$ contains draws from the current and preceding reverse-solver points. The initial reverse row uses one draw; the last stored row is boundary. Appendix A.1 gives the coefficient counts and byte costs.

Cancellation enables the EF and LEDITS++ predictors to omit the source latent. When a solver retains the source term, the predictor can include it if the source latent is stored and accounted for, as in TurboEdit (Appendix A.3). ReNoise replays each original draw and codes the correction introduced by inversion-time optimization. Appendix A.3 derives both constructions; their exact formats include the source latent, endpoints, and bitwise repair streams.

4.3 Encoding, decoding, and representation selection

For lossy coding, the encoder regenerates q_t, Ξ_t, fits c_t, and codes δ_t with coefficients, boundaries, endpoint, and replay metadata. At nominal tolerance ε, a factorized candidate is admitted when its achieved maximum-absolute fp32 error is 0.8–1.25 times that of the best generic representation. Selection minimizes accounted representation size, with generic coding as a candidate. We measure editing fidelity after conversion to the editor's native dtype. Exact variants losslessly code corrections to replay predictions and restore native input arrays exactly (Appendices F.2 and F.1).

The suite uses SZ3, ZFP, delta + byte-shuffle + zstd (dbz), and pcodec (Liang et al., 2023; Lindstrom, 2014; Collet & Kucherawy, 2021; Blosc Development Team, 2009–2026; Loncaric et al., 2025). Each generic menu includes every inner codec used by its factorized arm; Appendix B.1 lists codec menus; Appendix G.3 gives integrity checks and provenance fields.

The decoder regenerates the skeleton and carrier without a denoiser. Coefficients require only one scalar per EF interior row and at most two per LEDITS++ row. The defect retains the original tensor shape for standard tensor coding.

5 Experiments

5.1 Protocol

We evaluate EF and LEDITS++ on SD 1.5 and SDXL, and EF on PixArt-α, across all 700 PIE-Bench v1 tasks. A balanced 140-source cohort, with 14 sources per edit type, supports tolerance sweeps and mechanism comparisons. Each comparison uses identical sources, prompts, and seeds across representations. Nominal codec tolerances are 10^{-2} for EF, including PixArt, and 3×10^{-3} for LEDITS++; state-error admission precedes conversion to the editor dtype. Analytical-predictor comparisons apply maximum-error caps to native inputs. Four-step ReNoise and TurboEdit use exact-state coding on the same 140 sources. Appendix B and Appendix F.2 give the settings.

For source i, let $B_{i,g}$, $B_{i,s}$, and $B_{i,v}$ denote the smallest generic, selected (including fallback), and variant-v payloads at the common nominal tolerance (Section 4.3). No generic fallback occurred in the reported evaluations of these five configurations. Storage savings are

$$\Gamma_{i,\mathrm{sel}} = 1 - \frac{B_{i,s}}{B_{i,g}}, \qquad \Gamma_{i,v} = 1 - \frac{B_{i,v}}{B_{i,g}}. \tag{10}$$

Complete-file saving replaces B by S^{file}. A matched generic minimizes size subject to achieved state error no larger than the selected representation's. L_{mimic} denotes the AlexNet LPIPS distance to the raw-state edit (Zhang et al., 2018), where lower values indicate higher perceptual fidelity. For $\Delta L_{\mathrm{mimic}}$ (selected minus generic), positive values indicate greater raw-state deviation. We report absolute error, its tails, and generic-relative differences; Appendix B specifies the cohort criteria and failure accounting.

Reporting conventions. Unless an interval is explicitly labeled two-sided, brackets report configuration-specific one-sided 95 % source-bootstrap bounds.

5.2 Payload and complete-file reduction

Across all 700 PIE-Bench tasks, the four U-Net configurations save 35.77–$47.04\,\%$ in median complete-file size (Table 1). Their median LPIPS to raw-state edits is 0.00023–0.0023, with P95 below 0.009. The 295-task subset with no recorded calibration or development exposure retains 35.71–$47.04\,\%$ median savings (Appendix C.1).

For PixArt's diffusion Transformer, tolerance and execution-precision analyses characterize the lossy rate–fidelity trade-off in Table 1 (Appendix E.5). Exact-state coding restores native input bits against the fp32 solver predictor. Fresh-process decoding restores all native inputs on all 700 tasks, saving $14.21\,\%$ in median file size; the 629 tasks outside predictor and format diagnostics have the same median. Identical inputs and deterministic execution yield identical edits for each prompt (Appendix F.1). Expanded 140-source fidelity and tolerance comparisons are detailed in Table 16 and Appendices C.3 and C.2.

Table 1: Complete-file storage on all 700 PIE-Bench tasks. Lossy rows use the selection rule in Section 4.3; LPIPS measures deviation from raw-state edits. The exact PixArt row uses the per-source best of six lossless configurations. Every configuration yields smaller files on all 700 tasks. The benchmark contains 692 distinct source-file hashes.

Configuration	State contract	File saving	LPIPS median / P95
EF/SD 1.5	Lossy	46.69 %	0.00023/0.00795
EF/SDXL	Lossy	47.04 %	0.00124/0.00768
LEDITS++/SD 1.5	Lossy	36.37 %	0.00049/0.00547
LEDITS++/SDXL	Lossy	35.77 %	0.00226/0.00824
EF/PixArt-α	Lossy	45.25 %	0.03682/0.16912
EF/PixArt-α	Exact native inputs	14.21 %	Bitwise input restoration

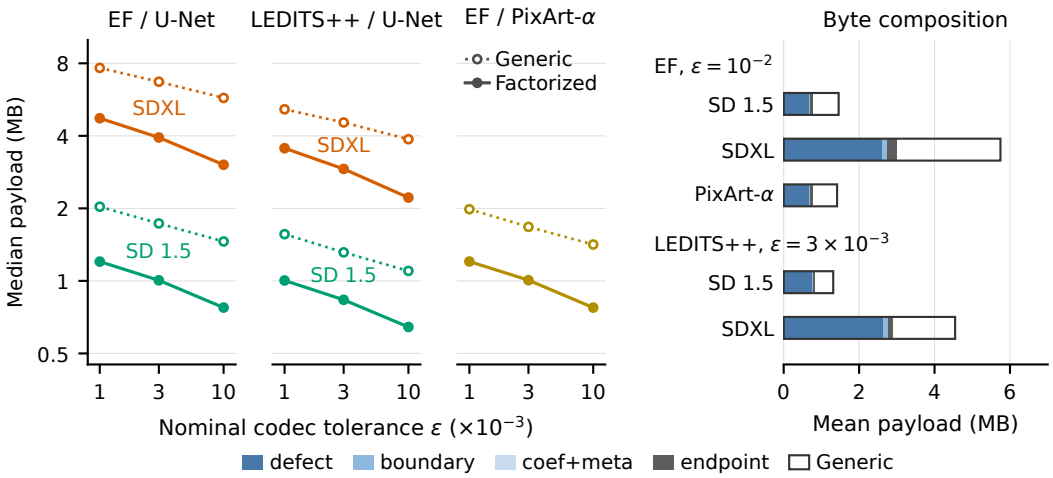

Figure 2: Payload size and composition on 140 sources per configuration. Left: median payloads versus nominal codec tolerance, with shared log axes. Right: mean payload components at the primary tolerance; outlines mark mean generic-best payloads. Coefficients and metadata are included.

Savings across tolerances. Figure 2 shows the storage advantage across three tolerances for U-Net and PixArt states. Full-dimensional defects dominate the stored payload; replay coefficients and metadata form a small part (Appendix C.2).

5.3 Structural and Seed Controls

Predictive information in the generating draws. On the 140 primary-cohort sources per configuration, dense OLS over all original draws leaves 0.15–1.48 % median residual energy relative to native-control energy across the five configurations. An equally wide unrelated-draw dictionary leaves 99.39–99.92 %. Each ratio sums energies over one source's active rows (Appendix D.3). In exact PixArt coding on 20 sources, local OLS over the original draws yields files comparable to the solver representation and a full original-draw dictionary retains most savings, whereas unrelated-draw and seed-blind temporal OLS increase file size under XOR/pcodec (Figure 3, B; Appendix D.5).

Solver structure and fitting. Exhaustive fixed-width search recovers the solver-prescribed support on 39,808 of 39,900 active rows from 140 sources per configuration. Across four U-Net configurations, median complete-file savings are 33.76–41.55 % with analytical replay and 35.70–43.17 % with fitted replay under common native-error caps (Figure 3, A). On the same 140 sources, exact PixArt analytical coding saves 13.51 % versus 14.25 % for fitted coding, with bitwise native-input restoration (Appendix D.1). Generating draws supply predictive information; solver analysis specifies a compact predictor without fitting or support search.

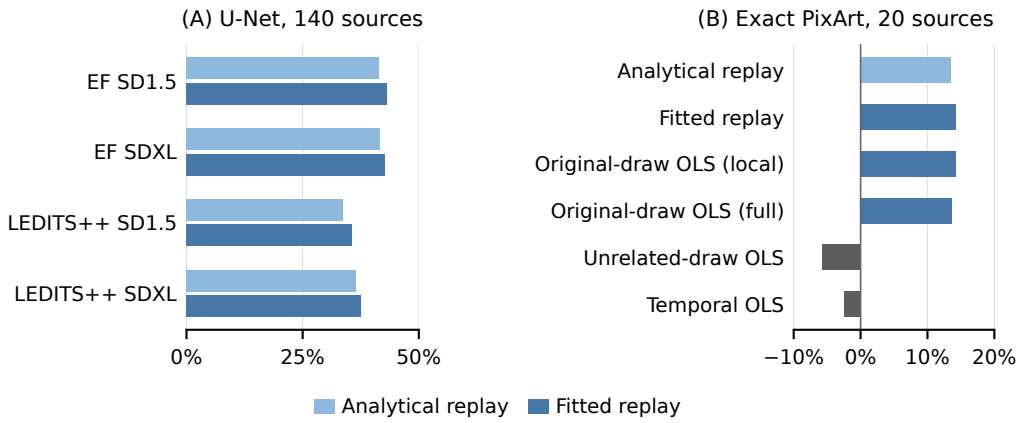

Figure 3: Median paired complete-file savings. A: analytical and fitted replay on 140 sources per U-Net configuration, with a common comparison container and per-source generic reference under native maximum-error caps of 0.01 (EF) and 0.003 (LEDITS++). B: exact PixArt coding on 20 sources, with a common native-lossless reference and XOR/pcodec. Local and full OLS use the original generating draws; unrelated-draw and temporal OLS increase file size.

Skeleton-only coding already outperforms generic coding in all five lossy configurations. Carriers add 10.73–14.32 percentage points of payload saving for EF. LEDITS++'s prescribed two-draw carrier improves on single-draw, shifted, and shuffled controls, whereas mismatched-seed replay increases file size (Appendix Figure 5).

5.4 Exact four-step states

ReNoise predicts corrected noise from its original draws. TurboEdit's shared noising path and model timesteps use different noise scales, motivating a source-shifted predictor that outperforms original-draw predictors on all 140 sources (Appendix F.2). Replaying the shared path saves 37.43 % relative to directly stored native arrays; analytical mean prediction saves a further 1.78 % relative to generic coding that already replays that path (Table 2).

Table 2: Exact four-step coding on 140 sources. Median paired complete-file savings (%) against the listed reference. Generic/analytical files reproduce native inputs and three-prompt pixels after fresh reload (Appendix F.2).

Editor	Replayed information	Reference	Analytical	Fitted
ReNoise	Original noise draws	Native state	10.56	10.85
TurboEdit	Shared noising path	Direct native arrays	37.43	–
TurboEdit	Source means	Generic with path replay	1.78	2.09

5.5 Fidelity and generalization

Editing fidelity. Figure 4 shows metric-selected EF and LEDITS++ edits; Appendix I provides the full galleries and selection rule. On 20 development sources, selecting candidates for each benchmark prompt using measured output LPIPS at a shared cap of 0.01 gives 35.31–42.92 % median complete-file savings across four U-Net configurations (Appendix E.2).

Reusing a state. One stored state serves three fixed prompts per source without re-extraction or request-specific selection (140×3 pairs per configuration). Across four U-Net configurations, median source-wise worst-prompt LPIPS to raw-state replay remains below 0.002, measured on 512-pixel PNGs with SDXL downsampled from 1024 (Appendix E.6). EF tolerance comparisons and PixArt guidance/grid sensitivity are given in Appendices E.3 and H.2.

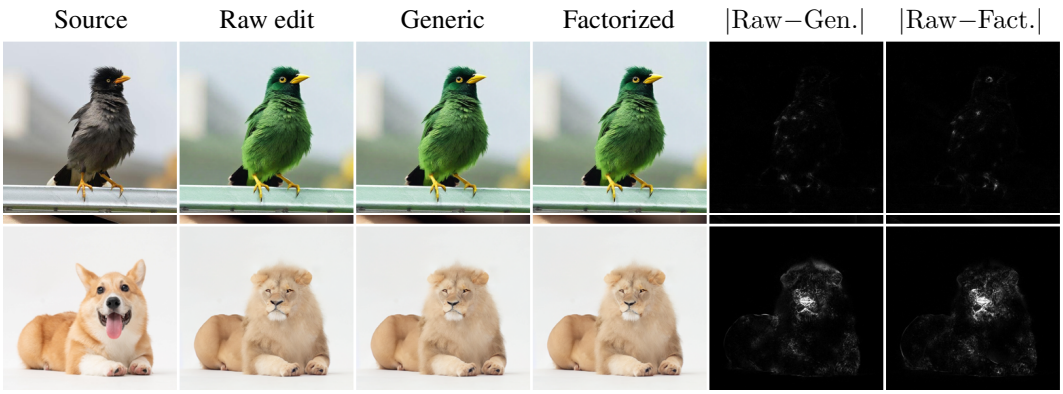

Figure 4: Metric-selected edits: the median ΔL_{mimic} case among 20 EF/SDXL sources (top) and the second-largest case among 40 LEDITS++/SD 1.5 sources (bottom), selected without visual inspection (Appendix I). Differences from raw-state edits are magnified $\times 10$.

Transfer beyond PIE-Bench. Both editors retain 36.77–48.68 % median complete-file savings on SD 1.5 and 35.13–50.15 % on SDXL across fixed-caption adaptations of Emu Edit (100 sources), EditBench (24), and MagicBrush (100) (Appendix H.1). A FlowEdit/SD3 study codes a materialized source-response object reused across target prompts (Appendix H.4).

5.6 Reusable assets and warm-query cost

State reuse avoids repeated inversion; factorization reduces storage. On 60 sources per configuration, base lossy-state queries require 0.379–0.579× the no-store time, with 0.04–1.16 % median paired overhead over generic storage. On 20 sources, ReNoise analytical-state queries require 0.316 s versus 4.012 s from source images, reducing U-Net calls from 44 to 4. Exact PixArt state reuse requires 6.072 s versus 15.442 s from the source image, achieving a two-query crossover on all 20 sources while saving 14.23 % in bytes with only 0.23 % warm-query overhead.

Similarly, TurboEdit uses four U-Net calls from either the source image or stored state; caching its latent and seed occupies 55.51 kB and requires 0.877 s per query (versus 268.63 kB and 0.880 s for analytical state), reproducing native pixels and serving as its most compact reusable asset. Timings use resident models and warm page caches on RTX 4090 GPUs (Appendices G.1 and G.2).

6 Limitations

Replay requires a compatible RNG environment and the recorded state-generation contract. State arrays match across two RTX 4090 devices, and draw streams match across two PyTorch versions (Appendix G.3); portability across GPU architectures remains untested. Optimized embeddings and feature banks need representations matched to those state objects (Appendix H.3). In lossy PixArt, state errors at the shared tolerance produce larger output deviations than in the U-Net editors for both generic and factorized storage, motivating its exact-state operating point. Query timings cover resident models and page-cache-warm reads; deployment costs also depend on preparation, model loading, and storage access.

7 Conclusion

Replay factorization uses generating provenance to predict persistent inversion state, reducing storage across U-Net and Transformer editors. Original-draw, unrelated-draw, and temporal controls identify the information supplied by the generating draws; solver analysis specifies compact predictors whose analytical coefficients retain most fitted savings, including in exact-state coding. ReNoise and TurboEdit extend this construction to optimized noise and shifted source paths. Query costs identify which state to retain for each editor. Generating provenance supplies reproducible structure at the decoder, reducing what must be stored for later editing.

AI USE STATEMENT

Generative AI tools were used solely to assist with language editing, including grammar checking, translation, and refinement of wording for clarity and readability. All technical content, methodology, experiments, results, and conclusions were developed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The appendices provide the solver derivations, codec and container contracts, fixed protocol and decision rules, sensitivity analyses, boundary evaluations, recorded software versions, model revisions, and asset licenses. The anonymized code supplement provides implementations, experiment inputs, synthetic tests, and environment specifications, together with scripts for running and analyzing selected experiments. Datasets and model weights are obtained separately; the analysis scripts consume outputs generated by the experiments.

ETHICS STATEMENT

Persistent inversion state can be used to reconstruct the source image. Appendix H.5 quantifies this property for EF using cache-only source-reconstruction PSNR.

REFERENCES

Soroush Abbasi Koohpayegani, K L Navaneet, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. NOLA: Compressing LoRA using linear combination of random basis. In *The Twelfth International Conference on Learning Representations (ICLR)*, pp. 23610–23626, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/66b99dbf9ed172abac5cb5ccfc82d1e2-Abstract-Conference.html.

Shubham Agarwal, Subrata Mitra, Sarthak Chakraborty, Srikrishna Karanam, Koyel Mukherjee, and Shiv Kumar Saini. Approximate caching for efficiently serving text-to-image diffusion models. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 1173–1189, Santa Clara, CA, 2024. USENIX Association. URL https://www.usenix.org/conference/nsdi24/presentation/agarwal-shubham.

Amirhossein Alimohammadi, Aryan Mikaeili, Sauradip Nag, Negar Hassanpour, Andrea Tagliasacchi, and Ali Mahdavi-Amiri. Cora: Correspondence-aware image editing using few step diffusion. In *ACM SIGGRAPH 2025 Conference Papers*, pp. 1–11. Association for Computing Machinery, 2025. doi: 10.1145/3721238.3730650. URL https://dl.acm.org/doi/10.1145/3721238.3730650.

Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=rkcQFMZRb.

Canberk Baykal, Valentin Deschaintre, Yannick Hold-Geoffroy, Michael Fischer, Anna Frühstück, Cengiz Öztireli, and Iliyan Georgiev. ResEdit: Residual embeddings for precise generative image editing. *Computer Graphics Forum*, 45(4):e70551, 2026. doi: 10.1111/cgf.70551. URL https://diglib.eg.org/handle/10.1111/cgf70551. EGSR 2026.

Zander W. Blasingame and Chen Liu. Rex: A family of reversible exponential (stochastic) Runge-Kutta solvers. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/66025.

Blosc Development Team. A fast, compressed and persistent data store library, 2009–2026. URL https://blosc.org.

Manuel Brack, Felix Friedrich, Katharina Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. LEDITS++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8861–8870, 2024. doi: 10.1109/CVPR52733.2024.00846. URL https://openaccess.

thecvf.com/content/CVPR2024/html/Brack_LEDITS_Limitless_Image_Editing_using_Text-to-Image_Models_CVPR_2024_paper.html.

Guantao Chen, Shikang Zheng, Yuqi Lin, and Linfeng Zhang. Forecast the principal, stabilize the residual: Subspace-aware feature caching for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23632–23641, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_Forecast_the_Principal_Stabilize_the_Residual_Subspace-Aware_Feature_Caching_for_CVPR_2026_paper.html.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations (ICLR)*, pp. 57611–57640, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/fe989bb038b5dcc44181255dd6913e43-Abstract-Conference.html.

Yann Collet and Murray S. Kucherawy. Zstandard compression and the 'application/zstd' media type. RFC 8878, Internet Engineering Task Force (IETF), February 2021. URL https://www.rfc-editor.org/info/rfc8878/.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, second edition, 2006. ISBN 978-0-471-24195-9. doi: 10.1002/047174882X. URL https://www.wiley.com/en-us/Elements+of+Information+Theory%2C+2nd+Edition-p-9780471241959.

Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. FireFlow: Fast inversion of rectified flow for image semantic editing. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13110–13128. PMLR, 2025. URL https://proceedings.mlr.press/v267/deng25c.html.

Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. TurboEdit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–12. Association for Computing Machinery, 2024. doi: 10.1145/3680528.3687612. URL https://dl.acm.org/doi/10.1145/3680528.3687612.

Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. ReNoise: Real image inversion through iterative noising. In *Computer Vision – ECCV 2024*, volume 15072 of *Lecture Notes in Computer Science*, pp. 395–413. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-72630-9_23. URL https://link.springer.com/chapter/10.1007/978-3-031-72630-9_23.

Xiaoxiao He, Quan Dao, Ligong Han, Song Wen, Minhao Bai, Di Liu, Han Zhang, Felix Juefei-Xu, Chaowei Tan, Bo Liu, Martin Renqiang Min, Kang Li, Faez Ahmed, Akash Srivastava, Hongdong Li, Junzhou Huang, and Dimitris N. Metaxas. DICE: Discrete inversion enabling controllable editing for masked generative models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 762–772, 2026. URL https://openaccess.thecvf.com/content/WACV2026/html/He_DICE_Discrete_Inversion_Enabling_Controllable_Editing_for_Masked_Generative_Models_WACV_2026_paper.html.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly DDPM noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12469–12478, 2024. doi: 10.1109/CVPR52733.2024.01185. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huberman-Spiegelglas_An_Edit_Friendly_DDPM_Noise_Space_Inversion_and_Manipulations_CVPR_2024_paper.html.

Guanlong Jiao, Biqing Huang, Kuan-Chieh Wang, and Renjie Liao. UniEdit-Flow: Unleashing inversion and editing in the era of flow models. In *The Fourteenth International Conference on Learning Representations (ICLR)*, pp. 941–977, 2026.

URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/023d94f44110b9a3c62329beec739772-Abstract-Conference.html.

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. In *The Twelfth International Conference on Learning Representations (ICLR)*, pp. 23395–23422, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/661caac7729aa7d8c6b8ac0d39ccbc6a-Abstract-Conference.html.

Mingyu Kang and Yong Suk Choi. Editable noise map inversion: Encoding target-image into noise for high-fidelity image manipulation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 28904–28920. PMLR, 2025. URL https://proceedings.mlr.press/v267/kang25b.html.

Jeongsol Kim, Yeobin Hong, Jonghyun Park, and Jong Chul Ye. FlowAlign: Trajectory-regularized, inversion-free flow-based image editing. In *The Fourteenth International Conference on Learning Representations (ICLR)*, pp. 121814–121846, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c613667b7f11fbf478293266a156aabe-Abstract-Conference.html.

Xianghao Kong, Hansheng Chen, Yuwei Guo, Lvmin Zhang, Gordon Wetzstein, Maneesh Agrawala, and Anyi Rao. Taming flow-based I2V models for creative video editing. arXiv preprint arXiv:2509.21917, 2025. URL https://arxiv.org/abs/2509.21917.

Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19721–19730, 2025. doi: 10.1109/ICCV51701.2025.01834. URL https://openaccess.thecvf.com/content/ICCV2025/html/Kulikov_FlowEdit_Inversion-Free_Text-Based_Editing_Using_Pre-Trained_Flow_Models_ICCV_2025_paper.html.

Peizheng Li, Xin Ai, Hanyuan Liu, Qiange Wang, and Yanfeng Zhang. RegionCache: Semantic-aware region reuse for efficient multi-turn image generation. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-26)*, pp. 1342–1350. International Joint Conferences on Artificial Intelligence Organization, 2026. doi: 10.24963/ijcai.2026/150. URL https://www.ijcai.org/proceedings/2026/150.

Xin Liang, Kai Zhao, Sheng Di, Sihuan Li, Robert Underwood, Ali M. Gok, Jiannan Tian, Junjing Deng, Jon C. Calhoun, Dingwen Tao, Zizhong Chen, and Franck Cappello. SZ3: A modular framework for composing prediction-based error-bounded lossy compressors. *IEEE Transactions on Big Data*, 9(2):485–498, 2023. doi: 10.1109/TBDATA.2022.3201176. URL https://doi.org/10.1109/TBDATA.2022.3201176.

Peter Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, 2014. doi: 10.1109/TVCG.2014.2346458. URL https://doi.org/10.1109/TVCG.2014.2346458.

Zehao Liu, Chuangchuang Fang, and Yang Ren. Activation-weighted seeded residual coding for low-bit LLM weight repair. arXiv preprint arXiv:2608.23144, 2026. URL https://arxiv.org/abs/2608.23144.

Martin Loncaric, Niels Jeppesen, and Ben Zinberg. Pcodec: Better compression for numerical sequences. arXiv preprint arXiv:2502.06112, 2025. URL https://arxiv.org/abs/2502.06112. Code available at https://github.com/pcodec/pcodec.

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. doi: 10.1007/s11633-025-1562-4. URL https://doi.org/10.1007/s11633-025-1562-4.

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

(CVPR), pp. 15762–15772, 2024. doi: 10.1109/CVPR52733.2024.01492. URL https://openaccess.thecvf.com/content/CVPR2024/html/Ma_DeepCache_Accelerating_Diffusion_Models_for_Free_CVPR_2024_paper.html.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047, 2023. doi: 10.1109/CVPR52729.2023.00585. URL https://openaccess.thecvf.com/content/CVPR2023/html/Mokady_NULL-Text_Inversion_for_Editing_Real_Images_Using_Guided_Diffusion_Models_CVPR_2023_paper.html.

Parsa Nooralinejad, Ali Abbasi, Soroush Abbasi Koohpayegani, Kossar Pourahmadi Meibodi, Rana Muhammad Shahroz Khan, Soheil Kolouri, and Hamed Pirsiavash. PRANC: Pseudo RAndom networks for compacting deep models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17021–17031, 2023. doi: 10.1109/ICCV51070.2023.01561. URL https://openaccess.thecvf.com/content/ICCV2023/html/Nooralinejad_PRANC_Pseudo_RAndom_Networks_for_Compacting_Deep_Models_ICCV_2023_paper.html.

Guy Ohayon, Hila Manor, Tomer Michaeli, and Michael Elad. Compressed image generation with denoising diffusion codebook models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 47044–47089. PMLR, 2025. URL https://proceedings.mlr.press/v267/ohayon25a.html.

Yongseong Park, Joeun Kim, HoEun Kim, and Young-Sik Kim. Compression asymmetry and trajectory binding in noise-anchored diffusion inversion, 2026. URL https://arxiv.org/abs/2607.09784.

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 *The Twelfth International Conference on Learning Representations (ICLR)*, pp. 1862–1874, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/081b08068e4733ae3e7ad019fe8d172f-Abstract-Conference.html.

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

Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. In *The Thirteenth International Conference on Learning Representations (ICLR)*, pp. 77330–77365, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c07bc8b520aa87326b015b0da6a70b48-Abstract-Conference.html.

Dvir Samuel, Barak Meiri, Haggai Maron, Yoad Tewel, Nir Darshan, Shai Avidan, Gal Chechik, and Rami Ben-Ari. Lightning-fast image inversion and editing for text-to-image diffusion models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, pp. 38384–38404, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/5f8ccd2cf8ca5876de06104e5f59f902-Abstract-Conference.html.

Rasoul Shafipour, David Harrison, Maxwell Horton, Jeffrey Marker, Houman Bedayat, Sachin Mehta, Mohammad Rastegari, Mahyar Najibi, and Saman Naderiparizi. SeedLM: Compressing LLM weights into seeds of pseudo-random generators. In *The Thirteenth International Conference on Learning Representations (ICLR)*, pp. 12755–12771, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/222a2a46018a1e7b55ba48ba11932d04-Abstract-Conference.html.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu Edit: Precise Image Editing via Recognition and Generation Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8871–8879, 2024. doi: 10.1109/cvpr52733.2024.00847. URL https://openaccess.thecvf.com/content/CVPR2024/html/Sheynin_Emu_Edit_Precise_Image_Editing_via_Recognition_and_Generation_Tasks_CVPR_2024_paper.html.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *The Ninth International Conference on Learning Representations (ICLR)*, 2021. URL https://iclr.cc/virtual/2021/poster/3177.

Desen Sun, Henry Tian, Tim Lu, and Sihang Liu. FlexCache: Flexible approximate cache system for video diffusion. arXiv preprint arXiv:2501.04012, 2024. URL https://arxiv.org/abs/2501.04012.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930, 2023. doi: 10.1109/CVPR52729.2023.00191. URL https://openaccess.thecvf.com/content/CVPR2023/html/Tumanyan_Plug-and-Play_Diffusion_Features_for_Text-Driven_Image-to-Image_Translation_CVPR_2023_paper.html.

Amit Vaisman, Guy Ohayon, Hila Manor, Michael Elad, and Tomer Michaeli. Turbo-DDCM: Fast and flexible zero-shot diffusion-based image compression. In *The Fourteenth International Conference on Learning Representations (ICLR)*, pp. 151475–151520, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f58c24798220ba724fe05c0fa786227d-Abstract-Conference.html.

Bram Wallace, Akash Gokul, and Nikhil Naik. EDICT: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22532–22541, 2023. doi: 10.1109/CVPR52729.2023.02158. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wallace_EDICT_Exact_Diffusion_Inversion_via_Coupled_Transformations_CVPR_2023_paper.html.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J. Fleet, Radu Soricut, Jason Baldridge, Mohammad Norouzi, Peter Anderson, and William Chan. Imagen Editor and EditBench: Advancing and Evaluating Text-Guided Image Inpainting. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18359–18369, 2023. doi: 10.1109/cvpr52729.2023.01761. URL https://openaccess.thecvf.com/content/CVPR2023/html/Wang_Imagen_Editor_and_EditBench_Advancing_and_Evaluating_Text-Guided_Image_Inpainting_CVPR_2023_paper.html.

Zirui Wang, Yunjia Zheng, Tingfeng Lan, Zhaoyuan Su, Haoran Ni, Juncheng Yang, and Yue Cheng. LatentBox: Storing AI-generated images at scale via a latent-first design. arXiv preprint arXiv:2605.19385, 2026. URL https://arxiv.org/abs/2605.19385.

Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7378–7387, 2023. doi: 10.1109/ICCV51070.2023.00678. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wu_A_Latent_Space_of_Stochastic_Diffusion_Models_for_Zero-Shot_Image_ICCV_2023_paper.html.

Chenxi Xie, Minghan Li, Shuai Li, Yuhui Wu, Qiaosi Yi, and Lei Zhang. DNAEdit: Direct noise alignment for text-guided rectified flow editing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 124456–124474, 2025. doi: 10.52202/085713-4147. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b44ae90136013a8d0e2d24f6015b6097-Abstract-Conference.html.

Desong Yang and Mang Ye. DirectEdit: Step-level accurate inversion for flow-based image editing. In *Forty-third International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/66686.

Yibo Yang, Stephan Mandt, and Lucas Theis. An introduction to neural data compression. *Foundations and Trends in Computer Graphics and Vision*, 15(2):113–200, 2023. doi: 10.1561/0600000107. URL https://www.nowpublishers.com/article/Details/CGV-107.

Zihao Yu, Haoyang Li, Fangcheng Fu, Xupeng Miao, and Bin Cui. Accelerating text-to-image editing via cache-enabled sparse diffusion inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):16605–16613, 2024. doi: 10.1609/aaai.v38i15.29599. URL https://ojs.aaai.org/index.php/AAAI/article/view/29599.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems*, volume 36, pp. 31428–31449, 2023. doi: 10.52202/075280-1365. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/64008fa30cba9b4d1ab1bd3bd3d57d61-Abstract-Datasets_and_Benchmarks.html.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Tianrui Zhu, Shiyi Zhang, Jiawei Shao, and Yansong Tang. KV-Edit: Training-free image editing for precise background preservation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16607–16617, 2025. doi: 10.1109/ICCV51701.2025.01542. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhu_KV-Edit_Training-Free_Image_Editing_for_Precise_Background_Preservation_ICCV_2025_paper.html.

A Derivations and Proofs

The derivations connect the source-cancellation condition to the predictors used by each editor. Corollary 2 follows the DPM-Solver convention, in which α_ℓ and σ_ℓ denote the signal and noise scales of the auxiliary path.

A.1 Normal form, stencil consistency, and the two solver instances

For the order-p notation, f_{t+j} denotes the model prediction and the affine update is

$$x_{t-1} = A_t x_t + \sum_{j=0}^{p-1} B_{t,j} f_{t+j} + \sigma_t z_t^{\mathrm{sol}}. \tag{11}$$

The general normal form in Equation 4 also accommodates multiple latent-state weights after converting data predictions to noise predictions.

Proof of Lemma 1. Fix a row with $\varsigma_u \neq 0$. Rearranging Equation 4 gives

$$z_u^{\mathrm{sol}} = \varsigma_u^{-1}\Big(x_u - \sum_{\ell\in J_u} A_{u,\ell} x_\ell - \sum_{\ell\in J_u} B_{u,\ell}\varepsilon_\theta(x_\ell)\Big). \tag{12}$$

Substituting $x_\ell = a_\ell x_0 + b_\ell \xi_\ell$ for every $\ell \in J_u \cup \{u\}$ and collecting the terms in x_0, in the draws, and in the predictions gives

$$z_u^{\mathrm{sol}} = \varsigma_u^{-1}\Big(a_u - \sum_{\ell\in J_u} A_{u,\ell} a_\ell\Big) x_0 + \varsigma_u^{-1}\Big(b_u\xi_u - \sum_{\ell\in J_u} A_{u,\ell} b_\ell \xi_\ell\Big) - \varsigma_u^{-1}\sum_{\ell\in J_u} B_{u,\ell}\varepsilon_\theta(x_\ell). \tag{13}$$

The first term vanishes for every source latent if and only if its scalar coefficient vanishes, which is Equation 5. If the solver is stated in data predictions, so that $x_u = \sum_\ell A'_{u,\ell} x_\ell + \sum_\ell C_{u,\ell} D_\ell + \varsigma_u z_u^{\mathrm{sol}}$ with $D_\ell = (x_\ell - b_\ell \varepsilon_\theta(x_\ell))/a_\ell$ and $a_\ell \neq 0$, then expanding D_ℓ yields the normal form with $A_{u,\ell} = A'_{u,\ell} + C_{u,\ell}/a_\ell$ and $B_{u,\ell} = -C_{u,\ell} b_\ell / a_\ell$, so Equation 5 must be evaluated with the converted weights. □

Remark (Affinity and stencil consistency). Equation 5 is not implied by affinity. For the VP-SDE, the reverse-diffusion predictor of Song et al. (2021),

$$x_{i-1} = (2 - \sqrt{1-\beta_i})x_i + \beta_i s_\theta(x_i) + \sqrt{\beta_i}\, z,$$

is affine in the latent and score terms but violates Equation 5 for every $0 < \beta_i < 1$. Writing $u_i = \sqrt{1-\beta_i}$ and $a_{i-1} = a_i/u_i$, the residual source coefficient in the extracted control is $a_i(1-u_i)^2/(u_i\sqrt{\beta_i}) = O(\beta_i^{3/2})$. Their ancestral predictor,

$$x_{i-1} = \frac{x_i + \beta_i s_\theta(x_i)}{\sqrt{1-\beta_i}} + \sqrt{\beta_i}\, z,$$

satisfies Equation 5.

Equation 5 characterizes cancellation of the explicit source coefficient. Without cancellation, $z_u^{\mathrm{sol}} - q_u$ remains a storable residual that includes the source term.

Corollary 1 (Edit-Friendly, order one)**.** *The Edit-Friendly auxiliary path sets $x_t = \sqrt{\bar\alpha_t}\, x_0 + \sqrt{1-\bar\alpha_t}\, \xi_t$ with independent draws, and the extracted solver control is $z_t^{\mathrm{sol}} = (x_{t-1} - \mu_t)/\sigma_t$ with*

$$\mu_t = \sqrt{\bar\alpha_{t-1}}\, \hat x_0 + \sqrt{1 - \bar\alpha_{t-1} - \sigma_t^2}\, \varepsilon_\theta(x_t),$$
$$\hat x_0 = \frac{x_t - \sqrt{1-\bar\alpha_t}\, \varepsilon_\theta(x_t)}{\sqrt{\bar\alpha_t}}, \qquad \sigma_t^2 = \frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}\Big(1 - \frac{\bar\alpha_t}{\bar\alpha_{t-1}}\Big)$$

at $\eta = 1$. Then $u = t-1$ and $J_u = \{t\}$, $A_t = \sqrt{\bar\alpha_{t-1}/\bar\alpha_t}$, $B_{t,0} = \sqrt{1-\bar\alpha_{t-1}-\sigma_t^2} - \sqrt{\bar\alpha_{t-1}(1-\bar\alpha_t)/\bar\alpha_t}$, $a_t = \sqrt{\bar\alpha_t}$ and $b_t = \sqrt{1-\bar\alpha_t}$. Equation 5 holds on every row whose target state is a seeded state, and Equation 6 follows with $\kappa_t = -B_{t,0}/\sigma_t$. The lowest-noise row targets x_0 itself, so cancellation requires $\sqrt{\bar\alpha_{\mathrm{final}}} = 1$. Otherwise the explicit source term remains; when equality holds, the injected noise scale is zero and the normalized-control formula is degenerate. The evaluated formats keep this lowest-noise stored row outside the factorization.

Proof. Substituting $\hat{x}_0$ into μ_t gives $\mu_t = A_t x_t + B_{t,0}\varepsilon_\theta(x_t)$ with the stated A_t and $B_{t,0}$, so the row is in the normal form of Equation 4 with $\varsigma_u = \sigma_t$ and $J_u = \{t\}$. Equation 5 then reads $\sqrt{\bar{\alpha}_{t-1}} = \sqrt{\bar{\alpha}_{t-1}/\bar{\alpha}_t} \cdot \sqrt{\bar{\alpha}_t}$, an identity, and Lemma 1 gives Equation 6. For the lowest-noise row, $a_u = 1$ and $b_u = 0$ while $\sum_\ell A_{u,\ell} a_\ell = \sqrt{\bar{\alpha}_{\text{final}}}$, so cancellation requires $\bar{\alpha}_{\text{final}} = 1$, which also makes $\sigma_t = 0$. □

We evaluate κ_t from these coefficients on the SD 1.5 and SDXL scaled-linear schedule (β from 0.00085 to 0.012) and on the PixArt-α linear schedule (β from 10^{-4} to 0.02); each of the 63 factorized rows contributes one carrier column, so the coefficient block holds 63 float32 scalars in 252 B.

Corollary 2 (LEDITS++ with the second-order midpoint rule)**.** *At seeded solver points, use $\alpha_\ell = a_\ell$ and $\sigma_\ell = b_\ell$ for the source-path signal and noise scales, and ς for the injected-control scale. Let the current, preceding, and next solver points s, s', u carry the draws $\xi_j, \xi_{j+1}, \xi_{j-1}$, with draws indexed in increasing-noise order, and set $\lambda_\ell = \log(\alpha_\ell/\sigma_\ell)$, $h = \lambda_u - \lambda_s$, $r_s = (\lambda_s - \lambda_{s'})/h$,*

$$K = \alpha_u(1 - e^{-2h}), \quad \varsigma = \sigma_u\sqrt{1 - e^{-2h}}, \quad g = \frac{K\sigma_s}{2r_s\alpha_s}, \quad g' = \frac{K\sigma_{s'}}{2r_s\alpha_{s'}}.$$

For the update

$$x_u = \frac{\sigma_u}{\sigma_s}e^{-h}x_s + K\,D_s + \frac{K}{2r_s}\big(D_s - D_{s'}\big) + \varsigma\, z_j^{\text{sol}}, \qquad D_\ell = x_0 + \frac{\sigma_\ell}{\alpha_\ell}\big(\xi_\ell - \varepsilon_\theta(x_\ell)\big),$$

Equation 5 holds on every interior row, and

$$\varsigma\, z_j^{\text{sol}} = \underbrace{\sigma_u\xi_{j-1} - \Big(\frac{\alpha_u\sigma_s}{\alpha_s} + g\Big)\xi_j + g'\xi_{j+1}}_{\varsigma\, q_j} + \underbrace{\Big(\frac{K\sigma_s}{\alpha_s} + g\Big)\varepsilon_\theta(x_s) - g'\varepsilon_\theta(x_{s'})}_{\varsigma\, m_j},$$

so the indexed carrier is $[\xi_j, \xi_{j+1}]$. The initial row is the case $g = g' = 0$ and has a single column; the terminal boundary remains outside the factorization.

Proof. Since $h = \lambda_u - \lambda_s$, we have $e^{-h} = \sigma_u\alpha_s/(\alpha_u\sigma_s)$ and $e^{-2h} = \sigma_u^2\alpha_s^2/(\alpha_u^2\sigma_s^2)$. In the data-prediction form the latent weight is $A'_{u,s} = (\sigma_u/\sigma_s)e^{-h}$ and the prediction weights are $C_{u,s} = K + K/(2r_s)$ and $C_{u,s'} = -K/(2r_s)$, while the auxiliary path has $a_\ell = \alpha_\ell$ and $b_\ell = \sigma_\ell$. The converted sum of Equation 5 is therefore

$$\sum_{\ell \in J_u} A_{u,\ell}a_\ell = A'_{u,s}\alpha_s + C_{u,s} + C_{u,s'} = \frac{\sigma_u^2\alpha_s^2}{\alpha_u\sigma_s^2} + K = \alpha_u e^{-2h} + \alpha_u(1 - e^{-2h}) = \alpha_u = a_u,$$

so the condition holds. Substituting the auxiliary path and the two data predictions into $\varsigma z_j^{\text{sol}} = x_u - \mu$, where μ collects the remaining update terms, gives the stated skeleton and model term using

$$\sigma_u e^{-h} + \frac{K\sigma_s}{\alpha_s} = \frac{\alpha_u\sigma_s}{\alpha_s}.$$

The predictions at s and s' correspond to ξ_j and ξ_{j+1}, so $\Xi_j = [\xi_j, \xi_{j+1}]$. On the initial row the multistep term is absent: $C_{u,s'} = 0$, $C_{u,s} = K$, and $g = g' = 0$. On the terminal row, the target is x_0, so $a_u = 1$, $b_u = 0$, and $\sum_\ell A_{u,\ell}a_\ell = \alpha_u$. If $\alpha_u < 1$, Equation 5 fails; otherwise the final noise scale and ς vanish. The terminal row therefore remains outside the factorization. □

At $T = 50$ the interior rows therefore contribute one carrier column on the initial row and two columns on each of the remaining 48 factorized rows, that is 97 float32 scalars occupying 388 B.

Solver coefficients. Corollary 2 gives the constants for the recorded second-order midpoint implementation that generates the stored controls and reconstructs the schedule. Applying the factorization to another solver variant requires deriving its coefficients and verifying Equation 5.

A.2 Affine update and replay coordinates

For EF and LEDITS++ interior rows with nonzero injected noise, Corollaries 1 and 2 give $z_t^{\mathrm{sol}} = q_t + m_t$ in exact arithmetic. The stencil indices of the model predictions, $\mathcal{I}_t = (i_1, \ldots, i_k)$, determine the carrier $\Xi_t = [\xi_{i_1}, \ldots, \xi_{i_k}]$ and the projection in Equation 7.

In the lossy factorized formats, boundary rows are serialized separately in fp16, while the endpoint retains its native dtype. The EF and PixArt-α implementations set these boundary rows to zero, so fp16 storage preserves their values; native-fp16 LEDITS++ boundaries are also preserved exactly. For EF/SD 1.5, the boundary occupies $16{,}384 \times 2 = 32{,}768$ B. The exact-state format preserves boundary bits using a zero predictor and a lossless whole-array XOR stream (Appendix F.1). In the lossy formats, separate boundary storage imposes a fixed cost that becomes more prominent on shorter trajectories.

For a D-element row and a full-rank k-column carrier, coefficient fitting incurs $O(Dk^2 + k^3)$ time complexity and reconstruction requires $O(Dk)$ operations; both scale linearly with row size for fixed $k \leq 2$.

Lag-one correlation. Equation 6 also quantifies the negative correlation between consecutive Edit-Friendly controls observed by Huberman-Spiegelglas et al. (2024). Consecutive controls share ξ_t with weights of nearly equal magnitude and opposite sign, giving a correlation close to $-1/2$ away from the lowest-noise rows. Across all 140 EF/SD 1.5 sources and 8,680 row pairs, the measured median is -0.499 (P5 -0.509, P95 -0.487). The lowest-noise pair departs from this value: its measured correlation is -0.473, versus -0.546 for the skeleton alone, because the model term has a non-negligible influence in this regime.

A.3 Predictors for four-step states

TurboEdit's shifted path. Let s be the source latent and $y_i = a_i s + b_i \xi_i$ its noising path. For the four model timesteps $(799, 599, 399, 199)$, the path uses noise levels $(599, 399, 199, 0, 0)$, with $y_3 = y_4 = s$. The noise-prediction DDPM mean is

$$\mu_i = A_i y_i + B_i \varepsilon_\theta(y_i, t_i).$$

The native editor retains these means and uses residuals $z_i = y_{i+1} - \mu_i$. Applying the expansion in Lemma 1, with injected scale one, gives

$$z_i = \gamma_i s + q_i - B_i \varepsilon_\theta(y_i, t_i), \qquad \gamma_i = a_{i+1} - A_i a_i, \tag{14}$$

$$q_i = b_{i+1}\xi_{i+1} - A_i b_i \xi_i. \tag{15}$$

The model-timestep scales $\bar{a}_i, \bar{b}_i$ differ from the path scales a_i, b_i. The noise target that writes this same input as $y_i = \bar{a}_i s + \bar{b}_i \eta_i$ is

$$\eta_i = \frac{(a_i - \bar{a}_i)s + b_i \xi_i}{\bar{b}_i}.$$

Substituting this target for the model prediction yields the analytical residual and mean predictors

$$p_i^z = \gamma_i s + q_i - B_i \eta_i, \qquad p_i^\mu = y_{i+1} - p_i^z.$$

Thus the solver coefficients determine both the explicit source term and the noise target appropriate to the shifted path. The final p_i^z is zero in exact arithmetic: substituting η_i makes the implied clean estimate equal to s at every step, and the final mean, whose target noise level is zero, returns this estimate. For fitted coding, the carrier consists of the source latent and all three generating draws. All predictors are applied to the materialized mean arrays; exact coding preserves their finite-precision values and the native residual clipping rule.

ReNoise's corrected draws. ReNoise initializes each stochastic noise row with a draw ξ_i and updates it during inversion to $n_i = \xi_i + \Delta_i$. Here Δ_i is the accumulated noise-optimization correction. Its analytical predictor is $p_i = \xi_i$; a fitted predictor is $p_i = c_i \xi_i$, where $c_i = \langle \xi_i, n_i \rangle / \|\xi_i\|^2$. The coded remainder is respectively Δ_i or $n_i - c_i \xi_i$. The final zero-noise row and native endpoint are stored exactly. This construction uses the recorded generating draws without evaluating the denoiser at decoding.

A.4 PROJECTION IDENTITIES

Remark 1 (Consistency of the stored representation)**.** *For each non-boundary row, matching provenance and exact preservation of c_t and δ_t reconstruct the materialized row in exact arithmetic. If the boundary rows included in the payload are also preserved exactly, their restoration completes S.*

Verification of Remark 1. The decoder evaluates $q_t + \Xi_t c_t + \delta_t$. Substituting $\delta_t = r_t - \Xi_t c_t$ gives $q_t + r_t = z_t$; boundary rows are restored exactly under the premise. □

Proposition 2 (Projected prediction deviations)**.** *For a factorized interior row whose stored control is the solver's output, write $r_t := z_t - q_t = \beta_t + \sum_j w_{tj}\widehat{p}_{tj}$, where w_{tj} are solver weights, $\widehat{p}_{tj}$ are network predictions, and q_t, Ξ_t are reproducible from the seed and the replay inputs already available to the decoder. Suppose the sampled targets $p^\star_{tj}$ satisfy $\beta_t + \sum_j w_{tj}p^\star_{tj} \in \mathrm{col}(\Xi_t)$, and let $P_t = \Xi_t\Xi_t^\dagger$. In exact arithmetic, uncentered least squares gives $c_t = \Xi_t^\dagger r_t$, $z_t = q_t + \Xi_t c_t + \delta_t$, and $\delta_t = (I - P_t)\sum_j w_{tj}\big(\widehat{p}_{tj} - p^\star_{tj}\big)$.*

Proof. Least squares projects r_t onto $\mathrm{col}(\Xi_t)$, so $\delta_t = (I - P_t)r_t$. The target combination $\beta_t + \sum_j w_{tj}p^\star_{tj}$ lies in this column space and is annihilated by $I - P_t$; subtracting it from r_t yields the stated identity. □

The factorized EF and LEDITS++ rows produced by the solver satisfy this identity through their solver decompositions, with one and up to two carrier columns, respectively. The initial LEDITS++ multistep row uses a first-order rule with one prediction and one carrier, so its target combination lies in the single-column span. For EF, $w_t = \kappa_t$, $\widehat{p}_t = \varepsilon_\theta(x_t)$, and $p^\star_t = \xi_t$, which recovers the formulation presented in Section 4.2. The two interior LEDITS++/SDXL rows overwritten to zero after inversion are an exception (Appendix A.7): their stored z_t is not the solver output, so Proposition 2 does not apply. The reconstruction identity still recovers the stored zero in exact arithmetic, but $\delta_t = -(I - P_t)q_t$ has no prediction-deviation interpretation. The implementation also incurs rounding in the seed term, fitted coefficients, and residual arithmetic. For FlowEdit's source-response object, the corresponding representation uses the pathwise target $\xi_\sigma - x_{\mathrm{src}}$ and a single carrier; its measurements are reported separately in Appendix H.4.

Remark 3 (Skeleton components inside the carrier span)**.** *In exact arithmetic, replacing q_t by $q_t + \Xi_t d$ leaves the defect unchanged because $(I - P_t)\Xi_t = 0$. The fitted coefficient becomes $c_t - \Xi_t^\dagger\Xi_t d$, or $c_t - d$ for a full-rank carrier, with the same reconstructed row. Thus the defect depends on the skeleton only through $(I - P_t)q_t$. A deterministic inner codec at fixed settings produces the same defect bitstream; if the coefficient count and fixed-width format, boundary rows, endpoint, and metadata are also unchanged, the idealized payload length is identical. Finite-precision predictor and residual arithmetic can change the computed defect and its bitstream (Remark 1).*

For EF, $q_t = (b_{t-1}\xi_{t-1} - A_t b_t \xi_t)/\sigma_t$ with $\Xi_t = [\xi_t]$, so $(I - P_t)q_t = (b_{t-1}/\sigma_t)(I - P_t)\xi_{t-1}$; for LEDITS++, Corollary 2 gives $\Xi_j = [\xi_j, \xi_{j+1}]$ and $(I - P_j)q_j = (\sigma_u/\varsigma)(I - P_j)\xi_{j-1}$. In both editors, only the carrier-orthogonal component of one draw, weighted by the solver, contributes to the defect. The other skeleton terms lie in $\mathrm{col}(\Xi_t)$ and can be absorbed into the fitted coefficients. The solver determines this draw's weight without storing an additional coefficient; direct local-seed regression fits and stores it instead (Proposition 5). The alignment controls in Appendix D.4 perturb this weighted, carrier-orthogonal component.

Projection derivation. The orthogonal decomposition $r_t - \Xi_t c = \delta_t + \Xi_t(c_t - c)$ gives $\|r_t - \Xi_t c\|_2^2 = \|\delta_t\|_2^2 + \|\Xi_t(c_t - c)\|_2^2 \geq \|\delta_t\|_2^2$. Thus δ_t is orthogonal to the columns of Ξ_t and satisfies $\|\delta_t\|_2 = \min_c \|r_t - \Xi_t c\|_2$. This establishes minimum residual energy over the recorded carrier span.

A.5 CARRIER COEFFICIENT AND THE NOISE PREDICTOR

Consider the joint law $x_t = a_t x_0 + b_t \xi_t$, where $x_0 \sim p_{\mathrm{data}}$ is independent of $\xi_t \sim \mathcal{N}(0, I)$. Expectations in this subsection are over this joint law. Let $\varepsilon^*(x_t) = \mathbb{E}[\xi_t \mid x_t]$ be the ideal MMSE noise predictor. Define the population least-squares coefficient

$$c_t^{\mathrm{pop}} := \arg\min_{c \in \mathbb{R}} \mathbb{E}\|\kappa_t \varepsilon^*(x_t) - c\xi_t\|^2 = \kappa_t \frac{\mathbb{E}\langle \xi_t, \varepsilon^*(x_t)\rangle}{\mathbb{E}\|\xi_t\|^2}.$$

For vector-valued random variables, $\mathrm{Var}(Y) = \mathrm{tr}\,\mathrm{Cov}(Y)$ denotes total variance. The tower property gives $\mathbb{E}\langle \xi_t, \varepsilon^*(x_t)\rangle = \mathbb{E}\langle \mathbb{E}[\xi_t \mid x_t], \varepsilon^*(x_t)\rangle = \mathbb{E}\|\varepsilon^*(x_t)\|^2$; for zero-mean noise and $\kappa_t \neq 0$, this yields $c_t^{\mathrm{pop}}/\kappa_t = \mathbb{E}\|\varepsilon^*(x_t)\|^2/\mathbb{E}\|\xi_t\|^2 = \mathrm{Var}(\varepsilon^*(x_t))/\mathrm{Var}(\xi_t)$.

For unmodified solver-produced EF rows in exact arithmetic, $r_t = m_t = \kappa_t \varepsilon_\theta(x_t)$ and $(I - P_t)\xi_t = 0$ yield

$$\delta_t = (I - P_t)m_t = \kappa_t(I - P_t)(\varepsilon_\theta(x_t) - \xi_t).$$

This projected-error identity holds rowwise for an arbitrary predictor; the MMSE assumption is needed only for the population coefficient interpretation. Under classifier-free guidance, the convention $\varepsilon_\theta^{\mathrm{cfg}} = \varepsilon_u + s(\varepsilon_c - \varepsilon_u)$ gives the pathwise decomposition

$$\varepsilon_\theta^{\mathrm{cfg}} - \xi_t = (\varepsilon_c - \xi_t) + (s - 1)(\varepsilon_c - \varepsilon_u).$$

The second term is guidance extrapolation. Neither this guided predictor nor a coefficient fitted spatially to one stored state inherits the population MMSE interval $[0, 1]$.

The relation $\varepsilon_\theta - \xi_t = \sqrt{\bar\alpha_t/(1 - \bar\alpha_t)}(x_0 - \hat x_0)$, where $\hat x_0$ is the clean-source estimate induced by $\varepsilon_\theta(x_t)$, explains the low-noise amplification. On 140 sources per configuration, coefficient signs agree with κ_t on all 63 decomposed rows. The 64 stored rows are the lowest-noise part of the 100-step grid, including the boundary.

Projection ensures $\|\delta_t\|_2 \leq \|r_t\|_2$, whereas skeleton removal need not reduce energy:

$$\|r_t\|_2^2 = \|z_t - q_t\|_2^2 = \|z_t\|_2^2 + \|q_t\|_2^2 - 2\langle z_t, q_t\rangle.$$

A large skeleton and model term can cancel in the stored control. The further identity $\delta_t = (I - P_t)z_t$ requires $q_t \in \mathrm{col}(\Xi_t)$. On every interior EF row, q_t contains the term $(b_{t-1}/\sigma_t)\xi_{t-1}$ outside the one-column carrier, so this membership does not hold.

Corollary 2 gives the LEDITS++ skeleton and model term. On second-order rows, the skeleton combines three local draws, while the carrier contains the two draws associated with the current and preceding model predictions, $[\xi_j, \xi_{j+1}]$. For coefficient diagnostics, the solver weights are $(K\sigma_s/\alpha_s + g)/\varsigma$ and $-g'/\varsigma$, using the corollary's notation. The first reverse-solver row has $g = g' = 0$ and one carrier column.

On the same 140 sources, fitted coefficient signs agree with the solver weights on all 49 rows for the first column and all 48 rows for the second column in both configurations; the second coefficient is negative. The medians of the row-median ratios are 0.89/0.90 (SD 1.5) and 0.90/0.91 (SDXL), reaching 0.99 at the highest-noise rows. The LEDITS++ lowest-noise ratios (first/second column) are 0.33/0.43 on SD 1.5 and 1.04/1.00 on SDXL. The EF and PixArt-α ratios c_t/κ_t approach one on the highest-noise stored rows. This diagnostic explains the second carrier column that is absent from the order-one arm in Figure 5; its fitted coefficients remain in the bitstream. At the lowest-noise decomposed LEDITS++/SDXL row, one of the two zeroed interior rows, post-inversion zeroing sets the stored control to zero, giving $r_t = -q_t$. The ratios on that row describe how the carrier fits the skeleton (Appendix A.7).

A.6 Rate relations

Proposition 4 (High-rate saving)**.** *Let X have a continuous density f on $\mathbb{R}$ with finite differential entropy $h(X)$, and let X^Δ denote X quantized to cells of width Δ, under the regularity conditions of Cover & Thomas (2006, Thm. 8.3.1). Then $H(X^\Delta) = h(X) - \log_2 \Delta + o(1)$ as $\Delta \to 0$. In particular, for $X \sim \mathcal{N}(0, \sigma^2)$ and $\Delta = 2\varepsilon$,*

$$H(X^\Delta) = \log_2 \frac{\sigma\sqrt{2\pi e}}{2\varepsilon} + o(1),$$

and for two such variables with standard deviations σ_z and σ_δ the rates differ by $\frac{1}{2}\log_2(\sigma_z^2/\sigma_\delta^2) + o(1)$, independently of ε.

Proof. The first statement is the standard high-rate relation (Cover & Thomas, 2006, Thm. 8.3.1). For the Gaussian, $h(X) = \frac{1}{2}\log_2(2\pi e\sigma^2)$, which gives the stated rate with $\Delta = 2\varepsilon$; subtracting the expressions for σ_z and σ_δ cancels the term in ε and leaves $\frac{1}{2}\log_2(\sigma_z^2/\sigma_\delta^2)$. □

At a fixed quantization tolerance, Proposition 4 relates the variance reduction to an asymptotic scalar rate saving. The storage experiments measure this effect with finite-rate codecs and count coefficients, boundary rows, endpoints, and metadata in the complete files.

Proposition 5 (Local-seed regression). *Let $\Xi \in \mathbb{R}^{D\times k}$ with $D > k$ have full column rank, let $\Psi = [\Xi, \xi']$ with $\xi' \in \mathbb{R}^D$, and let $q \in \mathrm{col}(\Psi)$. Let $z = q + \Xi c + \delta$ with $\delta \perp \mathrm{col}(\Xi)$, let $a = \Psi^\dagger z$, and let $r = z - \Psi a$. Then*

(i) $r = (I - P_\Psi)\delta$ and $\|r\| \le \|\delta\|$, where $P_\Psi = \Psi\Psi^\dagger$; and

(ii) if $\xi' \sim \mathcal{N}(0, I_D)$ is independent of (Ξ, δ), then $\mathbb{E}\big[\|r\|^2 \mid \Xi, \delta\big] = \big(1 - \frac{1}{D-k}\big)\|\delta\|^2$.

Proof. (i) The normal equations give $r = (I - P_\Psi)z$. Since $q \in \mathrm{col}(\Psi)$ and $\Xi c \in \mathrm{col}(\Xi) \subseteq \mathrm{col}(\Psi)$, we have $(I - P_\Psi)(q + \Xi c) = 0$, so $r = (I - P_\Psi)\delta$; and $\|(I - P_\Psi)\delta\| \le \|\delta\|$ because $I - P_\Psi$ is an orthogonal projector.

(ii) Condition on (Ξ, δ). Let $W = \mathrm{col}(\Xi)^\perp$, so $\dim W = D - k$, and let $w = (I - P_\Xi)\xi'$, which is almost surely nonzero. By the Gram–Schmidt construction, $\mathrm{col}(\Psi) = \mathrm{col}(\Xi) \oplus \mathrm{span}(w)$, hence $P_\Psi = P_\Xi + uu^\top$ with $u = w/\|w\|$. Because $\delta \in W$ we have $P_\Xi\delta = 0$, so $r = \delta - \langle\delta, u\rangle u$ and $\|r\|^2 = \|\delta\|^2 - \langle\delta, u\rangle^2$. Since ξ' is standard Gaussian and independent of (Ξ, δ), conditionally on (Ξ, δ) the vector w is standard Gaussian on W, so u is uniformly distributed on the unit sphere of W. Rotational symmetry within W gives $\mathbb{E}[\langle\delta, u\rangle^2 \mid \Xi, \delta] = \|\delta\|^2/\dim W$, which is $\|\delta\|^2/(D-k)$. □

For EF, take $\Xi = [\xi_t]$ and $\xi' = \xi_{t-1}$: in the exact-arithmetic auxiliary-path model of Corollary 1, δ_t depends on the source latent and ξ_t, while ξ_{t-1} is an independent draw. For second-order interior LEDITS++ rows, $\Xi = [\xi_j, \xi_{j+1}]$ and $\xi' = \xi_{j-1}$. Under the i.i.d. Gaussian-draw idealization, part (ii) therefore applies with $k = 1$ and $k = 2$, respectively. The initial first-order LEDITS++ row uses one carrier and hence $k = 1$. By Remark 3, direct regression fits the weight of this additional draw instead of taking it from the solver. For the SD-shaped controls, $D = 16{,}384$ gives an expected removed-energy fraction $1/(D-k)$ of about 6×10^{-5}. Substituting this expected energy ratio into the Gaussian high-rate approximation gives a nominal change of about 4.4×10^{-5} bits per coordinate, at the cost of one additional float32 coefficient per interior row. The proposition concerns expected residual energy under an idealized draw model; realized file sizes also depend on the codec, coefficient and metadata costs, finite precision, and admission. Appendix D.2 measures finite-precision file sizes and editing fidelity for direct local-seed regression and a wider seed window, complementing this expected-energy calculation.

The independence hypothesis fails on rows that the editor overwrites with zero after inversion (Appendix A.7). There $z_t = 0$, so the stored defect is $\delta_t = -(I - P_t)q_t$; writing $q_t = a\xi' + \Xi_t b$ gives $\delta_t = -a(I - P_t)\xi'$, a deterministic function of ξ', while the local-seed residual $(I - P_{\Psi_t})z_t$ is exactly zero. The $1/(D-k)$ estimate does not apply to these rows; LEDITS++/SDXL has 2 such interior rows among its 49. The comparison in Appendix D.2 uses a common boundary and signed-zero policy and accounts for coefficients and metadata across representations.

Remark 6 (Mismatched-seed replay). *The mismatched-seed control regenerates both the skeleton and the carrier from an unrelated seed and then fits the carrier following the standard procedure, so its coded defect is $\delta' = (I - P_{\Xi'})(z_t - q')$, with q' the mismatched skeleton and Ξ' the mismatched carrier. Writing $q' = a\xi'' + \Xi' b$ for its components outside and inside the carrier span gives $\delta' = (I - P_{\Xi'})z_t - a(I - P_{\Xi'})\xi''$, so the part of the mismatched skeleton lying in the carrier span is removed by the fit and does not contribute to the residual. If Ξ' and ξ'' are independent standard Gaussian draws independent of z_t, then $\mathbb{E}[\|\delta'\|^2 \mid z_t] = (1 - k/D)\|z_t\|^2 + a^2(D-k)$. For nonzero interior rows with $\rho_r \ll 1$, $\|q_t\|^2 \approx \|z_t\|^2$ (Table 3), so if $a^2(D-k)$ is of the order of $\|z_t\|^2$, Proposition 4 estimates the rate increase over generic coding to be on the order of $\frac{1}{2}$ bit per element, which corresponds to several percent under the LEDITS++ relative tolerance. This row-dependent heuristic uses the independent-Gaussian model above and approximates $\|q_t\|^2$ by $\|z_t\|^2$. Among SDXL's 49 interior rows, the 2 rows that the editor overwrites with zeros incur negligible storage under generic coding but carry full-amplitude residuals under both correct- and mismatched-seed replay. These terms explain the negative sign and approximate scale of the measured -3.63 % and -8.56 %.*

A.7 DEVIATION ENERGY AND SAVING

For an interior row, let $r = z - q$, with δ as in Proposition 2, and define $\rho_r = \|z - q\|^2/\|z\|^2$, $\rho_\delta = \|\delta\|^2/\|z\|^2$, and $R^2 = 1 - \|\delta\|^2/\|r\|^2$. Here Γ is the selected-representation saving versus the per-source best generic codec, as reported in Table 3. We take per-source medians over interior rows, then configuration medians; Spearman correlations use the 140 source medians within each configuration. Across the five configurations, $\rho_S(\rho_\delta, \Gamma)$ ranges from -0.98 to -0.63. Table 3 separates these configurations from the FlowEdit evaluation.

Table 3: Deviation energy and saving. Energy fractions, R^2, and savings are medians; ρ_S correlates ρ_δ with saving across sources, with approximate p-values. Panel A reports payload savings; Table 16 reports complete-file savings. Columns are aggregated independently, so their medians need not satisfy $\rho_\delta = \rho_r(1 - R^2)$. Panel B reports carrier-defect saving Γ_{carrier} against direct SZ3 response coding on the first target prompt, without pooling with Panel A.

Panel A: five factorized configurations

Configuration	n	ρ_r	ρ_δ	R^2	Γ (%)	ρ_S	p
EF/SD 1.5	140	0.00234	0.000351	0.827	46.81	-0.9005	9×10^{-52}
EF/SDXL	140	0.00228	0.000387	0.801	47.22	-0.9181	2×10^{-57}
PixArt-α	140	0.00639	0.000596	0.875	45.29	-0.6312	6×10^{-17}
LEDITS++/SD 1.5	140	0.01938	0.000899	0.935	36.55	-0.8463	1×10^{-39}
LEDITS++/SDXL	140	0.01795	0.000808	0.942	35.84	-0.9769	3×10^{-94}

Panel B: separate FlowEdit evaluation

Object	n	ρ_r	ρ_δ	R^2	Γ_{carrier} (%)	ρ_S	p
FlowEdit	15	0.087	0.083	0.019	21.8	-0.968	4×10^{-9}

Findings. In Panel A, the configuration medians of ρ_r are 0.23–1.94 % and those of ρ_δ are 0.035–0.090 %. These are medians of row-wise ratios rather than ratios of energies summed over the state, and because q and r need not be orthogonal, $1 - \rho_r$ measures the reduction in the energy to be coded rather than the share of $\|z\|^2$ explained by q. Across sources within a configuration, coefficients of variation in bytes are 0.04–0.29 % for the best generic codec and 2.6–4.7 % for the selected representation. Selected byte counts track ρ_δ (Spearman $+0.65$ to $+0.98$), so the ranking of savings within a configuration follows defect energy. The positive $\rho_S(\rho_r, \Gamma)$ range ($+0.20$ to $+0.92$) is consistent with larger residuals carrying more carrier-explainable structure.

Row aggregation. Boundary rows are excluded from the source medians. Each source median satisfies $\rho_\delta \le \rho_r < 1$; a single row adjacent to the boundary can have $\rho_r > 1$. Excluding these rows changes the within-configuration correlation by at most 0.003. The editor overwrites the last 3 LEDITS++/SDXL rows with zero, including 2 interior zero-energy rows. These rows are included in the computation of R^2 but excluded from the energy ratios; on them the carrier fits q_t. Fits use uncentered least squares with one EF carrier or up to two LEDITS++ carriers.

FlowEdit. Subtracting the pathwise target $\xi_\sigma - x_{\text{src}}$ produces most of FlowEdit's reduction in the energy to be coded; the carrier projection removes a further fraction $R^2 = 0.019$ of the remaining energy. The correlation with source-state residual saving relative to direct response coding is $\rho_S(\rho_r, \Gamma_{\text{residual}}) = -0.911$; Appendix H.4 reports that saving and the additional gain from carrier projection.

B PROTOCOL AND SELECTION

B.1 EDITORS, COHORTS, AND CODEC PARAMETERS

The primary cohort contains 14 sources per edit type (10 types). EF uses $T = 100$, skips 36 steps, and uses source/target classifier-free guidance (CFG) $3.5/15$; LEDITS++ uses $T = 50$ and skip fraction 0.15. SDXL uses 1024^2 resolution, with three zero-noise LEDITS++ steps.

LEDITS++ inversion uses $\eta = 1$ and the source caption with guidance 3.5. Editing uses the complete benchmark target caption as one positive semantic concept, guidance 5, no warm-up, and a magnitude mask at quantile 0.9 of the channel-summed absolute guidance; cross-attention and intersection masks are disabled. The SD 1.5 input retains the dataset's square brackets, while SDXL removes them and normalizes whitespace.

Our PixArt-α adaptation of EF uses $T = 100$, skip 36, CFG $3.5/15$, tolerance 10^{-2}, and 5 development sources. It uses the diffusers 4-channel CFG convention and `leading` timestep spacing, and the evaluation covers a single checkpoint under transformers 4.44.2. Appendix H.2 compares the native guidance convention and the default timestep grid. Four-step settings are given in Appendix F.2.

The qualitative galleries (Appendix I) use 40 shared EF and LEDITS++ sources and 20 sources per SDXL configuration, disjoint from the 5 development sources, with the same editor settings, primary nominal tolerances, and codec menus as the 140-source comparison.

The primary codec menus are fixed by editor and backbone before cohort evaluation; storage savings are measured relative to these menus. For EF/SD 1.5 and PixArt, the generic menu is $\{\mathrm{SZ3}, \mathrm{dbz}, \mathrm{ZFP}\}$ and the factorized-inner menu is $\{\mathrm{SZ3}, \mathrm{dbz}\}$. Both menus use $\{\mathrm{SZ3}, \mathrm{dbz}\}$ for EF/SDXL, $\{\mathrm{SZ3}, \mathrm{dbz}, \mathrm{ZFP}\}$ for LEDITS++/SD 1.5, and $\{\mathrm{SZ3}, \mathrm{dbz}, \mathrm{ZFP}, \mathrm{pcodec}\}$ for LEDITS++/SDXL.

Table 4: Generic-codec parameters. pcodec is part of the primary codec suite only for SDXL LEDITS++; Appendix C.3 reports its sensitivity elsewhere.

Codec	Fixed parameters
SZ3	absolute error bound ε; pysz default algorithm
ZFP	fixed-accuracy tolerance ε; zfpy
dbz	uniform step 2ε; adjacent-step delta on the step axis; Blosc2 byte shuffle; zstd level 9
pcodec	$q = \mathrm{round}(x/(2\varepsilon))$ as int32; pcodec 1.0.3 default chunk configuration

B.2 Selection rule and admission

For each source, we evaluate all eligible inner codecs under the comparison protocol in Section 5.1. We fix eligible modes for each editor family before evaluation, then select per source using decoding success, codec reconstruction error, decoded-state admission, and payload size. Missing required edit outputs from either the selected representation or its comparator count as operational failures in the full cohort; sources are not replaced.

B accounts for coefficients, fp16 boundaries, native endpoints, the inner bitstream, mode identifiers, shape/dtype information, and required per-source metadata. Candidates are coded at a common nominal tolerance ε. Admission of a factorized candidate requires achieved maximum-absolute fp32 codec reconstruction error within 0.8–$1.25\times$ the achieved error of the per-source best generic codec. The selector keeps the smallest payload among eligible candidates, with a generic fallback otherwise. The smallest decodable payload satisfies the admission criteria in all 5×140 primary source–configuration cases.

SZ3, dbz, and pcodec saturate nominal tolerance (achieved/nominal ratio 1.0000–1.0033); the largest overshoots from floating-point rounding were 0.02 % at $\varepsilon = 10^{-2}$, 0.10 % at 3×10^{-3}, and 0.33 % at 10^{-3}. In the primary nominal-tolerance comparison, ZFP attained 0.11–0.21 of nominal tolerance and never yielded the minimal byte count. In fixed-accuracy mode, ZFP quantizes to a bit plane at or below the requested tolerance, so its achieved error is below nominal. The per-source best generic codec was always SZ3, dbz, or pcodec, each saturating nominal tolerance. Appendix D.2 reports achieved-error comparisons using an expanded ZFP menu.

B.3 Statistical procedures and reconstruction checks

Primary-cohort confidence bounds use 9,999 edit-type-balanced source resamples with seed 0. Each configuration is analyzed separately, including configurations sharing sources. We report one-sided 95 % lower bounds for median savings and paired stencil differences, and upper bounds for selected-minus-generic $\Delta L_{\mathrm{mimic}}$. A catastrophic edit has selected-state $L_{\mathrm{mimic}} > 0.5$; a missing required edit output counts as an operational failure. All 140 sources complete in every configuration, with zero events of either kind; the exact one-sided 95 % upper count bound is 2.12 %.

Uncompressed float32 factorization round trips have maximum error 1.9×10^{-6} for EF/PixArt and 3.8×10^{-6} for LEDITS++, with all 700 source–configuration cases below 10^{-5}. Across five configurations, 1,960 containers reproduce their reference decoded controls bitwise in 3,080 standalone loads. Table 16 reports primary-cohort storage and output fidelity; all metrics use the representations selected by the state-error rule.

C Storage Evidence

C.1 Full PIE-Bench coverage

We evaluate the fixed EF/SD 1.5, EF/SDXL, LEDITS++/SD 1.5, LEDITS++/SDXL, and EF/PixArt-α operating points on all 700 PIE-Bench editing tasks: the 140-task primary cohort and its full 560-task complement, without filtering or replacement. The codec tolerance is 10^{-2} for EF and 3×10^{-3} for both LEDITS++ configurations. Table 5 reports complete-file savings against the generic candidate selected by accounted representation size under these fixed rules. Selected files are smaller across all $3{,}500$ configuration–task comparisons, with a minimum saving of $33.24\,\%$. Medians use the individual tasks in each cohort.

Exact source-file hashing identifies 692 byte-distinct images in the full benchmark and 555 in the complement; three file contents also occur in the primary cohort. We separately report 295 complementary tasks containing 292 byte-distinct images. This subset excludes tasks with recorded prior exposure, unresolved exposure status, or development use, and images whose bytes match a primary, exposed, or development source. The full-benchmark aggregate includes calibration and development data and is descriptive. In an exploratory weighting check, taking the median task saving within each image hash and then the median across the 692 hashes changes each configuration's median by at most 0.011 percentage points. Confidence bounds use edit-type-stratified task bootstrap resampling ($B = 9{,}999$); each task is a sampling unit, including tasks that share image contents.

Table 5: Median complete-file savings (%) on PIE-Bench. The exposure-qualified subset lies within the complement. The final column gives the one-sided 95% upper bound for the median paired LPIPS difference (selected minus generic) over all 700 tasks; it is a task-bootstrap summary.

Configuration	Primary 140	Complement 560	Qualified 295	Full 700	LPIPS upper bound
EF/SD 1.5	46.73	46.68	46.68	46.69	4.0×10^{-6}
EF/SDXL	47.20	47.03	47.04	47.04	2.85×10^{-5}
LEDITS++/SD 1.5	36.47	36.34	36.44	36.37	2.0×10^{-6}
LEDITS++/SDXL	35.82	35.75	35.71	35.77	4.85×10^{-5}
EF/PixArt-α	45.20	45.29	45.34	45.25	9.54×10^{-4}

Table 6 reports absolute LPIPS tails at these operating points. Their ordering depends on the configuration and quantile.

C.2 Tolerances and payload composition

Figure 2 compares per-source payload minima across all five configurations and three tolerances; Table 7 gives the median paired savings, and Appendix E.5 reports PixArt-α fidelity across tolerances. Payload components include the encoded controls and the native endpoint (65.536 kB for EF/SD 1.5).

Table 6: Absolute LPIPS tails on 700 PIE-Bench tasks at the fixed operating points in Table 5. Each entry is selected/generic. P95 and P99 use nearest-rank quantiles; counts use strict exceedance. These task-weighted summaries cover 692 distinct source-file hashes and use each configuration's original evaluation resolution.

Configuration	P95	P99	Maximum
EF/SD 1.5	0.007950/0.008690	0.021592/0.029907	0.158664/0.244126
EF/SDXL	0.007680/0.007335	0.023254/0.018232	0.049187/0.046732
LEDITS++/SD 1.5	0.005473/0.005806	0.016644/0.015140	0.042695/0.043718
LEDITS++/SDXL	0.008243/0.008158	0.017387/0.014625	0.065333/0.039850
EF/PixArt-α	0.169121/0.174818	0.258261/0.246525	0.420699/0.323797
Configuration	> 0.01	> 0.05	> 0.1
EF/SD 1.5	29/31	3/3	1/1
EF/SDXL	28/26	0/0	0/0
LEDITS++/SD 1.5	16/16	0/0	0/0
LEDITS++/SDXL	25/18	1/0	0/0
EF/PixArt-α	571/567	289/287	124/130
PixArt median (selected/generic): $0.036820/0.037249$			

Native controls and endpoints are fp16 for LEDITS++ and fp32 for EF and PixArt-α. The fp16 boundary rows and native endpoints impose a fixed overhead. As the tolerance increases, the defect payload decreases substantially while coefficients and metadata remain minimal.

Table 7: Tolerance ladder: medians on 140 sources per configuration.

Configuration	10^{-2}	3×10^{-3}	10^{-3}
EF / SD 1.5	46.81 %	41.93 %	40.83 %
EF / SDXL	47.22 %	41.31 %	38.18 %
LEDITS++ / SD 1.5	41.47 %	36.55 %	35.81 %
LEDITS++ / SDXL	42.74 %	35.84 %	31.07 %

C.3 Comparator robustness

Table 8 summarizes complete-file savings under comparators that differ from the primary rule of Section 4.3.

Four codecs for both representations. SZ3, ZFP, dbz, and pcodec are available to both representations on all 140 sources per configuration, under the primary admission rule. Every source in each of the five configurations yields a smaller file, the per-source minimum savings are 33.76–47.35 %, and all 1,400 selected and generic files reproduce their recorded decoded controls and endpoints in standalone decoding. On the 40-source LEDITS++/SD 1.5 control cohort, adding pcodec to the generic menu changes median savings from 36.37 to 30.54 %; adding it to both menus gives 37.79 %.

Refined tolerance grid and joint state errors. On the 20-source development subset, both representations additionally use SZ3, dbz, and pcodec at six fractions of the native cap (0.625, 0.75, 0.875, $1-2^{-8}$, $1-2^{-12}$, and $1-2^{-16}$), giving 110 candidates per source together with the menu of Appendix D.2. Admission requires the native maximum error to stay within the cap (0.01 for EF and PixArt-α, 0.003 for LEDITS++), and no edit output enters selection. The joint-error column admits a generic candidate only when both its achieved native maximum error and its MSE are no larger than those of the selected solver file. Solver files are smaller for all 100 source–configuration pairs, and all 200 selected files pass standalone native-tensor and endpoint hash checks. The ZFP column restricts both representations to ZFP, whose fixed-accuracy mode quantizes to a bit plane below the requested tolerance.

Native-input bound. For the fp16 LEDITS++ configurations, every candidate is decoded from its complete file, cast to fp16, and admitted when its maximum absolute error relative to the original native state is at most 3×10^{-3}. The codec-tolerance menu is $\{0.003, 0.00225, 0.0015, 0.00075\}$, and generic candidates also include lossless coding of the original fp16 bytes. All 3,640 candidates for SD 1.5 and 4,760 candidates for SDXL decode successfully. Lossless byte-shuffle and Zstandard coding of the native fp16 bytes achieves the lowest storage among generic candidates for every SD 1.5 source, with zero state error; on SDXL, both best-performing representation families use pcodec with inner tolerance 0.0015. The selected files attain a native maximum error of 0.001953125. On a fixed 20-source cohort selected by category-stratified hash ordering and disjoint in source identifiers and image hashes from the primary cohort and from the exact-state PixArt-α cohort, the same bound yields median complete-file savings of $34.15\,\%$ (SD 1.5) and $32.71\,\%$ (SDXL), with positive savings on every source. Appendix E.4 reports the corresponding output fidelity.

Table 8: Complete-file savings (%); medians over sources. Primary repeats Table 16; Four codecs expands its menu under the same admission rule. Panel B uses the container of Appendix D.2, and Joint errors constrains maximum error and MSE. Native bound applies to fp16 LEDITS++.

A. 140-source comparisons

Configuration	Primary	Four codecs	Native bound
EF/SD 1.5	46.73	48.73	–
EF/SDXL	47.20	48.99	–
LEDITS++/SD 1.5	36.47	38.07	33.81
LEDITS++/SDXL	35.82	35.82	32.68
EF/PixArt-α	45.20	46.79	–

B. 20-source development comparisons

Configuration	Common cap	Refined grid	Joint errors	ZFP only
EF/SD 1.5	42.92	48.07	48.09	34.17
EF/SDXL	42.02	48.05	48.07	33.33
LEDITS++/SD 1.5	35.31	36.20	36.30	27.52
LEDITS++/SDXL	36.85	37.76	37.76	28.09
EF/PixArt-α	42.00	46.15	46.15	33.77

D Mechanism Evidence

D.1 Analytical replay coefficients

Analytical and fitted coefficients. The noise-target substitution in the solver supplies schedule-derived weights c_t^{an} for the prescribed carrier. Fixing these weights gives the predictor $q_t + \Xi_t c_t^{\mathrm{an}}$ with no stored fitted coefficients. Its residual decomposes as

$$\delta_t^{\mathrm{an}} = \delta_t + \Xi_t(c_t - c_t^{\mathrm{an}}), \qquad \|\delta_t^{\mathrm{an}}\|^2 = \|\delta_t\|^2 + \|\Xi_t(c_t - c_t^{\mathrm{an}})\|^2,$$

where the equality holds in real arithmetic because $\Xi_t^\top \delta_t = 0$. This isolates the information supplied by the schedule from the correction supplied by fitting. The skeleton-only predictor q_t, with no carrier coefficients, provides a second control.

We compare skeleton, analytical, and fitted predictors on the common 140 sources per configuration, using the codec menu and native maximum-error caps of Appendix D.2. All predictors share the boundary and signed-zero handling, endpoint, and comparison container; generic coding includes two native-lossless candidates. Selection minimizes complete-file size using decoded state errors. Every selected generic and replay file passes standalone reconstruction checks.

Table 9 shows that analytical coefficients retain 33.76–$41.55\,\%$ median savings, compared with 35.70–$43.17\,\%$ for fitted coefficients. Both analytical and fitted representations yield smaller files than generic coding across every source in all five configurations.

For exact PixArt storage on the same source IDs, both predictors use whole-array XOR/pcodec on the native states of Appendix F.1. Analytical coding saves $13.51\,\%$ in median complete-file size

Table 9: Median paired complete-file savings (%) on 140 sources per configuration, relative to the per-source best generic candidate under a common native maximum-error cap. Skeleton and analytical predictors store no fitted coefficients.

Configuration	Skeleton	Analytical	Fitted
EF/SD 1.5	32.52	41.48	43.17
EF/SDXL	31.47	41.55	42.75
LEDITS++/SD 1.5	20.21	33.76	35.70
LEDITS++/SDXL	21.05	36.38	37.63
EF/PixArt-α	27.62	40.22	42.24

against the per-source best of six native-lossless configurations, versus $14.25\,\%$ for fitted coding. Analytical savings range from 12.12 to $22.47\,\%$. All 140 files in each predictor arm are smaller than their generic references, and standalone decoding restores native inputs bitwise.

Output fidelity on the common cohort. Table 10 evaluates each source's benchmark target against its raw-state edit using 512^2 PNGs, with SDXL downsampled from 1024 pixels. All 3,500 outputs complete. Analytical-replay median LPIPS is within 0.0007 of fitted replay in each configuration.

Table 10: LPIPS to raw-state edits on the common 140 sources, benchmark target prompt, 512^2 PNGs. P95 is the nearest-rank quantile; the last column counts sources above 0.01.

Configuration	Predictor	Median	Mean	P95	Maximum	> 0.01
EF/SD 1.5	Generic	0.000258	0.001147	0.003652	0.027261	5
EF/SD 1.5	Skeleton	0.000118	0.000746	0.003571	0.030723	2
EF/SD 1.5	Analytical	0.000109	0.000700	0.003438	0.013426	2
EF/SD 1.5	Fitted	0.000107	0.000912	0.002099	0.028121	4
EF/SDXL	Generic	0.000906	0.001694	0.004885	0.036786	2
EF/SDXL	Skeleton	0.000613	0.001685	0.004364	0.039298	3
EF/SDXL	Analytical	0.000624	0.001004	0.002275	0.010572	1
EF/SDXL	Fitted	0.000622	0.001351	0.004087	0.023188	4
LEDITS++/SD 1.5	Generic	0.000490	0.001053	0.002481	0.020761	2
LEDITS++/SD 1.5	Skeleton	0.000414	0.001118	0.002622	0.032818	3
LEDITS++/SD 1.5	Analytical	0.000465	0.001136	0.003586	0.012530	2
LEDITS++/SD 1.5	Fitted	0.000482	0.001463	0.003659	0.034426	5
LEDITS++/SDXL	Generic	0.001129	0.001796	0.005897	0.014213	3
LEDITS++/SDXL	Skeleton	0.001051	0.002028	0.004351	0.041540	3
LEDITS++/SDXL	Analytical	0.001038	0.002076	0.004894	0.052363	3
LEDITS++/SDXL	Fitted	0.001065	0.002470	0.007365	0.059369	3
EF/PixArt-α	Generic	0.048940	0.064443	0.195994	0.336700	118
EF/PixArt-α	Skeleton	0.044116	0.058072	0.166596	0.423912	111
EF/PixArt-α	Analytical	0.038887	0.059029	0.179319	0.443300	110
EF/PixArt-α	Fitted	0.039576	0.056443	0.169183	0.248587	111

D.2 Local-seed fits and achieved state errors

Protocol. We compare the prescribed solver representation with ordinary least-squares fits over the union of original draws used by q_t and Ξ_t (local OLS), and over that union expanded by one available draw at each end (wider OLS). Each configuration uses the same 20-source development subset of the primary cohort, with two sources per edit type. The five configurations yield 100 evaluations of 20 unique sources. All representations share the native boundary, endpoint, signed-zero policy, and a common comparison container that accounts for coefficient and metadata overhead. This format differs from the primary containers, so their file sizes are reported separately.

For each source, we decode each candidate, convert it to the editor's native dtype, and select the smallest admissible file in the fixed menu. The maximum-absolute native error cap is 0.01 for EF

and PixArt and 0.003 for LEDITS++. SZ3, dbz, and pcodec use the cap, half the cap, and one quarter of the cap as nominal tolerances; ZFP uses the cap times 2^j, $j = -3, \ldots, 5$. Generic storage also includes native-dtype lossless candidates. We retain the best generic, solver, local-OLS, and wider-OLS files and the best generic-ZFP and solver-ZFP files. Selection uses decoded state errors and file size. Achieved maximum error, MSE, and output fidelity can differ under the shared cap.

Storage and regression flexibility. Table 11 reports median savings and paired byte differences. The prescribed representation saves 35.31–$42.92\,\%$ relative to generic storage and has a positive median paired byte advantage over both fitted alternatives in every configuration. Both regressions use the original generating seed. Local OLS also shares the solver-local support and relaxes its prescribed coefficient constraint. Its near-matching rates complement the conditional residual-energy analysis in Proposition 5; the 177–257 B paired median gaps closely track the 188–252 B of added coefficients. Coded-residual differences are smaller than their per-source variation, consistent with Proposition 5's sub-bit-per-row expected gain. Wider OLS yields smaller mean file sizes for EF/SD 1.5, and both fitted alternatives achieve smaller mean file sizes for PixArt. Appendix D.5 compares exact storage against seed-blind temporal prediction under a common coding format.

Table 11: Complete-file comparisons on 20 shared sources per configuration. Savings are medians of per-source fractions relative to generic storage. Byte differences subtract the solver file size from that of the fitted alternative; positive values indicate a storage advantage for the solver. The last column counts smaller local/wider files, each out of 20.

Configuration	Solver saving	Local: median/mean (B)	Wider: median/mean (B)	Smaller: local/wider
EF/SD 1.5	$42.92\,\%$	256.5/2205.0	721.5/-8103.8	3/4
EF/SDXL	$42.02\,\%$	242.0/238.8	727.0/740.8	0/0
PixArt	$42.00\,\%$	247.0/-1568.0	742.5/-8018.9	5/6
LEDITS++/SD 1.5	$35.31\,\%$	194.5/200.1	577.0/575.4	0/0
LEDITS++/SDXL	$36.85\,\%$	177.0/184.4	524.0/541.1	0/0

Achieved state errors and editing. Under the common native cap, selected solver files attain median maximum error 0.005 for EF/PixArt versus 0.01 for generic storage, with approximately four times lower MSE; both attain 0.00195 for LEDITS++. Table 12 reports every retained representation's output fidelity against the same-runtime raw-state edit. P95 is the nineteenth ordered value among the 20 sources.

D.3 Draw support and generating information

The mechanism study uses all 140 sources per configuration from the primary cohort. We analyze their native materialized controls under the recorded noise-generation specification, without denoiser evaluation. Boundary rows and post-inversion zero rows remain outside the fit: each source contributes 63 active rows for EF/PixArt, 49 for LEDITS++/SD 1.5, and 47 for LEDITS++/SDXL.

Support discovery. For each active row, we fit every support with the width of the union of skeleton and carrier draws from the full original-draw dictionary: all $\binom{100}{2} = 4,950$ pairs for EF/PixArt, and all $\binom{50}{3} = 19,600$ triples for second-order LEDITS++ rows, with pairs for its first-order row. The objective is squared residual energy. Float64 Gram calculations shortlist candidates; candidates within a fixed numerical uncertainty band and the prescribed support are reevaluated by direct least squares. We record ties and the margin to the next support. All 39,808 recovered supports are unique under the recorded tolerance (Table 13). In the 92 LEDITS++ exceptions, the optimal draw triplet retains two prescribed draws and replaces ξ_{j+1}. Its maximum energy improvement is 4.35×10^{-7} and 3.61×10^{-8} of the raw row energy for SD 1.5 and SDXL, respectively. The search identifies a residual-energy optimum within the fixed-width original-draw family.

Table 12: Descriptive LPIPS on 512^2 PNG outputs against the raw-state edit rendered with the same runtime and original prompt, 20 sources per configuration. All retained representations are included; P95 is nearest rank.

Configuration	Statistic	Generic	Solver	Local OLS	Wider OLS	Generic ZFP	Solver ZFP
EF/SD 1.5	Median	0.000243	0.000110	0.000107	0.000121	0.000028	0.000037
	Mean	0.000664	0.001705	0.001575	0.001548	0.000038	0.000051
	P95	0.002505	0.001671	0.000495	0.000524	0.000103	0.000099
	Maximum	0.002584	0.028121	0.028232	0.027792	0.000156	0.000214
EF/SDXL	Median	0.000960	0.000649	0.000607	0.000530	0.000537	0.000554
	Mean	0.003028	0.001652	0.001700	0.002149	0.002701	0.002699
	P95	0.003926	0.001890	0.002968	0.004978	0.002208	0.001665
	Maximum	0.036786	0.017318	0.017111	0.025116	0.040040	0.040268
PixArt	Median	0.035346	0.036245	0.033884	0.042384	0.025384	0.020656
	Mean	0.062533	0.051679	0.054138	0.054052	0.040901	0.053358
	P95	0.195994	0.131568	0.213746	0.171378	0.111698	0.219781
	Maximum	0.244411	0.169183	0.218887	0.189491	0.132003	0.240316
LEDITS++/SD 1.5	Median	0.000417	0.000474	0.000533	0.000430	0.000443	0.000387
	Mean	0.000638	0.001008	0.000695	0.001225	0.000645	0.001164
	P95	0.001867	0.003026	0.001668	0.004075	0.001623	0.002454
	Maximum	0.002372	0.003659	0.001823	0.009845	0.001820	0.011818
LEDITS++/SDXL	Median	0.001129	0.001083	0.001187	0.001060	0.001043	0.000955
	Mean	0.001325	0.001327	0.001389	0.001263	0.001396	0.001255
	P95	0.002554	0.002562	0.002617	0.002457	0.003449	0.002663
	Maximum	0.002975	0.002571	0.003362	0.002591	0.004613	0.003028

Additional directions. We refit the prescribed carrier in float64 to obtain $\delta = (I - P_\Xi)(z - q)$. For extra draws W, let $\widetilde{W} = (I - P_\Xi)W$, $k = \operatorname{rank}(\Xi)$, and $m = \operatorname{rank}(\widetilde{W})$. We measure

$$F = \frac{\|P_{\widetilde{W}}\delta\|^2}{\|\delta\|^2}.$$

Conditional on fixed (Ξ, δ), independent Gaussian extra directions give $F \sim \mathrm{Beta}(m/2, (D-k-m)/2)$ and $\mathbb{E}[F] = m/(D-k)$. For each source we average $F/\mathbb{E}[F]$ over rows, then average sources equally. Table 13 compares the extra local original draw with an independent Gaussian direction. The latter agrees with the reference scale; original draws exhibit excess explanatory energy in LEDITS++. Expanding the local support by one available draw at each end gives the same qualitative distinction.

Table 13: Mechanism diagnostics, 140 sources per configuration. Support counts include every active row. Direction columns average the additional projection fraction, normalized by $m/(D-k)$, first over rows within each source and then equally over sources.

Configuration	Support recovered	Original	Independent
EF/SD 1.5	8,820/8,820	1.019	0.995
EF/SDXL	8,820/8,820	1.000	1.009
LEDITS++/SD 1.5	6,776/6,860	1.800	1.002
LEDITS++/SDXL	6,572/6,580	3.185	0.989
PixArt	8,820/8,820	1.013	1.008

Information access. Dense OLS over all original draws leaves a median per-source active-row residual-energy fraction of 0.15–1.48%. The same-width dictionary regenerated with seed $+1000$ leaves 99.39–99.92%. These are linear-prediction diagnostics on native controls; the exact PixArt files in Table 15 measure the corresponding coding comparison.

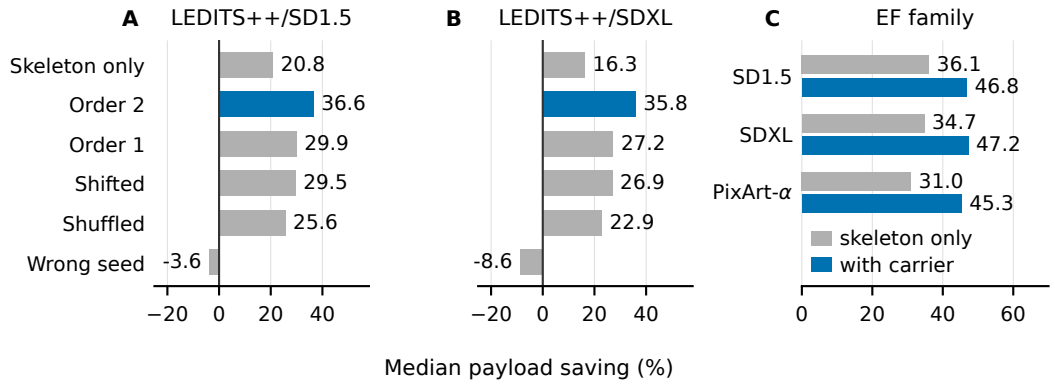

Figure 5: Median payload savings on 140 sources per configuration, counting all stored components and metadata. A/B: six LEDITS++ replay variants on SD 1.5/SDXL at 3×10^{-3}, with a shared horizontal scale; blue denotes the prescribed order-two carrier. C: skeleton-only and carrier coding for EF/SD 1.5, EF/SDXL, and PixArt-α at 10^{-2}.

D.4 LEDITS++ CARRIER CONTROLS

On the 140-source cohort at 3×10^{-3} (Figure 5), LEDITS++ median savings are 20.75 % (no carrier), 29.90 % (order-one carrier), 36.55 % (prescribed order-two carrier), 29.54 % (shifted), 25.64 % (shuffled), and -3.63 % (mismatched seed) on SD 1.5, and 16.33 %, 27.21 %, 35.84 %, 26.92 %, 22.88 %, and -8.56 % in the same order on SDXL. Skeleton-only EF coding saves 36.08 % (EF/SD 1.5), 34.73 % (EF/SDXL), and 30.97 % (PixArt-α) against the same generic comparator.

The shifted-carrier and shuffled-step controls test alignment while retaining carrier width. Shifting offsets Ξ_t by one draw and preserves q_t; shuffling permutes across rows the solver constants used to build q_t and leaves Ξ_t unchanged. Both reduce byte savings. The mismatched-seed control regenerates q_t and Ξ_t from an unrelated seed (the provenance seed incremented by 1000) and increases file sizes (Remark 6). All variants include the same stored components in their byte counts, and Table 14 gives paired differences with lower bounds.

Table 14: Paired per-source differences in LEDITS++ byte-saving rates at tolerance 3×10^{-3}, with $n = 140$ per configuration (bytes-only). Medians and one-sided 95 % lower bounds are in percentage points. Bounds use the same source bootstrap as the main results, resampling within edit type (9,999 resamples).

Configuration	Paired difference	Median (pp)	LB (pp)
LEDITS++/SD 1.5	order 2 − order 1	6.63	6.47
LEDITS++/SD 1.5	order 1 − no carrier	9.08	9.01
LEDITS++/SD 1.5	order 2 − no carrier	15.69	15.47
LEDITS++/SD 1.5	order 2 − shifted	6.96	6.84
LEDITS++/SD 1.5	order 2 − shuffled	10.88	10.76
LEDITS++/SD 1.5	order 2 − mismatched seed	40.13	40.04
LEDITS++/SDXL	order 2 − order 1	8.66	8.48
LEDITS++/SDXL	order 1 − no carrier	10.86	10.80
LEDITS++/SDXL	order 2 − no carrier	19.54	19.23
LEDITS++/SDXL	order 2 − shifted	8.95	8.78
LEDITS++/SDXL	order 2 − shuffled	12.96	12.85
LEDITS++/SDXL	order 2 − mismatched seed	44.41	44.25

D.5 EXACT-STATE PREDICTOR CONTROLS

Seed-informed and temporal predictors. Two alternative predictors test the role of the original draws on the primary-cohort subset and disjoint cohort of Appendix F.1, each containing 20 sources. Original-seed local OLS uses the same two-draw union as Appendix D.2, without constraining its coefficients to the solver parameterization. Seed-blind temporal OLS predicts z_t from the two preceding native rows in stored order, $\gamma_t z_{t-1} + \eta_t z_{t-2}$, with fitted coefficients γ_t, η_t; unavailable lags and the boundary predictor are zero. Rank-aware least squares handles zero or dependent lag rows. Both fits use float64 arithmetic and serialize fp32 coefficients; encoder and decoder evaluate these stored coefficients in the same fp32 order. The temporal decoder uses only rows it has already restored bitwise, so it needs no generating seed or scheduler for prediction. The stored state also includes the complete native editor metadata.

All three predictors use the same whole-array XOR correction and pcodec defaults. Coefficients occupy 252 B for the solver and 504 B for each OLS alternative; complete-file counts include these costs, the endpoint, and metadata. The comparator is the minimum of the six native lossless arms. The two fixed cohorts therefore compare predictor choice at exact native-state reconstruction with a common residual codec.

Both seed-informed predictors produce smaller files than all six native coding configurations on every source in both cohorts (Table 15). The original draws therefore improve compression over this two-lag temporal predictor on both cohorts. Paired median local-OLS-minus-solver file differences decompose as $362 = 252 + 3 + 107$ B on the primary-cohort subset and $255 = 252 + 3 + 0$ B on the disjoint cohort: extra coefficients, metadata, and residual payload, respectively. Coefficient and metadata differences are constant across sources and endpoints are identical, so these decompositions preserve the paired medians. The solver uses fewer parameters at comparable measured payload lengths. The full-dictionary comparison below tests original-draw dictionaries with no solver-local support supplied. Standalone decoding restores the complete input identically for all three predictors and the best native representation on all 40 sources.

Full-dictionary and support-discovery controls. A fixed dictionary containing all 100 original generation draws tests their coding value without supplying solver-local support or schedule coefficients. This comparison uses the same two fixed 20-source cohorts. Dense OLS fits all 63 non-boundary rows in float64 (relative singular-value cutoff 10^{-12}, rank 100), stores 25.2 kB of coefficients, and predicts by fixed-order fp32 accumulation. An otherwise identical control uses seed $+1000$, fixed before evaluation. Both retain the native manifest seed, boundary, endpoint, full-array XOR correction, and default pcodec. Relative to the per-source best of six native configurations, the original-draw dictionary saves 13.55–13.61%. Unrelated-draw OLS enlarges every file, with a median increase of 5.73% (Table 15).

Exhaustive pair selection tests whether fitting can discover the support under a two-draw budget. It evaluates all $\binom{100}{2} = 4,950$ draw pairs per row using float64 OLS residual energy, with lexicographic tie-breaking, then refits the optimal draw pair. Support selection uses OLS residual energy alone; the stored representation includes 504 coefficient bytes and 126 index bytes. Comparing the fitted pairs with the analytic adjacent-draw support yields complete agreement across all $2,520/2,520$ rows. Its exact-file savings closely match the solver's: paired median excess costs are 179 and 73 B, and the fitted files are smaller in 15/40 sources. The fitted non-carrier weight of ξ_{t-1} matches the solver value b_{t-1}/σ_t to a median relative error of 1.9×10^{-4} (P95 1.7×10^{-3}) over these 2,520 rows of the 40 exact-state sources. On the 15/40 sources with smaller fitted files, the fitted representation has a container header at least 305 B shorter. Pair-search and local-OLS residual payloads are byte-identical on all 40 sources.

The solver identifies the support and coefficient constraint analytically. Exhaustive regression recovers that support and comparable exact-state file sizes on the same two cohorts, at the cost of a per-source support search.

Table 15: Exact complete-file savings versus each source's best of six native configurations. All predictors use full-array XOR and the same pcodec settings. Counts and byte costs include the complete stored representation; the pair search minimizes residual energy, not file size.

Predictor	Coefficients + indices (B)	Primary subset	Disjoint cohort
Temporal OLS	504	-2.4440%	-2.4370%
Solver	252	14.3094%	14.2255%
Local OLS	504	14.2992%	14.2124%
Full original-draw OLS	25,200	13.6105%	13.5509%
Full unrelated-draw OLS	25,200	-5.7257%	-5.7345%
Original-draw pair search	630	14.3044%	14.2176%

E Output Fidelity

E.1 Absolute editing fidelity on the 140-source cohort

Table 16 reports fidelity to each source's benchmark-target raw-state edit. Primary L_{mimic} uses floating-point outputs for EF and PixArt and native-resolution PNGs for LEDITS++ (1024 pixels for SDXL). The three-prompt evaluation uses 512-pixel PNGs, with SDXL downsampled from 1024 (Appendix E.6). The precision and exact-state analyses are collected in Appendices E.5 and F.1.

Table 16: Storage and fidelity to raw-state edits on the 140-source cohort, benchmark target prompt. File savings are medians with one-sided 95 % lower bounds under Section 4.3's rule; comparisons are paired within each configuration, and $\Delta L_{\mathrm{mimic}}$ UB is the one-sided 95 % upper bound on the median selected-minus-generic difference. L_{mimic} uses floating-point outputs for EF/PixArt and native-resolution PNGs for LEDITS++. All arms have 0 catastrophic edits ($L_{\mathrm{mimic}} > 0.5$). Median relative tolerances ε/RMS are 0.20 % (EF), 0.25 % (PixArt), and 0.08 % (LEDITS++).

Configuration	Arm	File saving [LB]	$\Delta L_{\mathrm{mimic}}$ UB	Median	P95	Max	Count > 0.1
EF/SD 1.5	selected	46.73 [46.51] %	3×10^{-6}	0.0002	0.0046	0.034	0
	best generic	–	–	0.0002	0.0049	0.031	0
EF/SDXL	selected	47.20 [46.87] %	7.9×10^{-5}	0.0015	0.0096	0.035	0
	best generic	–	–	0.0013	0.0060	0.016	0
LEDITS++/SD 1.5	selected	36.47 [36.36] %	0	0.00045	0.0035	0.015	0
	best generic	–	–	0.00049	0.0050	0.023	0
LEDITS++/SDXL	selected	35.82 [35.67] %	1.31×10^{-4}	0.0024	0.010	0.065	0
	best generic	–	–	0.0024	0.0091	0.040	0
PixArt-α	selected	45.20 [44.99] %	1.3×10^{-5}	0.038	0.179	0.421	27
	best generic	–	–	0.046	0.197	0.247	33

E.2 Common output-error budgets

For each of the 20 development sources and its benchmark target prompt, we compare four rendered candidates: generic and factorized files from the fixed codec suite, and their ZFP candidates. LPIPS is measured on the same 512^2 PNG outputs against the raw-state edit. The generic menu also includes its smallest native-lossless file. These native files reproduce the controls and endpoint bitwise; their zero distortion denotes reuse of the raw output under the recorded rendering specification. Files use the comparison container of Appendix D.2, which includes native boundary rows, the endpoint, coefficients where required, and metadata.

At each output-error budget, we choose the smallest admissible complete file from each menu. The factorized menu retains the generic option, so nonnegative savings follow by construction. Selection uses measured output LPIPS for each source–prompt pair within the evaluated candidate menu.

Table 17 reports the descriptive sweep: the median reduction at LPIPS ≤ 0.01 is 35.31–42.92 % across the four configurations. At 10^{-4}, EF/SDXL and both LEDITS++ configurations use generic storage for all 20 sources. At zero error, all four configurations select generic storage.

Table 17: Median per-source complete-file savings (%) under a shared output-LPIPS budget on 20 development sources. Parentheses give the number of sources that select generic storage. Selection uses measured output LPIPS over the evaluated candidates for each source's benchmark prompt.

LPIPS budget	EF/SD 1.5	EF/SDXL	LEDITS++/SD 1.5	LEDITS++/SDXL
0	0.00 (20)	0.00 (20)	0.00 (20)	0.00 (20)
10^{-4}	51.70 (1)	0.00 (20)	0.00 (20)	0.00 (20)
5×10^{-4}	42.95 (0)	0.00 (11)	20.85 (8)	0.00 (19)
10^{-3}	42.86 (0)	41.84 (6)	34.73 (4)	22.44 (8)
2.5×10^{-3}	42.95 (0)	42.02 (1)	35.26 (2)	36.73 (1)
5×10^{-3}	42.92 (0)	41.92 (1)	35.31 (0)	36.85 (0)
10^{-2}	42.92 (0)	41.92 (1)	35.31 (0)	36.85 (0)
2.5×10^{-2}	42.92 (0)	42.02 (0)	35.31 (0)	36.85 (0)
5×10^{-2}	42.92 (0)	42.02 (0)	35.31 (0)	36.85 (0)

E.3 EF AT A TIGHTER TOLERANCE

This comparison uses the primary 140-source cohort. For each source's benchmark target prompt, we compare factorized EF files at nominal codec tolerance 0.003 with generic files at 0.01 and 0.003. Within each representation and tolerance, the encoder selects the smallest payload before evaluating any edit output. Each file includes the endpoint and decoder metadata; all savings use actual complete-file sizes. The editor, inversion state, source seed, and prompt are shared across arms.

LPIPS uses float outputs at 512^2 for SD 1.5 and 1024^2 for SDXL, with the same raw-state reference as the primary comparison. For two raw-state checks per configuration, repeated renders are pixel-identical to each other and to the reference output under the recorded environment. The compressed inputs are decoded from the selected bitstreams without re-encoding. Table 18 reports all three arms; P95 and P99 use nearest-rank quantiles.

Table 18: Complete-file size medians and LPIPS to raw-state EF edits on 140 sources per backbone. Error columns summarize the full source distribution; the final column counts threshold exceedances. For each representation, payload selection precedes output evaluation.

Backbone	File / tolerance	Bytes (10^6)	Median	P95	P99	Maximum	> 0.01
SD 1.5	Generic / 0.01	1.4589	0.000221	0.004947	0.026779	0.030643	4/140
	Generic / 0.003	1.7329	0.000020	0.001039	0.001952	0.002816	0/140
	Factorized / 0.003	1.0075	0.000021	0.001652	0.004692	0.009161	0/140
SDXL	Generic / 0.01	5.7489	0.001328	0.005981	0.013658	0.016214	3/140
	Generic / 0.003	6.7095	0.000821	0.004472	0.013475	0.025403	3/140
	Factorized / 0.003	3.9395	0.000807	0.003550	0.015261	0.028812	2/140

Relative to generic storage at 0.01, every factorized file is smaller; median paired complete-file savings are $30.97\,\%$ and $31.48\,\%$ on SD 1.5 and SDXL, respectively. Median paired factorized-minus-generic LPIPS differences are -0.000178 and -0.000418, with one-sided 95 % upper bounds of -0.000148 and -0.000332 from 9,999 edit-type-stratified source-bootstrap resamples (seed 0). Both observed P95 values decrease. Individual source LPIPS increases in 6/140 and 19/140 cases, respectively, and SDXL has larger P99 and maximum errors.

E.4 LEDITS++ UNDER A NATIVE-INPUT BOUND

For the native-input bound of Appendix C.3, the original uncompressed states and the admitted states are evaluated in the same rendering environment. For each configuration, all 140 uncompressed-state

outputs are pixel-identical to the recorded reference images. Table 19 reports PNG-based LPIPS against these same-environment references. The lossless generic controls and endpoint on SD 1.5 are bitwise identical to the original state, so their reference images are reused; the SDXL generic states are rendered separately.

Table 19: Editing deviation under the native-input bound of Appendix C.3. Each configuration contains 140 sources and one benchmark target per source. P95 uses the nearest-rank definition; no output exceeds LPIPS 0.1.

Configuration	Representation	Median	P95	Maximum	> 0.01
LEDITS++ / SD 1.5	Generic	0	0	0	$0/140$
	Selected	0.000440	0.003727	0.017191	$3/140$
LEDITS++ / SDXL	Generic	0.001129	0.005897	0.014213	$3/140$
	Selected	0.001112	0.004752	0.041940	$4/140$

Disjoint evaluation cohort. On the disjoint 20-source cohort of Appendix C.3, selected-output 512-pixel PNG LPIPS on SD 1.5 relative to the stored raw output has median 0.000418, nearest-rank P95 0.006705, and maximum 0.006836. The generic state is lossless, so its reference image is reused after native-state identity verification. On SDXL, both admitted representations are rendered for all 20 benchmark targets. Selected/generic 512-pixel PNG LPIPS has medians $0.001436/0.001068$, nearest-rank P95 values $0.004189/0.003629$, and maxima $0.007442/0.004176$; no output exceeds 0.01. Mean LPIPS is $0.001957/0.001576$ for selected/generic storage.

E.5 PixArt-α precision and tolerance

We examine six diagnostic sources spanning the displayed PixArt-α examples, including the largest observed deviation. Let S be the original control array and e the error of its stored reconstruction. We hold S, e, text embeddings, the VAE, and the fp32 reverse loop fixed, and compare fp16 transformer execution with fp32 execution obtained by exactly upcasting the same fp16 weights. Each condition is compared with its own raw-state output.

Table 20: LPIPS to the corresponding raw-state output on six diagnostic sources, summarized as median / maximum. Scaled perturbations are prescribed state-error interventions; storage operating points are evaluated in the file-based experiments. The unquantized row reconstructs S from the factorization with an uncompressed defect; the other rows add the stored reconstruction error e, or e scaled by $1/4$ or $1/16$, to S.

Control array	fp16 transformer	fp32 transformer
Unquantized factorization	$0.00855/0.07608$	$1.42\times10^{-6}/3.89\times10^{-6}$
$S + e$	$0.08207/0.42070$	$0.07320/0.26130$
$S + e/4$	$0.04877/0.40399$	$0.02313/0.20809$
$S + e/16$	$0.07537/0.16254$	$0.01829/0.07227$

Raw-state replay is exact on all six sources in the original fp16 editor. Unquantized factorization introduces small reconstruction-rounding errors that are amplified during editing, and the fp16 responses vary nonmonotonically with perturbation magnitude. Under fp32 execution, the same rounding errors change the output by at most 3.89×10^{-6} LPIPS, while the stored reconstruction error e still gives median LPIPS 0.07320. These findings demonstrate the necessity of evaluating state reconstruction and edited-output fidelity independently.

Payload and replay fidelity across tolerances. We compare three nominal tolerances on the 140 PixArt-α sources in the primary cohort. For each source, we select the smallest generic payload among SZ3, dbz, and ZFP, and the smallest factorized payload among SZ3 and dbz. We fix the 3×10^{-3} and 10^{-3} choices before measuring output fidelity.

This exploratory analysis reuses the primary cohort's inversion states, seeds, and benchmark target prompts. The model, scheduler, precision, and guidance settings are fixed across tolerances. We verify exact raw replay against the stored reference on two fixed sources. Reported payload sizes include native endpoints.

Table 21: PixArt-α tolerance sensitivity on 140 paired sources. Generic (G) and factorized (F) candidates minimize payload bytes separately. P95 uses the nearest-rank convention. State errors are maximum absolute fp32 reconstruction errors before the native-dtype cast, with ratio F/G; the last column of panel B counts pairs outside $[0.8, 1.25]$. LPIPS compares each decoded-state replay with its raw reference; $\Delta L = L_F - L_G$ is paired by source.

A. Payload (decimal MB); medians

ε	Generic	Factorized	Paired saving (%)
10^{-2}	1.416	0.775	45.29
3×10^{-3}	1.676	1.007	39.93
10^{-3}	1.983	1.204	39.32

B. Achieved state error ($\times 10^{-3}$); median / P95

ε	Generic	Factorized	Error ratio	Outside band
10^{-2}	9.99999 / 9.99999	10.00023 / 10.00023	1.00002 / 1.00002	0
3×10^{-3}	2.99996 / 2.99999	3.00026 / 3.00026	1.00010 / 1.00011	0
10^{-3}	0.99999 / 0.99999	1.00040 / 1.00040	1.00042 / 1.00042	0

C. Replay LPIPS; median / P95 / maximum

ε	Generic	Factorized	> 0.1 (G/F)	ΔL (med. / P95)
10^{-2}	0.046 / 0.197 / 0.247	0.038 / 0.179 / 0.421	33 / 27	$-0.0009/0.0551$
3×10^{-3}	0.036 / 0.164 / 0.285	0.030 / 0.141 / 0.211	25 / 20	$-0.0026/0.0263$
10^{-3}	0.036 / 0.177 / 0.254	0.033 / 0.139 / 0.219	19 / 20	$-0.0006/0.0356$

Payload savings reach $39.32\,\%$ at 10^{-3}. All 420 selected pairs satisfy the primary achieved-error ratio band, with selection based on payload size alone.

Tightening ε from 10^{-2} to 10^{-3} reduces factorized LPIPS P95 from 0.179 to 0.139 and the count above 0.1 from 27 to 20. Output fidelity is not monotone in the nominal tolerance: from 3×10^{-3} to 10^{-3}, factorized P95 decreases from 0.141 to 0.139 while median LPIPS increases from 0.030 to 0.033. LPIPS increases for 80 of 140 sources, with paired median change $+0.0010$. At 10^{-3}, 19 generic and 20 factorized outputs exceed 0.1.

E.6 Repeated editing from one stored state

To test reuse across requests, each of five configurations applies the fixed generic and selected representations of 140 sources to each source's benchmark target and two fixed global prompts. Table 22 reports worst-prompt fidelity. The global prompts are "turn the entire scene into a watercolor painting" and "make the scene look like it is night time". Both are applied verbatim to every source and representation. Each request starts independently from the same stored state, using the same codec choice and requiring no state re-extraction. LPIPS is computed on the stored 512-pixel PNGs against the raw-state output for the same prompt; SDXL outputs are downsampled from 1024 pixels.

The four U-Net configurations have selected worst-prompt LPIPS medians below 0.002 across the three requests.

Table 22: Descriptive reuse fidelity across three prompts. Each entry reports generic/selected. LPIPS is first maximized across prompts within each source, then summarized across 140 sources; P95 is the nearest-rank empirical percentile.

Configuration	Median LPIPS	P95 LPIPS	Maximum LPIPS	Sources above 0.1
EF / SD 1.5	0.00089/0.00080	0.01351/0.01248	0.04568/0.03596	0/0
EF / SDXL	0.00117/0.00123	0.00774/0.00928	0.07944/0.03488	0/0
EF / PixArt-α	0.08189/0.07975	0.20377/0.19606	0.52446/0.42063	52/50
LEDITS++ / SD 1.5	0.00073/0.00077	0.00721/0.00673	0.02307/0.01506	0/0
LEDITS++ / SDXL	0.00193/0.00197	0.00610/0.00725	0.03581/0.05457	0/0

F EXACT-STATE FORMATS

F.1 EXACT-STATE FORMAT FOR PIXART-α

The solver predictor also supports exact state coding. Let z_t be the original native fp32 control row and $\tilde{z}_t = q_t + \Xi_t c_t$ the fp32 solver predictor, with a zero predictor for the boundary row, and let bits map an fp32 value to its 32-bit representation. Instead of coding a floating-point difference, we losslessly code

$$d_t = \mathrm{bits}(z_t) \oplus \mathrm{bits}(\tilde{z}_t), \qquad \widehat{z}_t = \mathrm{bits}^{-1}(d_t \oplus \mathrm{bits}(\tilde{z}_t)) .$$

Here $\oplus$ is bitwise exclusive-or. Stored coefficients and provenance reproduce the predictor; the file also carries the endpoint, metadata, and checksums. The XOR correction restores the control bits without floating-point residual subtraction or addition. Exact decoding requires identical predictor replay under the validated provenance; a native-state fingerprint rejects mismatched reconstruction. Decoding requires no denoiser evaluation.

Exact-input reuse. Let $E(S, p; U, \omega)$ be the unchanged editor on complete native input S, prompt p, runtime settings U, and execution randomness ω. If decoding restores S bitwise and execution is deterministic given the same U and ω, then $E(\mathrm{Dec}(b; U), p; U, \omega) = E(S, p; U, \omega)$ for each prompt. Bitwise XOR restoration supplies the input-identity premise; Appendix E.5 measures the reconstruction rounding introduced by ordinary floating-point residual subtraction and addition.

pcodec's default configuration for the uint32 residual is fixed using six format-selection sources. A zero-predictor control applies the same codec directly to the original uint32 bit patterns. The primary-cohort subset contains 20 sources from the primary cohort: the first two sorted IDs per edit category after excluding sources used for format selection or precision diagnostics. Six native lossless baselines include pcodec on both fp32 values and uint32 bit patterns, Zstd, lossless bit-shuffle/Zstd, and LZ4 variants. Every coefficient, boundary, endpoint, and metadata byte is included in the byte count.

All 20 solver-predicted bit-residual files and their 20 minimum-size native comparators reproduce the original native artifact bitwise in newly started processes, including controls, endpoint, manifest, scales, and source prompt. Native fp32 pcodec is the smallest generic representation for every source. The median per-source complete-file reduction is $14.31\,\%$, with range 13.71–$18.10\,\%$. On two diagnostic sources, including the largest-deviation case, lossless bit correction of the state followed by the original fp16 editor produces pixel-identical outputs.

The disjoint cohort contains 20 category-stratified sources, disjoint in identifiers and image hashes from the primary cohort, and gives a median exact-state complete-file saving of $14.23\,\%$ (range 13.48–$15.19\,\%$). All 20 solver-predicted bit-residual files and 20 minimum-size native files pass standalone checks of the complete decoded artifact, and every source has positive savings. Four standalone checks at sorted full-benchmark indices 0, 233, 466, and 699, one on each of four GPUs, restore the native input from a standalone exact-state file. Recorded file-access traces show reads of the encoded file, program libraries, and runtime resources, with no source-image, native-state-cache, or model-weight reads.

Table 23: Exact-state file sizes and implementation costs on the 20-source PixArt-α primary-cohort subset, excluded from format selection. Entries are medians. Encoding includes seed regeneration, coefficient fitting, compression, metadata, and writing; common raw-input loading is separate. Standalone decoding includes file reading, checks, and CUDA initialization where needed, but excludes writing the restored artifact.

Representation	File (MB)	Encode (ms)	Decode (ms)	Process (ms)
Native fp32 pcodec	3.5070	22.9	14.0	1275.5
Solver-predicted bit residual	3.0052	40.4	370.4	1850.8

Table 23 measures exact-state encoding and standalone decoding on a shared GPU; Section 5.6 and Appendix G.2 measure lossy and exact-state warm queries. Encoding uses a persistent process, and both standalone decoders use the same executable with PyTorch import. Process times include startup.

Full-benchmark exact-state storage. We evaluate seven representations on all 700 PIE-Bench states using the same encoding and validation procedure: the solver-predicted bit residual and six native-lossless configurations, including the uint32 pcodec zero-predictor control. The full benchmark contains 71 tasks used in the two cohorts, six-source format selection, or precision diagnostics. We report this group separately from the remaining tasks and treat the full aggregate as descriptive, following Appendix C.1. The remaining 629 tasks retain the full-benchmark median saving of $14.21\,\%$ (Table 24). Native fp32 pcodec is the smallest native configuration on all 700 tasks; both it and the solver-predicted bit-residual format restore the original native editor input (controls, endpoint, and manifest) bitwise in newly started processes on $700/700$ tasks. Taking the median within each image hash and then across hashes gives $14.21\,\%$ for both all 692 hashes and the remaining tasks' 622 hashes. The two 20-source cohorts give identical sizes for all 280 files across evaluations, with median savings of $14.31\,\%$ and $14.23\,\%$.

Table 24: Exact-state complete-file savings (%) against the per-source best codec among six native-lossless configurations. LB denotes the one-sided 95% lower bound from the edit-type-stratified task bootstrap of Appendix C.1 ($B = 9{,}999$).

Task group	n	Median	95% LB	Range	Positive
All	700	14.21	14.17	12.76–22.18	700/700
Predictor/format diagnostics	71	14.22	14.06	13.48–18.10	71/71
Remaining	629	14.21	14.17	12.76–22.18	629/629
Primary cohort	140	14.25	14.16	12.76–20.64	140/140
Complement	560	14.20	14.17	13.24–22.18	560/560

F.2 Exact four-step state coding

We evaluate ReNoise and TurboEdit on SDXL-Turbo at 512×512 using the common 140 PIE-Bench sources, with the source IDs, seeds, and three prompts of Appendix E.6. ReNoise uses four Euler inversion/editing steps, nine re-noising iterations (five on the first step), noise correction, and guidance scale zero. Regularization is evaluated in float32; stored noise and endpoint tensors retain float16. TurboEdit uses four steps, source means in float32, its shifted path from Appendix A.3, guidance weight 1.5, and final residual norm clipping at 15.5. Native editing uses a batch of three latent samples. The software environment is PyTorch 2.4.1/CUDA 12.4 with diffusers 0.24.0 for ReNoise and 0.32.2 for TurboEdit; the code supplement lists the pinned ReNoise and TurboEdit reference-code revisions.

Exact coding and accounting. For each source, generic coding selects the smallest complete file among six native-lossless configurations: pcodec on floating-point or unsigned-bit arrays, zstd, lz4, and two bitshuffle/zstd settings. Predictor coding uses the same codec menu on a bitwise XOR

or on a float32 remainder with an additional native-bit XOR repair. The file includes predictors' coefficients when fitted, source latent or endpoint, seeded-path repair, metadata, and integrity checks. TurboEdit's shared-path representation regenerates the noising path from the source latent and seed; the direct-array reference losslessly stores all required native arrays. Its analytical mean predictor is compared with a generic representation that already has the same shared-path replay.

Storage and output preservation. ReNoise analytical/fitted coding saves $10.56/10.85\,\%$ in median complete-file size; analytical savings range from 9.83 to $11.57\,\%$. TurboEdit's shared path saves $37.43\,\%$ against direct native-array coding. Analytical/fitted mean prediction then saves $1.78/2.09\,\%$ against generic coding with that path. Analytical savings against the shared-path reference range from 0.96 to $4.18\,\%$; against direct native arrays, the combined median saving is $38.57\,\%$. Each comparison yields smaller files on all 140 sources.

Encoding and decoding with every selected predictor preserves native state bits. For each editor, generic and analytical files restore all 140 native states in newly started processes. Native-reference checks and standalone replay from raw, generic, and analytical files reproduce all 420 source–prompt outputs pixelwise. The source-shifted TurboEdit predictor is smaller than both its original-draw counterparts and a zero predictor on all 140 sources. Adding the source term reduces file sizes for 131 shifted-target cases while increasing file sizes for all 140 original-target cases.

ReNoise regularization precision. A paired numerical comparison evaluates float16 and float32 regularization while retaining float16 state arrays. Finite inversion completes on 19 of 20 sources in float16 and all 20 in float32. On the 19 shared sources, median analytical savings are $10.5727\,\%$ and $10.5782\,\%$, respectively; fitted savings are $10.8870\,\%$ and $10.9153\,\%$. The remaining source has a KL variance ratio of $557{,}479$, above the float16 maximum $65{,}504$; float32 regularization represents this ratio directly.

G Query Cost, Storage–Computation Trade-off, and Reproducibility

G.1 Warm-query latency and preparation

Storage capacity. For assets with a common generic file size and complete-file reduction fraction γ, a fixed storage budget holds $1/(1-\gamma)$ times as many compressed assets. The reported median reductions correspond to capacity factors of 1.56–1.89. This conversion counts the endpoint and decoder metadata already included in complete-file sizes; realized capacity depends on the file-size mix of the retained sources.

Each of the five configurations uses the same nested 60-source cohort, with six sources per edit type. Each source and arm is timed five times after one warm-up, and we take the within-source median. Tables 25 and 26 report medians across sources and paired per-source ratios, respectively. Ratio intervals use 9,999 edit-type-stratified source-bootstrap resamples.

Models are resident and file reads are page-cache warm. Stored queries include container reading, header and checksum validation, codec decoding and seed regeneration, host-to-device transfer, editing, and output transfer to the host. Recomputation queries include source transfer, inversion, editing, and output transfer. Model loading, initial model/VAE hash verification, host-image preparation, and serialization of output tensors are outside both timers. The editor's scaling constants are set once before timing.

Each configuration uses one dedicated 24-GB RTX 4090 and two pinned CPU threads on a shared host; its three arms share the device and runtime. EF and PixArt use PyTorch 2.4.1/CUDA 12.4 with diffusers 0.32.2; LEDITS++ uses PyTorch 2.0.1/CUDA 11.8 with diffusers 0.27.2. GPU-process monitoring found no competing compute process on any measured device; other configurations may run concurrently on separate devices.

Execution completes successfully on all 60 sources in every configuration. Stored selected queries require 0.379–$0.579\times$ the recomputation cost. Relative to generic storage, the median paired overhead ranges from 0.04 % to 1.16 %.

Table 25: Warm query times with resident models in seconds. S: selected storage; G: generic storage; N: inversion and editing without a stored state. Each entry is the median of 60 per-source trial medians.

Configuration	S	G	N
EF / SD 1.5	6.620	6.615	16.845
EF / SDXL	16.772	16.732	42.578
LEDITS++ / SD 1.5	2.285	2.257	6.015
LEDITS++ / SDXL	12.950	12.831	22.329
EF / PixArt-α	6.378	6.366	16.188

Table 26: Median paired query-time ratios. S/N has a one-sided 95 % upper bound; S/G has a two-sided 95 % confidence interval. Ratios are computed within sources before aggregation.

Configuration	S/N	Upper bound	S/G	95 % CI
EF / SD 1.5	0.3930	0.3931	1.0004	$[0.9996, 1.0009]$
EF / SDXL	0.3944	0.3947	1.0017	$[1.0010, 1.0023]$
LEDITS++ / SD 1.5	0.3793	0.3796	1.0116	$[1.0110, 1.0121]$
LEDITS++ / SDXL	0.5793	0.5795	1.0102	$[1.0095, 1.0105]$
EF / PixArt-α	0.3940	0.3944	1.0006	$[0.9989, 1.0019]$

One-time preparation. Table 27 measures post-inversion preparation on the same 60 sources, using each configuration's recorded runtime, dedicated RTX 4090, and two pinned CPU threads. Preparation costs comprise raw-state reading, candidate encoding, state-only admission and selection, and container packing. One measured pass includes the first source and lazy codec initialization; shared generic-candidate measurements are included in both searches. Comparing preparation from a source image with repeated recomputation also incorporates inversion latency (Equation 16).

Table 27: Post-inversion preparation in seconds, including candidate search and file packing. For each source, we sum the measured preparation components; each entry is the median of these sums over 60 sources.

Configuration	Generic	Selected
EF / SD 1.5	0.438	0.820
EF / SDXL	1.401	2.791
LEDITS++ / SD 1.5	0.377	1.001
LEDITS++ / SDXL	1.347	3.771
EF / PixArt-α	0.433	0.823

Preparation and query cost. Let $P > 0$ be the one-time preparation cost of the stored representation, and let Q_{s} and Q_{n} be its stored-query and recomputation-query costs under the same measurement boundary. The costs of N queries are

$$C_{\mathrm{s}}(N) = P + NQ_{\mathrm{s}}, \qquad C_{\mathrm{n}}(N) = NQ_{\mathrm{n}}. \tag{16}$$

For $\Delta Q = Q_{\mathrm{n}} - Q_{\mathrm{s}} > 0$, the smallest positive query counts for non-strict and strict improvement are

$$N^*_{\leq} = \max\left(1, \left\lceil \frac{P}{\Delta Q} \right\rceil\right), \qquad N^*_{<} = \left\lfloor \frac{P}{\Delta Q} \right\rfloor + 1. \tag{17}$$

If $\Delta Q \leq 0$, no finite crossover exists for positive preparation cost. Here P includes all preparation attributed exclusively to the stored strategy, including inversion; Table 27 measures its post-inversion component.

G.2 EXACT-STATE QUERIES AND CACHE-OBJECT CHOICE

Each study uses 20 sources, resident models, warm page caches, one warm-up, and five paired trials per source and path. We first take trial medians within each source, then report medians across sources; ratios and savings are computed within source before aggregation. ReNoise and the four-step state comparison each use one RTX 4090. The TurboEdit cache-object and PixArt studies distribute sources across four RTX 4090 GPUs with disjoint CPU affinities; all alternatives for a source use the same device. Table 28 reports full query time, including file reading, restoration, input/output transfer, and editing. The source-image path includes VAE encoding and the editor's native source processing. File sizes include the metadata and state required to replay each path.

Table 28: Complete-file size and warm-query latency on 20 sources. File sizes are medians in decimal kB; times are medians of source-level trial medians. Calls count UNet invocations; samples count latent samples processed by those invocations.

Study	Stored object	kB	Query (s)	Calls / samples
Four-step state reuse				
ReNoise	Source image	81.30	4.012	44 / 48
	Generic state	113.57	0.313	4 / 4
	Analytical state	101.59	0.316	4 / 4
TurboEdit	Source image	80.55	0.890	4 / 12
	Generic state	273.37	0.851	4 / 12
	Analytical state	268.63	0.851	4 / 12
TurboEdit cache objects, three prompts per source				
TurboEdit	Source image	80.55	0.917	4 / 12
	Source latent and seed	55.51	0.877	4 / 12
	Analytical state	268.63	0.880	4 / 12
Exact PixArt state reuse				
PixArt	Source image	80.59	15.442	–
	Source latent	66.28	15.439	–
	Generic state	3507.27	6.052	–
	Factorized state	3008.23	6.072	–
	Resident native state	–	6.054	–

Choosing the retained object. ReNoise's analytical-state/source-image median paired time ratio is 0.0795: state reuse avoids the repeated inversion. TurboEdit has the same four UNet calls from its source-image and stored-state paths. Caching its source latent and seed reduces median paired bytes by $31.12\,\%$ and query time by $4.24\,\%$ relative to the image path. This object uses $79.34\,\%$ fewer bytes than the analytical full state. All 60 native-reference reproductions, 120 alternative-path comparisons, and 900 timed outputs match native pixels. Table 2 complements this choice of stored object by comparing encodings of the same materialized state.

Exact PixArt queries. The PixArt native states are generated in the timing runtime. Factorized files save $14.23\,\%$ over generic native state, yielding smaller file sizes across all 20 sources. The median paired warm-query ratio is 1.00229. Preparation accounts for VAE encoding, inversion, codec search, and final encoding. Applying Equation 17 to measured costs gives continuous crossover counts 1.015–1.235, corresponding to two queries at integer counts.

G.3 CONTAINERS, VERIFICATION, ENVIRONMENTS, AND ASSETS

Containers and verification. Generic and selected candidates share a container specification. Complete-file size S^{file} includes B, a per-file integrity checksum, and decoding metadata: the PyTorch version, device/dtype specification, model and VAE identifiers, and schedule digest. Before decoding, the loader checks the payload checksum, container version, PyTorch version, dtype specification, and schedule digest; a mismatch aborts the loading process. When an editing runtime is supplied, it also compares stored model/VAE weight hashes. This comparison is excluded from the timed standalone queries.

The recorded RNG kernel regenerates indexed draws on CUDA for EF and PixArt and on CPU for LEDITS++. Replay uses PyTorch's seeded default generators: EF draws fp32 Gaussian tensors on CUDA, whereas LEDITS++ draws fp16 tensors on CPU before transfer to CUDA. Each draw has shape $1 \times 4 \times H \times W$, with $H = W = 64$ for SD 1.5/PixArt and 128 for SDXL, and follows increasing diffusion time. EF replays the first 64 draws of its 100-step inversion; LEDITS++ replays 50 draws, after discarding the initial VAE-sampling draw for SDXL. The file records the seed, shape, draw recipe, device, schedule, and codec PyTorch version. Coefficient arithmetic and tensor addition reconstruct the controls without denoiser evaluation. The float32 uncompressed-defect round trip checks reassembly under the editor-specific schedule; standalone verification compares full decoded-state hashes with external references. The manifest and container bind the schedule to the editor and RNG settings. Coded-defect error is measured in reconstructed controls and edited outputs. The verified lossy variant checks stored controls and endpoint fingerprints during decoding (decoded-state verification below); the exact-state decoder checks stored predictor and native-control fingerprints (Appendix F.1).

CUDA replay was verified on RTX 4090 GPUs. Median skeleton-regeneration times are 8.1 ms for EF/SD 1.5 and PixArt and 9.2 ms for EF/SDXL, versus 43.4 and 128.7 ms for LEDITS++/SD 1.5 and SDXL on CPU. With CUDA disabled and the recorded PyTorch version 2.0.1, all 840 evaluated LEDITS++ containers decoded bitwise in newly started processes, at a median of 1.3 s (SD 1.5) and 1.4 s (SDXL) per container. Complete-file validation covers 3,080 standalone loads across the five configurations.

For PixArt, the lossy study passes all 140 uncompressed-factorization identity checks and 280 standalone reloads. The exact-state predictor comparison in Appendix D.5 includes 360 encoded files and 160 decoding records on 40 sources. The exact-state timing study in Appendix G.2 uses states generated in its measurement runtime, with 140 candidate encode/decode round trips verified.

An asset can be migrated by decoding it under a supported runtime and losslessly storing the native controls and endpoint. The migrated asset then decodes without seed replay. On four LEDITS++ assets, migration preserves the decoded-state fingerprints and the existing lossy error. The resulting files successfully complete eight read trials across two runtime environments on the same little-endian host; their total size increases from 7.18 to 13.97 MB.

For fp16 editors, rounding the fp32 reconstruction to fp16 can increase the final control error beyond the nominal codec tolerance. The codec overshoot bounds above cover the fp32 reconstruction stage. Across both 140-source LEDITS++ configurations at $\varepsilon = 3 \times 10^{-3}$, all 840 selected and generic container replays have native maximum-absolute error 0.00390625. Selected, best-generic, and matched-generic counterparts match in maximum-absolute error after conversion; their mean-squared errors can differ. All reported edit metrics use the native-dtype controls consumed by the editor.

Environment records. We record each evaluation's configuration, seed manifest, model revisions, command, raw outputs, and available environment lockfile (Table 29). Editor execution and codec checks have separate environment records, and containers are decoded under their encoding environment. Replaying two sources per configuration on two physical RTX 4090 devices with the same host and software gives identical controls and endpoints in all ten comparisons; EF/PixArt regenerate draws on CUDA and LEDITS++ on CPU. To test RNG replay across PyTorch versions, we compare 100-draw streams for two seeds and two tensor shapes between PyTorch 2.0.1 and 2.4.1 on the same RTX 4090. All four full-stream hashes match. This draw-replay test runs without model evaluation.

Decoded-state fingerprint verification. To verify the controls supplied to the editor, the verified lossy variant stores SHA-256 fingerprints of its decoded controls and endpoint, along with their shapes, dtypes, byte order, and contiguous memory order. The decoder checks the reconstructed tensors against these fields before returning them. The fingerprint reference is the lossy reconstruction used for editing. Verification leaves the coded payload, endpoint bytes, and reconstructed editor inputs unchanged.

Across all 700 tasks and five configurations, 7,000 generic and factorized containers reproduce the reference controls and endpoints in newly started processes. We test rejection by changing endpoint bytes and perturbing controls for each stored mode in each configuration (14 cases); all 28 mismatches are rejected. The factorized-file test perturbs controls by changing the regenerated draws.

Table 29: Software environments for the listed execution stages. NTI inversion and editing ran under torch 2.0.1+cu118 without a recorded package lockfile.

Environment	torch	Other packages	Scope
A	2.0.1+cu118	diffusers 0.27.2; transformers 4.30.2	Primary U-Net EF and LEDITS++ extraction and editing; LEDITS++ coding and warm queries
B	2.4.1+cu124	diffusers 0.32.2; transformers 4.44.2	EF codecs and reload; NTI coding/decoding and containers; EF/PixArt warm queries; 700-task exact-state coding and standalone verification
C	2.0.1+cu118	diffusers 0.27.2; transformers 4.44.2	Primary PixArt extraction and editing
D	2.4.1+cu124	diffusers 0.31.0; transformers 4.44.2	FlowEdit/SD3 generation

Each verified file adds 516–523 B, including fingerprint metadata and the source-file digest. All 3,500 paired comparisons retain smaller factorized files, with configuration-level median complete-file savings of 35.77–$47.03\,\%$.

Container decode timing uses 20 development sources per configuration (Appendix D.2), warm-cache reads, one warm-up, and seven alternating paired trials. EF/PixArt replay uses RTX 4090 CUDA under PyTorch 2.4.1; LEDITS++ uses CPU under PyTorch 2.0.1. All configurations use two CPU threads. Fingerprint verification adds 0.82–9.13 ms to factorized decoding in median paired time. Appendix G.1 reports end-to-end query timings for the base lossy format.

Models, data, and licenses. Our experiment records include the source manifest, target prompts, random seeds, editor settings, candidate grids, selected representations, charged-byte fields, edit metrics, bootstrap outputs, and container checks for every primary run. Raw outputs are recorded separately for each editor, backbone, and reference or sensitivity setting. Table 30 lists the reference assets and their recorded revisions.

Table 30: Reference assets, revisions, and distribution terms. Local modifications used only for FireFlow and EDICT boundary measurements are not distributed.

Asset and revision/identifier	**License or distribution**
Stable Diffusion 1.5 (Rombach et al., 2022) ID: `stable-diffusion-v1-5/stable-diffusion-v1-5` Revision: `451f4fe16113bff5a5d2269ed5ad43b0592e9a14`	CreativeML OpenRAIL-M
SDXL base 1.0 (Podell et al., 2024) Revision: `462165984030d82259a11f4367a4eed129e94a7b`	CreativeML OpenRAIL++-M
PixArt-α (Chen et al., 2024) ID: `PixArt-alpha/PixArt-XL-2-512x512` Revision: `50f702106901db6d0f8b67eb88e814c56ded2692`	CreativeML OpenRAIL++-M
SDXL VAE (fp16 fix) ID: `madebyollin/sdxl-vae-fp16-fix` Revision: `207b116dae70ace3637169f1ddd2434b91b3a8cd`	upstream terms
PIE-Bench v1 (Ju et al., 2024) Source: PnP Inversion; artifact manifest	distributed through the source's form
LEDITS++ (Brack et al., 2024) Reference code; pinned in artifact	upstream terms

Compute. Model evaluations use RTX 4090 GPUs with 24 GB of memory. The 140-source reconstruction and editing evaluations require 4.81 GPU-hours for EF/SD 1.5, 8.81 for EF/SDXL, 1.79 for LEDITS++/SD 1.5, 7.49 for LEDITS++/SDXL, and 4.39 for PixArt, measured from execution timestamps. These are per-evaluation costs. Table 27 gives state preparation costs, and Appendix G.1 defines and measures query costs.

H Generalization and Other Persistent Objects

H.1 Transfer to external image–caption cohorts

We evaluate fixed representations on 100 Emu Edit sources (Sheynin et al., 2024), a fixed 24-example EditBench subset (Wang et al., 2023), and 100 MagicBrush evaluation sources (Zhang et al., 2023). Emu Edit supplies public source–target captions. EditBench uses its masked RGBA input composited on a black background, a fixed source caption, and the released target caption. MagicBrush uses caption adaptation of first-turn examples from its development split: two of the first 102 sources were development sources, and the remaining 100 form the evaluation cohort. For MagicBrush, 24 caption pairs were written from source and target images before observing model outputs. The remaining 76 were manually written by the authors from source images and public instructions, without inspecting target images or model outputs. We check exact source duplication by comparing image hashes.

The SD 1.5 EF configuration uses 100 steps, skips 36 steps, and uses source/target guidance scales $3.5/15$. LEDITS++ uses 50 steps without skipped or zero-noise terminal steps. The nominal codec tolerances are 0.01 for EF and 0.003 for LEDITS++, with fixed representation-selection rules. The PIE evaluation uses LEDITS++ skip-0.15; Appendix C.3 specifies the separate native-input admission condition. Here, we compare complete stored files and measure image deviation from each editor's raw-state output under the specified captions in a common 512-pixel PNG evaluation.

Table 31: External SD 1.5 evaluation; n is the number of sources. File savings are median paired reductions against the best-performing generic codec. Each LPIPS entry is generic / selected, measured against the raw-state output of the same editor; P95 is nearest rank.

Dataset	Editor	n	File saving	Median LPIPS	P95 LPIPS	Maximum LPIPS
Emu Edit	EF	100	$46.05\,\%$	$0.00098/0.00101$	$0.01239/0.02377$	$0.04000/0.09090$
Emu Edit	LEDITS++	100	$37.26\,\%$	$0.00107/0.00095$	$0.00823/0.00785$	$0.01585/0.01755$
EditBench	EF	24	$48.68\,\%$	$0.00054/0.00048$	$0.01101/0.01867$	$0.02168/0.02397$
EditBench	LEDITS++	24	$39.51\,\%$	$0.00073/0.00060$	$0.00540/0.00442$	$0.01447/0.02325$
MagicBrush	EF	100	$46.13\,\%$	$0.00098/0.00101$	$0.01229/0.01274$	$0.09593/0.04549$
MagicBrush	LEDITS++	100	$36.77\,\%$	$0.00143/0.00135$	$0.01055/0.01116$	$0.07095/0.02203$

Across these six fixed-seed comparisons, selected files have positive median savings and small median deviations from the raw-editor outputs. All selected and generic outputs have raw-reference LPIPS at most 0.1.

SDXL editors. With SDXL, median complete-file savings are $46.36\,\%$ (EF) and $35.13\,\%$ (LEDITS++) on the 100 Emu Edit sources, $50.15\,\%$ and $38.56\,\%$ on the 24 EditBench sources, and $46.35\,\%$ and $35.20\,\%$ on the 100 MagicBrush sources (MagicBrush results in Table 32). On Emu Edit, mean LPIPS to raw-state edits is 0.004854 (selected) versus 0.004948 (generic) for EF and 0.002128 versus 0.002069 for LEDITS++ in the common 512-pixel evaluation. No selected output in any external cohort exceeds LPIPS 0.1.

MagicBrush with SDXL. On the 100 MagicBrush sources, the one-sided 95 % upper bounds for median selected-minus-generic LPIPS at the original output resolution are -0.000125 for EF/SDXL and 0.000111 for LEDITS++/SDXL. The separate 512-pixel PNG evaluation quantifies fidelity to each editor's raw-state output under the same caption adaptation.

For LEDITS++, both representations have maximum native control error 0.00390625; nominal codec tolerance and final fp16 input error are distinguished in Appendix G.3.

Table 32: MagicBrush transfer with SDXL on 100 fixed sources per editor. Savings compare complete files with the best generic representation. The four rightmost columns summarize LPIPS on 512-pixel PNG outputs against each editor's raw-state output; P95 uses the nearest-rank convention. These descriptive output summaries are separate from the original-resolution cohort fidelity tests.

Editor	Storage	File saving	Median	Mean	P95	Maximum
EF	Selected	46.35 %	0.001964	0.004550	0.016810	0.034637
EF	Generic	—	0.002829	0.004951	0.019444	0.038918
LEDITS++	Selected	35.20 %	0.001730	0.002826	0.006972	0.043667
LEDITS++	Generic	—	0.001688	0.002189	0.004589	0.010065

H.2 PixArt-α adaptation and ablations

Our PixArt-α (Chen et al., 2024) adaptation of EF is evaluated on the balanced 140-source cohort with the settings of Appendix B.1. Editing covers 3 representations (generic SZ3, factorized SZ3, and factorized dbz), and byte selection also considers generic dbz and ZFP. Median payload saving is 45.29 [45.07] %; Table 16 reports complete-file savings and output fidelity.

For each flattened non-boundary residual $r = z - q$, a seed-linear diagnostic fits one seed carrier without an intercept, $c = (\xi^\top r)/(\xi^\top \xi)$ and $\delta = r - c\xi$, and reports the explained fraction $1 - \mathrm{Var}(\delta)/\mathrm{Var}(r)$ with centered variance over the flattened row; its median over 63 rows and then over sources is 0.875, using the lower-middle value for even counts.

The scheduler places T steps either at a fixed integer stride from the low-noise end (`leading`) or by rounding equally spaced points spanning the training timestep range (`linspace`). Table 33 compares the diffusers 4-channel guidance convention with `leading` spacing, the native 3-channel guidance convention, and the default `linspace` grid on a fixed 20-source subset with 2 sources per edit type. The `linspace` grid has gaps $\{10, 11\}$ and moves the endpoint from $t = 630$ to $t = 636$; identity checks pass on $20/20$ sources under this non-uniform grid. Relative to the diffusers/`leading` setting, the paired saving differences are +0.69 pp for native guidance (two-sided 95 % interval $[+0.59, +0.73]$) and +0.05 pp for the `linspace` grid ($[-0.03, +0.08]$). Fidelity is measured against the raw-state output of each condition. The median selected-minus-generic LPIPS differences (one-sided 95 % upper bounds) are 0.002328 (0.0077925), 0.001587 (0.0023465), and -0.001518 (0.0056035) in the row order of the table.

Table 33: PixArt-α guidance and timestep-grid ablations on 20 sources. Savings are medians with one-sided 95 % lower bounds. Output LPIPS uses each condition's raw-state reference and benchmark target prompt, with nearest-rank P95. G/S denotes generic/selected storage.

Guidance / grid	Γ_{sel} [LB]	LPIPS > 0.1 (G/S)
Diffusers / `leading`	45.04 [44.72] %	$4/4$
Native / `leading`	45.85 [45.36] %	$1/2$
Diffusers / `linspace`	45.00 [44.75] %	$2/4$

Guidance / grid	Storage	Median	P95	Maximum
Diffusers / `leading`	Generic	0.03452	0.21019	0.21738
	Selected	0.03249	0.11425	0.22777
Native / `leading`	Generic	0.01731	0.08820	0.15256
	Selected	0.01546	0.11482	0.24546
Diffusers / `linspace`	Generic	0.04568	0.14129	0.21238
	Selected	0.04360	0.17330	0.17377

H.3 OTHER PERSISTENT OBJECTS

A retained object's generating computation determines its coding representation. Table 34 groups the objects by that computation. The source-cancellation identity applies to the stochastic controls derived above; optimized embeddings, feature banks, and endpoints have their own predictors and storage costs.

Table 34: Retained objects and their generating computations. Numerical evaluations are identified explicitly; other entries classify the published methods' stored objects.

Object	Methods	Representation and evaluation
Stochastic controls	Edit-Friendly, LEDITS++; ReNoise, TurboEdit; Cora (Alimohammadi et al., 2025)	Lossy for EF/LEDITS++; exact for EF/PixArt, ReNoise, TurboEdit
Optimized embeddings and noise	NTI, ENM, ResEdit (Mokady et al., 2023; Kang & Choi, 2025; Baykal et al., 2026)	Stored predictors and embedding/noise residuals; NTI evaluated in Table 36
Feature banks	PnP, FireFlow, KV-Edit (Tumanyan et al., 2023; Deng et al., 2025; Zhu et al., 2025)	Nonlinear network activations; the tested PnP and FireFlow menus select generic storage
Compact endpoints	EDICT, RF-Inversion, GNRI, Rex, UniEdit-Flow (Wallace et al., 2023; Rout et al., 2025; Samuel et al., 2025; Blasingame & Liu, 2026; Jiao et al., 2026)	Native endpoint latents; Rex also stores a seed; UniEdit-Flow reference code optionally retains a feature bank
Corrections and displacements	DirectEdit, DNAEdit (Yang & Ye, 2026; Xie et al., 2025)	Source-dependent deterministic or optimized tensors
Source responses	FlowEdit, IF-V2V (Kulikov et al., 2025; Kong et al., 2025)	FlowEdit persistence evaluated in Appendix H.4; IF-V2V reuses vectors within a run
Feature reuse	FISEdit, RegionCache, DeepCache (Yu et al., 2024; Li et al., 2026; Ma et al., 2024)	Features reused across requests or denoising steps
Tensor and seed coding	NIRVANA, FlexCache, LatentBox; SeedLM, AWSRC (Agarwal et al., 2024; Sun et al., 2024; Wang et al., 2026; Shafipour et al., 2025; Liu et al., 2026)	Generic tensors or static weights, without the producing solver's stencil

For FireFlow's eight-step value-feature bank, generic storage is selected on all eight sources. A measured workload requires 3.85 s for page-cache-warm reload and 17.8 s for recomputation. The 20-source PnP development comparison selects generic storage with median saving zero, generic-state $L_{\mathrm{mimic}} = 0.13$, and 1/20 catastrophic edits. EDICT's coupled endpoints select generic coding in the storage-only comparison.

FlowAlign and the inversion-free KV-Edit variant rematerialize from source and prompt inputs (Kim et al., 2026; Zhu et al., 2025); the FlowEdit study instead retains the source-response object. NARC reports source reconstruction from quantized EF and LEDITS++ caches (Park et al., 2026); our fidelity reference is each editor's raw-state edit.

H.4 PERSISTENT SOURCE-RESPONSE STORAGE FOR FLOWEDIT/SD3

Storing the source responses of FlowEdit on SD3 renders them reusable across queries, eliminating their default per-query recomputation. We evaluate this persistent representation, which is related to IF-V2V's within-run reuse of source vectors (Kulikov et al., 2025; Kong et al., 2025).

Response representation and coding. FlowEdit constructs $Z_\sigma = (1-\sigma)x_{\mathrm{src}} + \sigma\xi_\sigma$. Holding the seeded draw fixed when differentiating gives the pathwise target $u_\sigma = \xi_\sigma - x_{\mathrm{src}}$. We compare direct coding of the source response, coding its residual from u_σ, and carrier-defect coding

$$\widehat{v}_\sigma^{\mathrm{src}} = u_\sigma + c_\sigma\xi_\sigma + \delta_\sigma, \qquad c_\sigma = \langle \widehat{v}_\sigma^{\mathrm{src}} - u_\sigma, \xi_\sigma\rangle / \|\xi_\sigma\|^2.$$

All three representations retain the source latent, the response representation, and replay metadata. The fidelity reference is uncompressed stored-response replay with the same source and target. The recomputation path uses a different batch size.

Source-inclusive files and standalone reuse. The serialized representation carries the source latent, occupying 524.288 kB per source. Table 35 reports actual complete-file sizes for all 15 sources and three codecs, including this latent, the response representation, manifest, and checksums. Relative to direct response coding, the carrier defect saves a median $20.57\,\%$ with SZ3, $10.01\,\%$ with zfp, and $23.75\,\%$ with dbz. These paired file reductions differ from response-only savings because the source latent is shared overhead.

Table 35: Source-inclusive FlowEdit files on 15 sources. Sizes are medians in decimal MB; reductions are medians of paired per-source file reductions. The uncompressed response plus source latent occupies 17.8299 MB.

Codec	Direct	Source residual	Carrier defect	vs. direct	vs. residual
SZ3	9.2354	7.6803	7.3226	$20.57\,\%$	$4.37\,\%$
zfp	17.5272	16.2038	15.7866	$10.01\,\%$	$2.54\,\%$
dbz	10.2741	7.9095	7.8115	$23.75\,\%$	$0.14\,\%$

We verify standalone reuse in 60 newly started processes, with one process per source for the uncompressed representation and each of the three SZ3-coded representations. Each process checks the reconstructed response hash and restores the model, scheduler, and random-stream configuration from the file before editing. Queries reconstruct the source response from the file and use shared model weights in the recorded software environment. Storing the source latent also eliminates source-image re-encoding.

The fixed prompt set contains 33 source–target pairs: 15 first targets, 10 second targets, and 8 third targets. We compare the 99 compressed outputs with uncompressed replay of the same source and target; the source worst-target median LPIPS is 0.003452 for direct response coding, 0.003668 for the source-state residual, and 0.003692 for the carrier defect, and the maximum over all outputs is 0.015354. All pairs complete with LPIPS at most 0.05. We take the maximum LPIPS over targets within each source, then summarize these source-level values. SZ3 is evaluated for standalone editing and file size; zfp and dbz are evaluated for complete-file size. File savings are descriptive source-level summaries.

H.5 Optimized null-text embeddings and source reconstruction

Null-text inversion (NTI) (Mokady et al., 2023) stores optimized unconditional embeddings. We compare their predictor-inclusive coding on 40 sources. Source reconstruction from the retained state reaches PSNR 28.59 dB for EF and 27.98 dB for NTI; these assets retain source-image information.

Embedding representations. Let u_t denote the optimized unconditional embedding at step t and $u_\emptyset$ the base empty-prompt embedding. Direct coding stores $u_{1:T}$. Base-embedding residuals store $u_t - u_\emptyset$ and carry $u_\emptyset$. Temporal differencing codes the first row and adjacent-row differences; its base-centered variant applies this transform to $u_t - u_\emptyset$. A partial-optimization predictor uses a source-specific trajectory obtained with two or five inner optimization iterations per step and codes the residual from that trajectory. The inner numeric codec is selected from the same fixed menu at each tolerance. Files carry their predictors for denoiser-free reconstruction, and selection below counts the complete serialized predictor size. Recomputing a partial-optimization predictor trades its stored bytes for model computation at query time.

Terminal latents from each evaluated state are decoded with a common fp32 VAE at batch size two. The inputs are cached sampler outputs or terminal latents verified bitwise against their stored references. Sampler runtimes can differ across evaluations; VAE decoding is fixed.

Selection over predictor-inclusive embedding files. An NTI embedding file includes its predictor and header; the complete package also contains the native endpoint. Selection chooses the smallest

eligible SZ3 or dbz file across generic coding, residuals from the base empty-prompt embedding, temporal differences of those residuals, and residuals from a five-iteration-per-step optimization predictor. This exploratory comparison uses the same 40 sources and an achieved-error admission ratio of $[0.8, 1.25]$ relative to the best generic candidate. All 960 candidate embedding files decode successfully in separate processes. In all 240 selected/generic cases, the decoded-state hashes match those of the states that produced the evaluated terminal latents. The decoded images undergo the source-paired quality screen below; output quality does not enter file selection.

The NTI quality screen rejects a candidate if the one-sided 95 % bootstrap lower bound exceeds zero for either of two statistics: the difference between arm medians of L_{mimic} (LPIPS to each source's raw edit), or the difference between arm means of the worst two values. It also rejects candidates with more sources at $L_{\mathrm{mimic}} > 0.5$ than the generic arm. We estimate these statistics with 10,000 source-paired bootstrap resamples.

Table 36 pairs predictor-inclusive storage with the quality screen. At $\varepsilon = 0.01$, the two-sided source-bootstrap interval for median file savings is $[8.17, 12.14]\%$; selected/generic maximum LPIPS is $0.5223/0.4768$, producing the additional > 0.5 event counted by the screen.

Table 36: NTI selection over all eligible predictor-inclusive embedding files, $n = 40$. File savings are per-source medians; package savings include the endpoint and are aggregated across sources. The quality screen includes the additional LPIPS > 0.5 event check.

ε	Smaller files	Median saving	Package saving	LPIPS medians selected/generic	Screen
0.01	37/40	10.19%	9.33%	0.0983/0.0965	Fail
0.003	18/40	0.00%	1.83%	0.0441/0.0347	Pass
0.001	0/40	0.00%	0.00%	0.0114/0.0114	Pass

I Qualitative Galleries

The reference is the unchanged editor's raw-state output under the documented settings. Columns are source, raw-state edit, per-source best generic-codec edit in the fixed suite, replay-factorization edit, $|\mathrm{raw} - \mathrm{generic}| \times 10$, and $|\mathrm{raw} - \mathrm{ours}| \times 10$. Rows were selected without image inspection. Define $\Delta L_{\mathrm{mimic}}$ as selected-state edit LPIPS to the raw-state edit minus generic-state edit LPIPS to the raw-state edit, sorted in ascending order with image-ID tie-breaking. Take one-based ranks $\lceil n/4 \rceil$, $\lceil n/2 \rceil$, and $\lceil 3n/4 \rceil$, then the two largest remaining $\Delta L_{\mathrm{mimic}}$ values and the largest remaining applicable selected-minus-raw unedited-region LPIPS delta. Row IDs are listed in the captions, which also give target instructions in Figures 6–9; Figure 10 prints source and target captions beneath each image row.

Unedited-region metrics apply to sources whose benchmark edit-region annotation covers a nonempty proper subset of the image. This gives 32/40 sources in the EF and LEDITS++ galleries and 16/20 in each SDXL gallery.

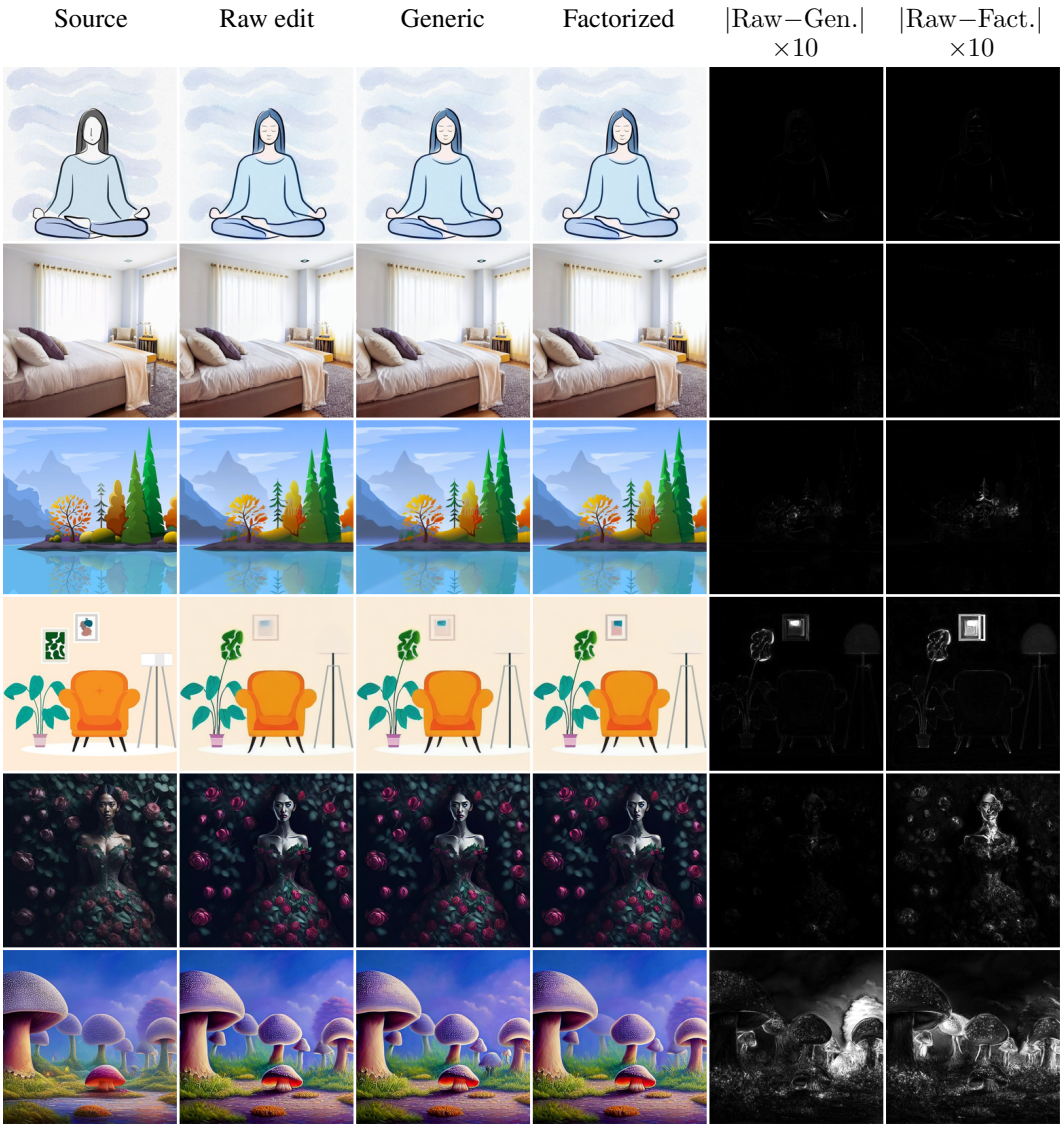

Figure 6: EF/SD 1.5 gallery cohort ($n = 40$). Target instructions by row/image ID: first-quartile example (`812000000006`): "Add stars to the background"; median (`423000000003`): "Make the bedroom mussy"; third-quartile example (`514000000004`): "Change to an overhead shot to the illustration of the lake with trees and mountains"; largest ΔL_{mimic} (`313000000006`): "Remove the chair"; second-largest ΔL_{mimic} (`212000000007`): "Add snakes around the woman"; largest unedited-region delta (`414000000004`): "Change the size of the mushrooms in the painting from huge to small".

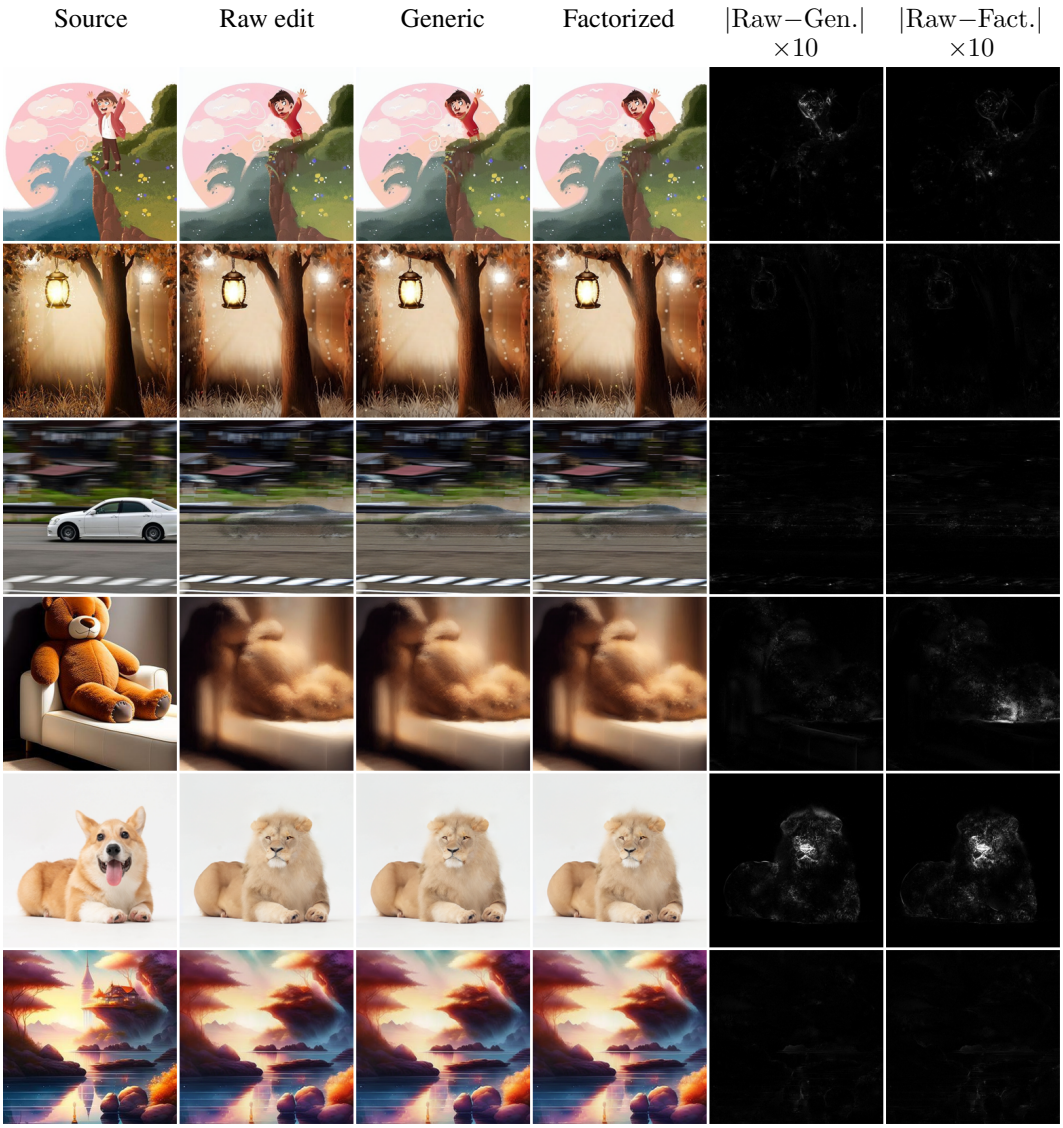

Figure 7: LEDITS++/SD 1.5 gallery cohort ($n = 40$). Target prompts are reproduced verbatim. Target instructions by row/image ID: first-quartile representative (`312000000007`): "Remove sun from the background"; median (`714000000002`): "Change the lantern to a wool lantern"; third-quartile representative (`424000000002`): "Change the color of the car from white to stripe"; largest ΔL_{mimic} (`213000000005`): "Add a girl to the couch with the teddy bear"; second-largest ΔL_{mimic} (`000000000009`): "Change the animal from a dog to a lion"; largest unedited-region delta (`114000000002`): "Change the castle to gaint" [sic].

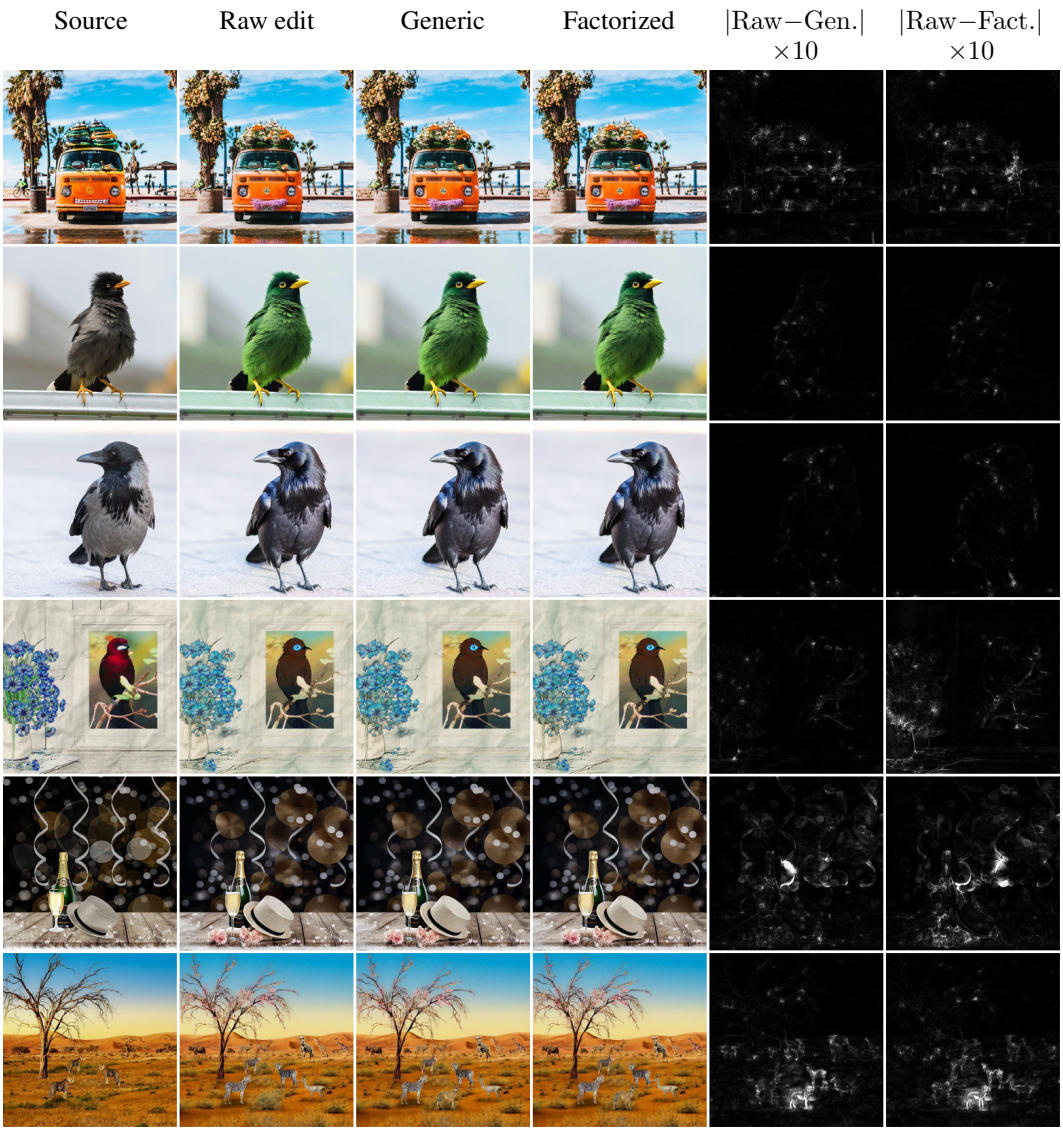

Figure 8: EF/SDXL gallery cohort ($n = 20$). Target instructions by row/image ID: first-quartile representative (`000000000022`): "Replace the surfboards with flowers"; median (`621000000002`): "Change the bird's color to green"; third-quartile representative (`821000000001`): "Change the snow to grass"; largest ΔL_{mimic} (`913000000000`): "Add anime style to the bird"; second-largest ΔL_{mimic} (`213000000006`): "Add flowers to the hat"; largest unedited-region delta (`524000000002`): "Turn the dead tree into a blooming tree".

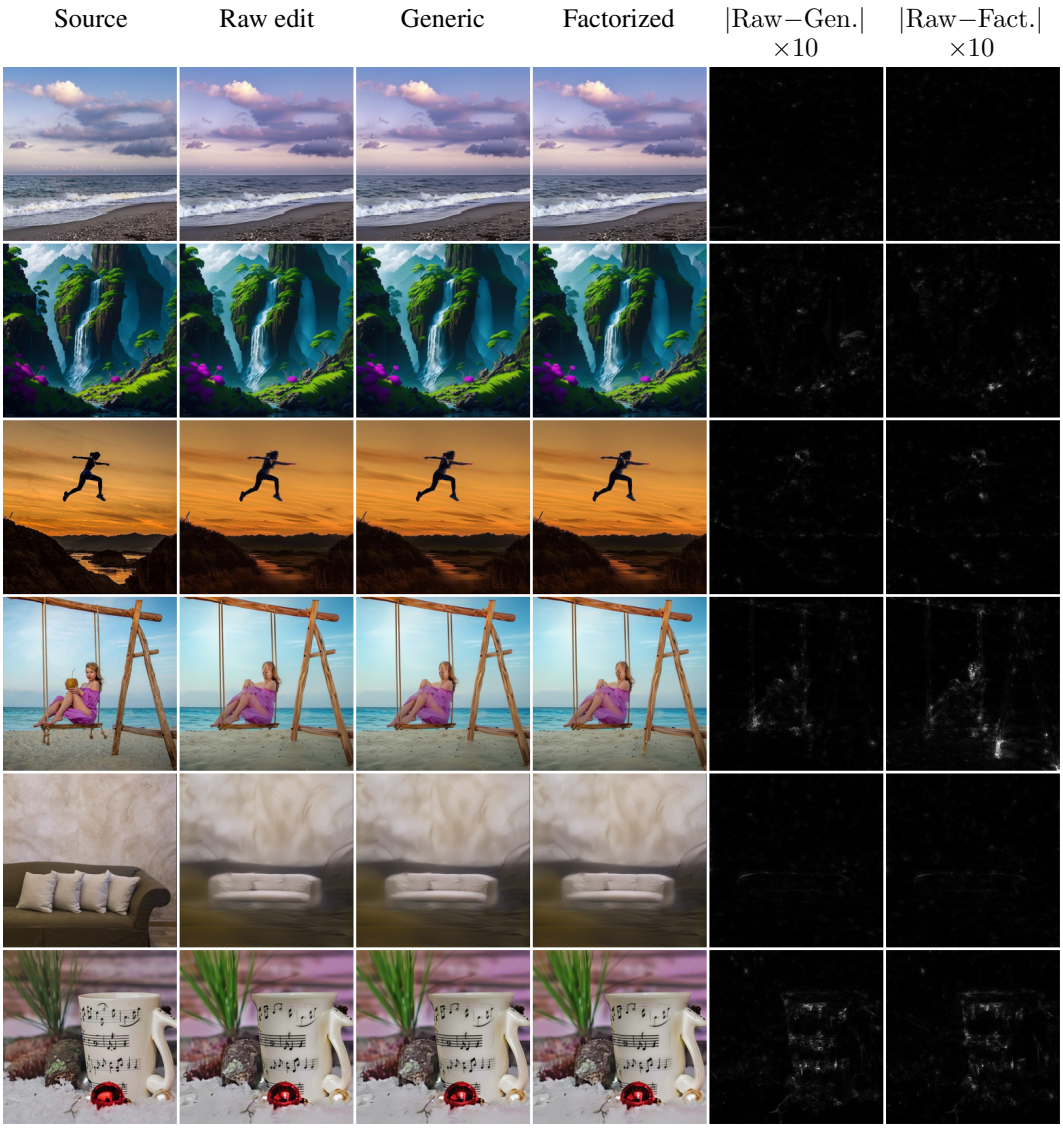

Figure 9: LEDITS++/SDXL gallery cohort ($n = 20$). Target instructions by row/image ID: first-quartile representative (`824000000006`): "Change the weather to a foggy day"; median (`214000000000`): "Add angels to the painting of the waterfall in the mountains"; third-quartile representative (`124000000006`): "Change the object the woman is jumping over to a zombie"; largest ΔL_{mimic} (`322000000002`): "Remove the coconut from the woman's hand"; second-largest ΔL_{mimic} (`823000000008`): "Change the wall to a forest"; largest unedited-region delta (`723000000002`): "Add a Chinese ceramic element to the coffee cup".



Source: a rat is peeking out from behind a pink wall
Target: a digital art of a rat is peeking out from behind a pink wall



Source: a watercolor painting of a couple holding hands
Target: a watercolor painting of a couple holding hands in the field



Source: a brwon cabin in the snow near a lake
Target: a golden cabin in the snow near a lake



Source: spiderman on the roof
Target: robot on the roof



Source: a woman in a dress holding an umbrella with trees in the background
Target: a woman in a dress holding an umbrella with elves in the background



Source: a bird in a cage with a flower on top
Target: a bird in a cage

Figure 10: PixArt-α edits on the 140-source cohort at tolerance 10^{-2}. Rows follow the selection rule of this section; source and target captions are reproduced verbatim below each row. Source IDs, in order: `921000000001`, `812000000009`, `624000000003`, `122000000001`, `822000000006`, and `313000000007`. The last three rows show the largest and second-largest ΔL_{mimic} and the largest unedited-region LPIPS delta.