# PHYSFLOW: PHYSICS-AWARE IMAGE EDITING VIA CONTINUOUS LATENT THOUGHTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Instruction-based image editing has achieved impressive progress in visual quality and prompt adherence, yet maintaining physical consistency remains challenging when edits involve complex dynamics such as refraction, deformation, or object interactions. Existing efforts either rely solely on textual reasoning, which struggles with fine-grained visual changes and lengthy token sequences, or resort to video generative models to synthesize intermediate transformation states, incurring prohibitive sampling overhead. In this paper, we present **PhysFlow**, an efficient continuous-thought framework that jointly internalizes textual editing planning and visual evolution within a compact latent space. By encoding the underlying transformation process as continuous latent thoughts, this way elegantly bypasses the limitations of both language-only reasoning and multi-frame rendering. Specifically, we train PhysFlow via a three-stage curriculum: first learning from explicit multi-step textual rationales and temporally ordered visual states, then progressively compressing them into continuous latent tokens, and finally steering the image editor with the learned latent editing dynamics. To enable this curriculum, we curate a high-quality dataset (namely PhysTrace) spanning diverse physical phenomena, annotated with hierarchical reasoning chains and dense visual trajectories. Extensive evaluations on PICABench and RISEBench demonstrate that our PhysFlow substantially outperforms competitive baselines in physical plausibility while introducing minor inference latency.

## 1 INTRODUCTION

Instruction-guided image editing has witnessed remarkable strides in semantic fidelity and visual realism (Brooks et al., 2023; Liu et al., 2025; Wu et al., 2025a). Nevertheless, enforcing physical plausibility across complex visual scenes poses an open challenge, as an edit may trigger a cascade of coupled environmental responses. For example, turning on a lamp should illuminate the nearby table and curtains and cast physically consistent shadows, as illustrated in Figure 1(c).

A common line of work (Deng et al., 2025; Li et al., 2026b; Ji et al., 2025; He et al., 2026; Yin et al., 2026) typically queries the rich commonsense in multimodal large language models (MLLMs) to unravel interconnected physical consequences and multi-step editing dependencies. Among them, BAGEL (Deng et al., 2025) verbalizes explicit textual thoughts prior to synthesis to deliberate on cascading changes behind an edit. ReasonEdit (Yin et al., 2026) extends this paradigm by adopting an explicit think-and-reflect procedure to refine intermediate reasoning and correct instruction flaws. Ultimately, the enriched textual plans are fed into the generator to steer image synthesis. Despite outlining physical causality, text-only reasoning cannot adequately capture fine-grained visual dynamics involving spatial and geometric shifts, yielding edits with compromised physical fidelity.

Alternatively, another line of work frames physics-aware editing as temporal generation. These approaches employ video generative models to simulate visual evolution and extract the final edited image from candidate frames. Specifically, ChronoEdit (Wu et al., 2025b) treats intermediate frames as visual reasoning tokens, discarding them early during denoising to bypass full video rendering while preserving the terminal states for editing. Although video-based editors harness learned physical priors to model visual state transition, they inevitably incur prohibitive temporal denoising overhead.



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.

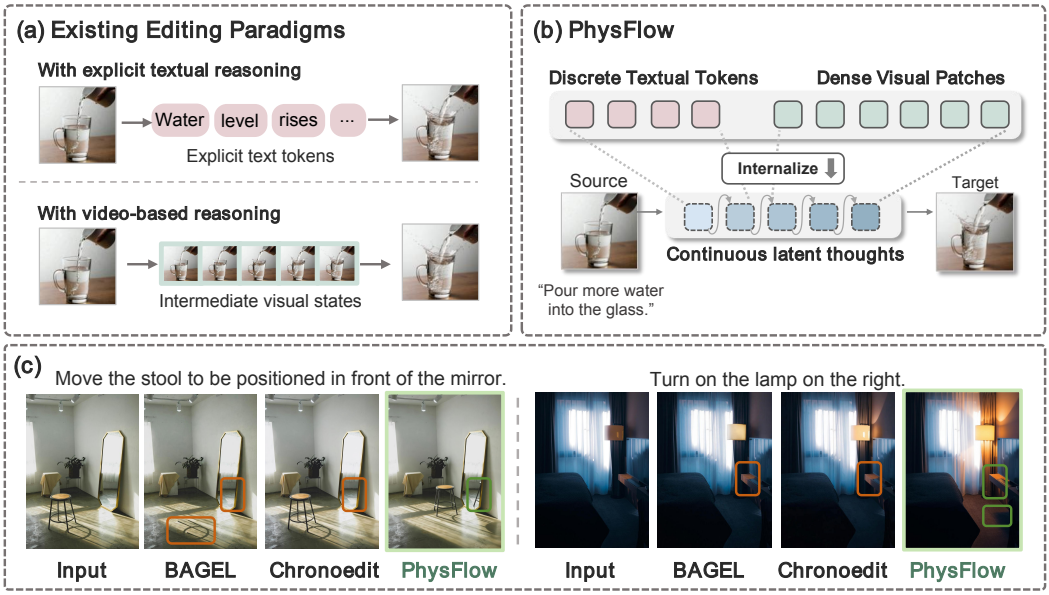



Figure 1: Conceptual comparison and qualitative results. (a) Existing editors either rely solely on explicit textual reasoning, which struggles with fine-grained visual changes, or employ video models to synthesize intermediate states, incurring prohibitive generation costs. (b) Our proposed PhysFlow internalizes textual planning and visual transitions as continuous latent thoughts, mitigating both limitations. (c) Qualitative examples generated by existing state-of-the-art models and PhysFlow.

The above analysis raises a key question: can textual editing planning and visual transformation dynamics be unified into a compact reasoning process, which mitigates both the perceptual limitations of text-only reasoning and the extra inference costs of long text-token decoding and video synthesis?

To this end, we introduce **PhysFlow**, an image editing framework that internalizes such joint reasoning directly into continuous latent thoughts. As illustrated in Figure 1(b), PhysFlow organizes multimodal physical reasoning as a cohesive latent roadmap, where high-level causal deductions ground and orchestrate the low-level fine-grained visual state evolutions. We achieve this through a progressive training curriculum: starting from an explicit reasoning warm-up supervised by paired CoT textual rationales and visual intermediate states, we then distill this explicit trajectory into compact latent thoughts, ultimately conditioning the downstream generator on these representations. At inference time, given the source image and instruction, PhysFlow conducts reasoning within the continuous space, bypassing autoregressive text decoding and intermediate frame rendering. To facilitate this learning process, we construct **PhysTrace**, a benchmark comprising 11,448 video trajectories across six categories of physical transformations synthesized by an advanced video model (NVIDIA, 2026). All instances undergo rigorous dual-level MLLM verification, evaluating both static editing fidelity and continuous transition dynamics. Each trajectory is annotated with hierarchical causal rationales alongside change-uniformly sampled keyframes, providing dense supervision for multimodal physical reasoning in image editing.

In summary, our main contributions are as follows:

- We introduce PhysFlow, a physics-aware image editing framework that embeds causal editing planning and physically consistent visual transition dynamics into continuous latent thoughts via a progressive three-stage curriculum, effectively overcoming the core perception and latency bottlenecks of prior editors.
- We introduce PhysTrace, a high-quality dataset covering diverse physical transformations. Synthesized by an advanced video foundation model (NVIDIA, 2026) and refined via dual-level MLLM verification, it provides high-fidelity supervision that empirically enables more effective training than existing datasets (Pu et al., 2026; Zhao et al., 2026).

- Extensive experiments on PICABench (Pu et al., 2026) and RISEBench (Zhao et al., 2025) demonstrate that our PhysFlow consistently outperforms strong baselines, producing more physically consistent image edits with minor inference overhead.

# 2 Related Work

## 2.1 Textual Reasoning for Image Editing

Although instruction-based image editing has achieved remarkable visual fidelity and prompt alignment, it still struggles with multi-step transformations and physics-demanding edits, where concise user prompts provide neither sufficient guidance nor the requisite causal constraints. To bridge this gap, recent studies use large language models (LLMs) or multimodal LLMs (MLLMs) to enrich editing guidance through explicit textual reasoning. For instance, BAGEL and ThinkRL-Edit derive intermediate linguistic rationales before image generation to clarify implicit user intents and specify editing constraints (Deng et al., 2025; Li et al., 2026b). PRG (Ji et al., 2025) and X-Planner (Yeh et al., 2026) further break down complex commands into actionable subtasks while localizing target regions. Despite these successes in high-level task planning and reasoning, such language-driven methods struggle to faithfully capture fine-grained visual transformation dynamics throughout complex edits.

## 2.2 Visual Transition Modeling for Image Editing

An emerging paradigm leverages the priors of video generative models (Wan et al., 2025; NVIDIA, 2026) to model the intermediate visual evolution in complex editing tasks, thereby enhancing performance particularly in physics-grounded scenarios. Early work curates trajectory-level data to fine-tune video generators. Specifically, AURORA (Krojer et al., 2024) curates editing triplets from videos and simulations. PICABench (Pu et al., 2026) also provides source-target pairs from videos annotated with textual reasoning chains for supervision, alongside a benchmark for physical consistency evaluation. Frame2Frame (Rotstein et al., 2025) steers video generation for image editing using prompts that depict temporal transformations and extracts the edited image from the generated sequence. To reduce inference cost, methods such as ChronoEdit (Wu et al., 2025b) improve inference efficiency by discarding intermediate frame tokens after early diffusion steps. IF-Edit (Zhang et al., 2026) combines a similar latent dropout mechanism with explicit chain-of-thought planning and a post-refinement stage. Although these frameworks capture visual transition dynamics, they require full or partial temporal denoising at inference, incurring substantial computational overhead.

## 2.3 Latent Reasoning

Many MLLMs now explore latent reasoning for intermediate prediction in continuous hidden states, facilitating answer prediction and downstream planning. With textual supervision, latent reasoning methods progressively remove CoT steps (Deng et al., 2024) or replace them with recurrent hidden states (Hao et al., 2024). For instance, CODI transfers explicit reasoning into latent states through hidden-state alignment before answer generation (Shen et al., 2025), while SIM-CoT provides step-level supervision through auxiliary text reconstruction (Wei et al., 2026). Beyond text, LVR uses embeddings of question-relevant image regions to supervise latent reasoning for visual question answering (Li et al., 2026a), while Monet combines image-informed latent distillation with policy optimization (Wang et al., 2026). Visual supervision can also extend to future states: IVT jointly trains textual answer generation and future-frame embedding prediction to support video event anticipation (Zhu et al., 2026). Latent reasoning further benefits diverse downstream tasks like robotic control and autonomous driving, where models learn compact latent plans to guide subsequent action and trajectory generation (Huang et al., 2026; Bai et al., 2026).

# 3 Method

## 3.1 Overview

Instruction-based image editing maps a source image $I_s$ and an instruction $c$ to an edited image $\hat{I}_t = \mathcal{F}(I_s, c)$. However, physics-aware editing further requires modeling how the requested change

conforms to physical laws to shape the resulting scene. Existing methods either extract world knowledge from large language models (LLMs) via text-only reasoning to formulate editing plans, or leverage physical priors in video generative models to explicitly render visual transition sequences. To address the limited perception of fine-grained visual changes in the former and the computational overhead of temporal sampling in the latter, PhysFlow introduces continuous latent thinking to reason about causal dependencies and visual state evolution before image generation.

As illustrated in Figure 2, PhysFlow builds on Qwen-Image-Edit-2511 (Wu et al., 2025a; Qwen Team, 2025), adapting its Qwen-VL encoder into a latent planner $P_\theta$ while retaining its multimodal diffusion transformer (MMDiT) as the image generator $G_\phi$. The planner produces a latent plan $Z$ and visual patches $F$ of the target edit that supplement the generator's native image and instruction conditioning:

$$[Z, F] = P_\theta(I_s, c), \qquad \hat{I}_t = G_\phi(I_s, c, Z, \mathrm{MLP}_\psi(F); \epsilon), \tag{1}$$

where $\epsilon$ denotes the initial generation noise and $\mathrm{MLP}_\psi$ is a feature adapter with parameters $\psi$. To learn $Z$ and $F$, we construct PhysTrace with additional chain-of-thought textual rationales and visual evolution trajectories as supervision for each editing sample $(I_s, c)$ (Section 3.2). We detail our PhysFlow in Section 3.3 and its three-stage training procedure in Section 3.4.

### 3.2 PhysTrace Dataset Construction

We construct **PhysTrace**, a high-quality dataset of approximately 11K trajectories spanning six physical domains and 26 subcategories. The pipeline (Figure 5) integrates video synthesis, dual-level verification, hierarchical reasoning annotation, and change-uniform frame sampling, yielding high-fidelity multimodal supervision that pairs causal rationales with intermediate visual states to facilitate effective training. Specifically, we first select approximately 60K image–instruction pairs from PICA100K (Pu et al., 2026) and augment underrepresented physical categories. Leveraging the physical priors of Cosmos 3 Super (NVIDIA, 2026), we generate videos conditioned on source images and detailed prompts expanded from the editing instructions. Subsequently, we perform dual-level filtering using Qwen2.5-VL-72B-Instruct (Bai et al., 2025), checking instruction compliance at the outcome level while assessing physical plausibility and temporal coherence at the procedure level. Only videos that pass both verification stages are retained, with their final frames taken as the editing targets. Moreover, hierarchical reasoning annotations for each video are derived using Qwen2.5-VL-72B-Instruct, comprising high-level physical rules and causal dependencies as well as low-level transition steps. Alongside the endpoint frame, four intermediate frames are sampled at approximately equal increments of cumulative visual change. The adjacent-frame change for the $j$-th frame is computed as

$$s_j = 0.6\, d_j^{\mathrm{D}} + 0.3\, d_j^{\mathrm{R}} + 0.1\, d_j^{\mathrm{P}}, \tag{2}$$

where $d_j^{\mathrm{D}}$, $d_j^{\mathrm{R}}$, $d_j^{\mathrm{P}}$ are DINOv2 feature differences (Oquab et al., 2023), RAFT optical flow (Teed & Deng, 2020), and pixel differences, respectively. Further details on $d_j^{\mathrm{D}}$, $d_j^{\mathrm{R}}$, $d_j^{\mathrm{P}}$ are provided in Appendix A.2.

### 3.3 PhysFlow

**Continuous Latent Representation.** In this work, we leverage textual reasoning to specify the underlying physical principles and transition steps, and visual reasoning to track fine-grained visual progression toward the desired outcome. Both forms of reasoning are internalized into the continuous latent plan $Z$ to jointly guide image generation:

$$Z = [h, l, m_1, m_2, m_3, m_4] \in \mathbb{R}^{6\times d}, \tag{3}$$

where $d$ denotes the hidden dimension. $h$ and $l$ denote the textual latent tokens for high-level physical causality and low-level transformation process, respectively, each allocated a single token. $m_1, \ldots, m_4$ denote four visual latent tokens representing intermediate evolutionary visual states.

**Causal Prediction.** The latent planner $P_\theta$ in our PhysFlow predicts both the latent plan $Z$ and the visual patches $F$ of the edited image, which are utilized to steer the image editing. We arrange latent thought prediction in a causal order, from high-level physical constraints through low-level transition details to successive visual states. This ordering enables textual physical rationales inferred from the instruction to guide the visual evolution. Specifically, unlike discrete text generation, $P_\theta$ generates

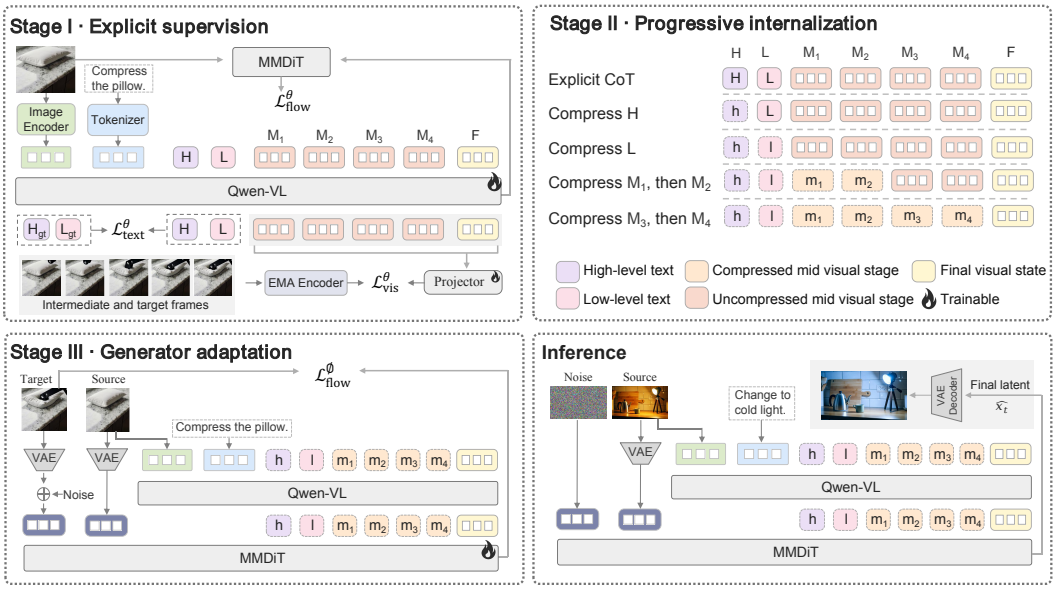



Figure 2: Overview of our PhysFlow. Training proceeds in three stages: (i) Explicit reasoning warm-up supervised by paired textual chain-of-thought (CoT) rationales and visual intermediate states. (ii) Progressive transition from explicit text CoT and visual state tokens into compact continuous representations. (iii) Generator adaptation on the learnt latent representations. (iv) Inference.

$h$ and $l$ sequentially by feeding the final-layer hidden state of the preceding position as the input embedding for the subsequent position. Given the prior tokens of the source image $I_s$, the original instruction $c$ and the textual latent thoughts $[h, l]$, $P_\theta$ then infers the visual transitions $[m_1, \ldots, m_4]$ and the visual patches $F$. We treat each textual token and the visual tokens of each image as separate attention blocks. Block-causal attention is applied during prediction: causal masking is enforced between blocks, whereas bidirectional self-attention is maintained within each visual block to model spatial dependencies.

**Generator Conditioning.** Finally, we condition the MMDiT generator $G_\phi$ on the planner outputs $Z$ and $\mathrm{MLP}_\psi(F)$, the VAE-encoded source image, and the native conditions extracted by Qwen from the source image and editing instruction. The generator iteratively denoises the target latent, which the VAE then decodes into the edited image.

### 3.4 Three-Stage Training Curriculum

**Stage I: Explicit Reasoning Supervision.** In the first stage, we fine-tune both the transformer backbone and the visual encoder $\mathcal{E}_{\theta_v}$ of the planner $P_\theta$ on PhysTrace (Section 3.2) with explicit textual and visual supervision to establish structured reasoning that connects causal understanding with visual state prediction.

For textual supervision, we use the ground-truth high- and low-level reasoning trajectories $H$ and $L$ to supervise the planner through teacher forcing. For the text sequence $T = [H, L]$, we minimize the average negative log-likelihood:

$$\mathcal{L}^\theta_{\text{text}} = -\frac{1}{|T|}\sum_{i=1}^{|T|} \log p_\theta(t_i \mid t_{<i}, I_s, c), \tag{4}$$

where $t_i$ is the $i$-th ground-truth token, $t_{<i}$ denotes the preceding tokens, and $p_\theta$ is the planner's conditional token distribution.

For visual reasoning, we use the intermediate images $I_1, \ldots, I_4$ and final target image $I_t$ as supervision signals. These images are processed by the exponential moving average (EMA) version $\mathcal{E}_{\bar{\theta}_v}$ of the visual encoder and pooled into $4 \times 4$ grid tokens to obtain target features $Y_1, \ldots, Y_4, Y_{\text{final}} \in \mathbb{R}^{16\times d}$, which prevents representation collapse and ensures training stability. Let $M_k$ and $F$ denote the reasoning block predicted by $P_\theta$ for the $k$-th intermediate visual state $I_k$ and the target edit $I_t$,

respectively, each comprising 16 visual tokens. Given the source image, instruction, textual rationales, and preceding blocks $M_{<k}$, the planner predicts the next state:

$$M_k = P_\theta(I_s, c, H, L, M_{<k}), \qquad k = 1, \ldots, 4. \tag{5}$$

Subsequently, the final block $F$ is predicted from the complete latent reasoning sequence. We align the predicted visual reasoning tokens with their ground-truth targets via an adapter $\mathrm{MLP}_\psi$:

$$\mathcal{L}_{\mathrm{vis}}^\theta = \sum_{k=1}^{4} \left\| \mathrm{MLP}_\psi(M_k) - Y_k \right\|_1 + \left\| \mathrm{MLP}_\psi(F) - Y_{\mathrm{final}} \right\|_1 . \tag{6}$$

The parameters of the momentum encoder $\mathcal{E}_{\bar{\theta}_v}$ are updated as:

$$\bar{\theta}_v \leftarrow \mu \bar{\theta}_v + (1 - \mu)\theta_v, \tag{7}$$

where $\mu \in [0, 1)$ denotes the EMA momentum decay. The teacher receives no gradient updates and evolves solely through this moving average after each optimizer step.

Beyond alignment with ground-truth text and visual trajectories, we apply a flow-matching loss with the image generator $G_\phi$ frozen, thereby tailoring the learned representations to the downstream image editing task. The VAE first encodes the source and target images $I_s$ and $I_t$ into their latent representations $x_s$ and $x_0$, respectively. We then sample Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ alongside a noise level $\sigma \in [0, 1]$, and construct the perturbed latent $x_\sigma = (1-\sigma)x_0 + \sigma\epsilon$. The generator predicts the velocity vector field:

$$v_\phi = G_\phi\big(x_\sigma, \sigma, x_s; \hat{Z}, \mathrm{MLP}_\psi(F)\big), \qquad \hat{Z} = [H, L, M_1, \ldots, M_4]. \tag{8}$$

The flow-matching loss is formulated as:

$$\mathcal{L}_{\mathrm{flow}}^\theta = \mathrm{MSE}\big(v_\phi, \epsilon - x_0\big). \tag{9}$$

Therefore, the overall training loss for this stage combines the three objectives:

$$\mathcal{L}_{\mathrm{I}}^\theta = \lambda_{\mathrm{text}}\mathcal{L}_{\mathrm{text}}^\theta + \lambda_{\mathrm{vis}}\mathcal{L}_{\mathrm{vis}}^\theta + \lambda_{\mathrm{flow}}\mathcal{L}_{\mathrm{flow}}^\theta. \tag{10}$$

**Stage II: Progressive Internalization for Latent Thoughts.** Building upon the representations learned in Stage I, Stage II progressively internalizes explicit reasoning into compact latent thoughts. Specifically, we sequentially replace the explicit reasoning texts and intermediate visual blocks with individual latent tokens, while preserving the identical trainable components and loss weighting scheme from Stage I. We first replace $H$ with $h$, then $L$ with $l$, and finally $M_1, \ldots, M_4$ with $m_1, \ldots, m_4$ in order. At each replacement, we take the planner's hidden state from the preceding step and feed it back as the next input embedding, following the mechanism described in Section 3.3. When an intermediate visual block $M_i$ is replaced by a latent token $m_i$, its direct alignment loss is removed. As replacement proceeds, the losses on later uncompressed blocks continue to provide supervision through the causal sequence. After all four intermediate blocks are compressed, we keep the final 16-token block $F$ uncompressed and retain its direct alignment loss along with the flow-matching loss. By the end of this stage, the planner conducts reasoning entirely within the continuous latent sequence $Z = [h, l, m_1, \ldots, m_4]$ before predicting the visual features $F$.

**Stage III: Generator Adaptation.** In the final stage, we freeze both the planner $P_\theta$ and projector $\mathrm{MLP}_\psi$, and train only the LoRA parameters of the generator $G_\phi$ via the flow-matching loss $\mathcal{L}_{\mathrm{flow}}^\phi$. This stage adapts the image generator to effectively leverage the latent reasoning trajectory $Z$ alongside the projected target features $\mathrm{MLP}_\psi(F)$, ensuring that the rendered images strictly adhere to the inferred physical principles.

# 4 Experiments

## 4.1 Experimental Setup

**Implementation Details.** We build PhysFlow on Qwen-Image-Edit-2511 (Wu et al., 2025a) and fine-tune it on PhysTrace using LoRA (Hu et al., 2022). We train with AdamW on eight GPUs, using a batch size of 64. The planner $P_\theta$ is trained with a learning rate of $10^{-5}$ in Stages I and II, while the

Table 1: Quantitative comparison on PICABench-Superficial (Pu et al., 2026) evaluated by Qwen2.5-VL-72B-Instruct (Bai et al., 2025).

| Model | LP↑ | LSE↑ | RFL↑ | RFR↑ | DFM↑ | CSL↑ | GST↑ | LST↑ | Overall↑ |
|---|---|---|---|---|---|---|---|---|---|
| **Proprietary Models** | | | | | | | | | |
| Nano Banana | 48.91 | 47.98 | 51.43 | 52.63 | 44.25 | 44.30 | 52.21 | 50.73 | 49.08 |
| GPT-Image-1 | 58.35 | 56.06 | 50.76 | **63.60** | 40.83 | 46.27 | 63.36 | 52.81 | 53.96 |
| Seedream 4.0 | 55.45 | 57.41 | 55.82 | 54.39 | 45.72 | 47.82 | 59.16 | 56.13 | 54.23 |
| Nano Banana Pro | 52.54 | 57.95 | 55.65 | 54.39 | 47.19 | 50.35 | 65.01 | 58.84 | 56.06 |
| GPT-Image-1.5 | 54.96 | 63.56 | 54.30 | 61.40 | 44.99 | 48.95 | 66.11 | 60.29 | 56.60 |
| **Open-Source Models** | | | | | | | | | |
| DiMOO | 38.26 | 24.80 | 30.19 | 46.05 | 24.94 | 31.50 | 18.21 | 31.19 | 28.57 |
| Uniworld-V1 | 33.41 | 29.11 | 32.88 | 46.49 | 26.16 | 32.91 | 18.54 | 29.94 | 29.18 |
| Bagel | 38.50 | 36.39 | 37.27 | 51.75 | 30.32 | 36.29 | 40.18 | 40.12 | 38.23 |
| Bagel-Think | 39.71 | 43.67 | 36.26 | 48.68 | 33.74 | 36.71 | 39.62 | 39.09 | 38.86 |
| OmniGen2 | 36.32 | 41.78 | 45.36 | 48.25 | 32.52 | 36.29 | 42.72 | 33.89 | 39.52 |
| Hidream-E1.1 | 40.44 | 41.24 | 41.15 | 48.25 | 30.32 | 37.13 | 48.23 | 38.46 | 40.95 |
| Step1X-Edit | 37.29 | 43.94 | 40.64 | 43.42 | 32.03 | 35.72 | 48.79 | 38.46 | 40.59 |
| Flux.1 Kontext | 48.43 | 53.64 | 43.84 | 43.86 | 33.74 | 34.04 | 41.06 | 37.01 | 41.07 |
| ChronoEdit | 51.02 | 58.96 | 45.89 | 56.74 | 42.94 | 44.50 | 55.59 | 47.34 | 49.92 |
| Qwen-Image-Edit-2509 | 52.54 | 52.02 | 49.07 | 57.46 | 38.14 | 42.62 | 57.73 | 47.82 | 49.71 |
| Qwen-Image-Edit-2511 | 53.29 | 62.98 | 56.78 | 57.66 | 44.16 | 45.99 | 59.37 | 55.34 | 54.26 |
| PhysFlow (Ours) | **54.45** | **65.19** | **58.52** | **57.99** | **44.91** | **47.31** | **64.32** | **56.43** | **56.32** |

Table 2: Quantitative comparison on RISEBench (Zhao et al., 2025) evaluated by Qwen2.5-VL-72B-Instruct (Bai et al., 2025).

| Model | Temporal↑ | Causal↑ | Spatial↑ | Logical↑ | Overall↑ |
|---|---|---|---|---|---|
| FLUX.1 Kontext | 3.53 | 0.00 | 0.00 | 4.71 | 1.94 |
| OmniGen2 | 2.35 | 1.11 | 0.00 | 2.35 | 1.39 |
| ChronoEdit | 1.18 | 1.11 | 2.00 | 0.00 | 1.11 |
| BAGEL | 5.88 | 7.78 | 3.00 | 2.35 | 4.72 |
| Step1X-Edit | 4.71 | 12.22 | 2.00 | 5.88 | 6.11 |
| Qwen-Image-Edit-2511 | 9.41 | 12.22 | 6.00 | 4.71 | 8.06 |
| **PhysFlow (Ours)** | **11.76** | **13.33** | **7.00** | **5.88** | **9.44** |

generator $G_\phi$ is fine-tuned with $10^{-4}$ in Stage III. We train Stage I for one epoch, Stage II for six epochs, followed by three epochs for Stage III. Further training details are provided in Appendix B.

**Evaluation Benchmarks.** We compare our PhysFlow against state-of-the-art open-source methods (Xin et al., 2025; Lin et al., 2025; Deng et al., 2025; Wu et al., 2026; HiDream.ai, 2025; Liu et al., 2025; Black Forest Labs et al., 2025; Wu et al., 2025a;b) and leading closed-source models (Fortin et al., 2025; OpenAI, 2025b; Team Seedream, 2025; Raisinghani, 2025; OpenAI, 2025c) on two benchmarks: PICABench (Pu et al., 2026) and RISEBench (Zhao et al., 2025). For the main results in Tables 1 and 2, we adopt Qwen2.5-VL-72B-Instruct (Bai et al., 2025) as the judge. To mitigate potential in-family bias arising from the usage of Qwen models in dataset construction and our planner, we additionally report results obtained with GPT APIs in Appendix Tables 7 and 8.

### 4.2 Main Results

**Quantitative Results on PICABench.** Table 1 presents the comparative results of all evaluated models on PICABench-Superficial. PhysFlow achieves 56.32% overall accuracy, leading all compared open-source models across all eight categories. Notably, PhysFlow trails the top-performing proprietary model, GPT-Image-1.5, by a mere 0.28 percentage points. Compared with its base model Qwen-Image-Edit-2511 (Wu et al., 2025a), PhysFlow achieves an overall gain of 2.06 points, with the most pronounced improvements in global state transitions (GST, +4.95), light-source effects (LSE, +2.21), and reflection (RFL, +1.74). Moreover, although ChronoEdit-14B Think ($N_r = 10$) outperforms most image editors by modeling intermediate visual transitions using learned physical priors from video generative models, it still lags behind PhysFlow by 6.40 points. Such empirical gains underscore the effectiveness of jointly modeling textual causal dependencies and continuous visual evolution through latent thoughts in our PhysFlow. This improvement consistently holds when evaluated by GPT-5 (OpenAI, 2025a): PhysFlow reaches 66.43%, compared with 64.99% for Qwen-Image-Edit-2511 (Appendix Table 7).

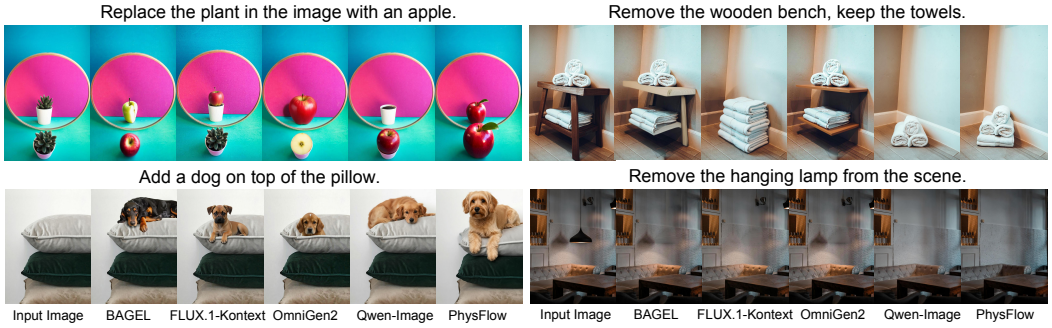



Figure 3: Qualitative comparisons on PICABench-Superficial (Pu et al., 2026).

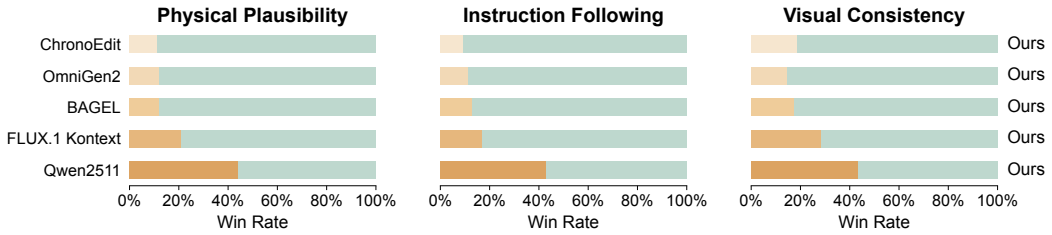



Figure 4: Pairwise user preferences on 100 image-instruction pairs randomly sampled from PICABench (Pu et al., 2026) and RISEBench (Zhao et al., 2025).

**Quantitative Results on RISEBench.** We evaluate PhysFlow on RISEBench (Zhao et al., 2025), which measures strict-completion accuracy on temporal, causal, spatial, and logical editing tasks (Table 2). PhysFlow achieves 9.44% accuracy, consistently surpassing its Qwen-Image-Edit-2511 backbone (8.06%) and all competing baselines to rank first across all four categories. Notably, the most substantial improvement occurs on temporal tasks, where accuracy increases from 9.41% to 11.76%, confirming the benefits of supervising intermediate visual transitions. While ChronoEdit also models visual trajectories using a video generative model, it falls behind PhysFlow. This gap stems from two inherent limits of video-based methods that become critical in physics-aware image editing: unwanted camera movement alters the original scene context, while the lack of reliable frame selection often misses the exact final physical state. This performance advantage also holds when evaluated by GPT-4o (OpenAI, 2024) (Appendix Table 8).

**Qualitative Results.** Figure 3 compares PhysFlow with several strong baselines. Although these baselines generally adhere to edit instructions, they frequently neglect cascading physical consequences across complex contexts such as reflections, mechanical contact, and lighting. In the top-left case, unlike the other models that fail to replace the plant or update the mirror reflection accordingly, PhysFlow successfully executes the replacement while synthesizing a consistent reflection. Similarly, in the bottom-left example, existing methods fail to simulate the compression of the pillow under the dog's weight, whereas PhysFlow realistically renders the resulting elastic deformation. These comparisons show that PhysFlow faithfully propagates localized edits to associated physical consequences across the scene. More comparisons on RISEBench are in Appendix Figure 6.

**User Study.** We conduct a comprehensive user study to assess the physical plausibility and perceptual quality of the edited images. We randomly sample 100 image-instruction pairs from PICABench (Pu et al., 2026) and RISEBench (Zhao et al., 2025) and invite 17 participants from diverse background: computer vision (4), civil engineering (4), mechanical engineering (3), art (3), and general users (3). Participants compare PhysFlow with five competing models on physical plausibility, instruction following, and visual consistency. Figure 4 summarizes the average win rates, showing that participants prefer PhysFlow over all competing models across the three criteria.

### 4.3 Ablation Study and Analysis

**Effect of Pivotal Components in PhysFlow.** In Table 3, we compare explicit textual chain-of-thought (CoT) with latent reasoning, and separately ablate textual and visual latent reasoning to assess their

Table 3: Component ablation on PICABench-Superficial (Pu et al., 2026). Text-CoT denotes explicit textual reasoning; Latent Text-CoT and Latent Vis-CoT denote textual and visual latent reasoning.

| Text-CoT | Latent Text-CoT | Latent Vis-CoT | LP↑ | LSE↑ | RFL↑ | RFR↑ | DFM↑ | CSL↑ | GST↑ | LST↑ | Overall↑ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ✗ | ✗ | ✗ | 53.29 | 62.98 | 56.78 | 57.66 | 44.16 | 45.99 | 59.37 | 55.34 | 54.26 |
| ✓ | ✗ | ✗ | 53.72 | 63.06 | 58.30 | 58.24 | 43.29 | 46.21 | 62.35 | 53.65 | 54.94 |
| ✗ | ✓ | ✗ | 53.87 | 64.26 | 57.82 | **58.92** | 44.28 | 45.67 | 63.01 | 55.51 | 55.40 |
| ✗ | ✗ | ✓ | 53.65 | 64.50 | 57.52 | 58.64 | 44.43 | 46.06 | 62.18 | 55.40 | 55.24 |
| ✗ | ✓ | ✓ | **54.45** | **65.19** | **58.52** | 57.99 | **44.91** | **47.31** | **64.32** | **56.43** | **56.32** |

Table 4: Dataset comparison on PICABench.

| Dataset | Overall ↑ |
|---|---|
| PICA100K-SFT | 53.38 |
| PhysicTran38K | 53.82 |
| **PhysTrace (Ours)** | **56.32** |

Table 5: Computation cost comparison.

| Method | Latency ↓ |
|---|---|
| Qwen-Image-Edit | 44.34 sec |
| **PhysFlow (Ours)** | 49.39 sec |
| Step1X-Edit | 59.18 sec |
| Qwen-Image-Edit + explicit reasoning | 60.12 sec |
| ChronoEdit | 228.68 sec |

individual contributions on PICABench-Superficial. Performance is evaluated using Qwen2.5-VL-72B-Instruct (Bai et al., 2025). As shown, explicit textual CoT (Row 2) improves overall accuracy only slightly over the base model without reasoning (Row 1) (54.94% versus 54.26%), whereas latent textual reasoning (Row 3) further improves upon both and reaches 55.40%. This suggests potential advantages of textual latent reasoning: continuous-space operation may help model physical causal dependencies in internal representations and reduce the accumulation of discrete errors in causal prediction. Visual latent reasoning alone (Row 4) also improves accuracy to 55.24% by effectively capturing subtle visual dynamics, with particular gains on global state transitions and light-source effects. In Row 5, combining the complementary guidance from textual and visual latent reasoning gives the best overall result (56.32%) and leads in seven of eight categories.

**Training Data Comparison.** We train Qwen-Image-Edit-2511 (Wu et al., 2025a) on three datasets: our proposed PhysTrace (11K), PICA100K-SFT (100K), and PhysicTran38K (38K), and evaluate each trained model on PICABench-Superficial (Table 4). Despite using substantially fewer training examples, PhysTrace achieves 56.32% accuracy, outperforming PICA100K-SFT and PhysicTran38K by 2.94 and 2.50 percentage points, respectively. These results suggest that PhysTrace provides more effective supervision through hierarchical textual physical rationales and change-uniformly sampled dense visual intermediate states, enabling learning physically plausible edits more efficiently.

**Computational Efficiency.** We measure inference latency on a single H100 GPU and report the average over 50 image edits in Table 5. PhysFlow takes 49.39 seconds per edit, adding 5.05 seconds (11.4%) over the Qwen-Image-Edit (Qwen Team, 2025). With the same backbone, generating and conditioning on explicit textual CoT takes 60.12 seconds. PhysFlow thus improves editing accuracy at a minor latency cost over the backbone, while remaining faster than explicit textual CoT. It also runs faster than Step1X-Edit (Liu et al., 2025) and ChronoEdit Think ($N_r = 10$) (Wu et al., 2025b).

## 5 Conclusion

In this paper, we introduced PhysFlow, a physics-aware image editor that models causal physical planning and visual state evolution as continuous latent thoughts. Through a progressive three-stage training curriculum, PhysFlow internalizes both textual rationale and visual transition dynamics into compact latent representations, which subsequently steer downstream image editing. This novel design effectively overcomes the limitations of existing paradigms: it bypasses the lengthy token decoding and inability to capture fine-grained visual changes in explicit textual reasoning, while avoiding the prohibitive sampling costs of video-based visual reasoning. To enable this unified latent supervision, we constructed PhysTrace, which provides paired hierarchical causal explanations and dense intermediate visual states. Extensive quantitative and qualitative evaluations demonstrate that PhysFlow substantially improves editing fidelity and physical consistency over leading open-source approaches, while achieving performance competitive with proprietary models.

### AI USE STATEMENT

In this work, we used GPT-6 (OpenAI, 2026) during manuscript preparation to improve grammar, sentence structure, and clarity. The authors take full responsibility for the paper's scientific content, technical accuracy, and conclusions. For PhysTrace construction, we used Cosmos 3 Super (NVIDIA, 2026) to generate videos from source images and detailed prompts expanded from editing instructions. We used Qwen2.5-VL-72B-Instruct (Bai et al., 2025) to check whether each video fulfilled its instruction and whether the sequence of changes was physically plausible and temporally coherent. We retained only videos that passed both checks. The same model then produced hierarchical reasoning annotations, including high-level physical rules and causal dependencies and low-level transition steps.

### ETHICS STATEMENT

All authors have read and adhere to the ICLR 2027 Code of Ethics. We use pretrained models in accordance with their respective licenses and cite their sources. PhysTrace combines image–instruction pairs with synthetically generated transformation videos; we describe their provenance and screening procedure in Section 3.2. PhysFlow can produce realistic images of events that did not occur. Such images could be presented as authentic records or used to mislead viewers. We therefore recommend clear disclosure of edited content and provenance information in downstream applications.

### REPRODUCIBILITY STATEMENT

We detail PhysFlow's architecture and three-stage training curriculum, explain how we construct and filter PhysTrace, and specify our training setup and evaluation protocols in the main paper and appendix. We report the key settings needed to reproduce our experiments, including the training data scale, optimizer and learning-rate schedules, batch size, number of epochs, loss weights, EMA decay, numerical precision. To support reproduction of our results, we will release the code, data-processing scripts, and evaluation pipeline associated with our work.

## REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Shuanghao Bai, Jing Lyu, Wanqi Zhou, Zhe Li, Dakai Wang, Lei Xing, Xiaoguang Zhao, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Badong Chen, and Shanghang Zhang. Latent reasoning vla: Latent thinking and prediction for vision-language-action models. *arXiv preprint arXiv:2602.01166*, 2026. URL https://arxiv.org/abs/2602.01166. Accepted by ICML 2026.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025. URL https://arxiv.org/abs/2506.15742.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging Properties in Unified Multimodal Pretraining. *arXiv preprint arXiv:2505.14683*, 2025. URL https://arxiv.org/abs/2505.14683.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*, 2024. URL https://arxiv.org/abs/2405.14838.

Alisa Fortin, Guillaume Vernade, Kat Kampf, and Ammaar Reshi. Introducing Gemini 2.5 Flash Image, our state-of-the-art image model, 2025. URL https://developers.googleblog.com/introducing-gemini-2-5-flash-image/.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024. URL https://arxiv.org/abs/2412.06769.

Qingdong He, Xueqin Chen, Chaoyi Wang, Yanjie Pan, Xiaobin Hu, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xiangtai Li, and Jiangning Zhang. Reasoning to edit: Hypothetical instruction-based image editing with visual reasoning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2507.01908.

HiDream.ai. HiDream-E1, 2025. URL https://github.com/HiDream-ai/HiDream-E1. GitHub repository; HiDream-E1.1 release.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chi-Pin Huang, Yunze Man, Zhiding Yu, Min-Hung Chen, Jan Kautz, Yu-Chiang Frank Wang, and Fu-En Yang. Fast-ThinkAct: Efficient Vision-Language-Action Reasoning via Verbalizable Latent Planning. *arXiv preprint arXiv:2601.09708*, 2026. URL https://arxiv.org/abs/2601.09708.

Liya Ji, Chenyang Qi, and Qifeng Chen. Instruction-based image editing with planning, reasoning, and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17506–17515. IEEE, 2025.

Benno Krojer, Dheeraj Vattikonda, Luis Lara, Varun Jampani, Eva Portelance, Christopher Pal, and Siva Reddy. Learning action and reasoning-centric image editing from videos and simulation. *Advances in Neural Information Processing Systems*, 37:38035–38078, 2024.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 148076–148090, 2026a.

Hengjia Li, Liming Jiang, Qing Yan, Yizhi Song, Hao Kang, Zichuan Liu, Xin Lu, Boxi Wu, and Deng Cai. Thinkrl-edit: Thinking in reinforcement learning for reasoning-centric image editing. *arXiv preprint arXiv:2601.03467*, 2026b. URL https://arxiv.org/abs/2601.03467.

Bin Lin, Zongjian Li, Xinhua Cheng, Yuwei Niu, Yang Ye, Xianyi He, Shenghai Yuan, Wangbo Yu, Shaodong Wang, Yunyang Ge, Yatian Pang, and Li Yuan. UniWorld-V1: High-Resolution Semantic Encoders for Unified Visual Understanding and Generation. *arXiv preprint arXiv:2506.03147*, 2025. URL https://arxiv.org/abs/2506.03147.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025. URL https://arxiv.org/abs/2504.17761.

NVIDIA. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

OpenAI. Hello GPT-4o, 2024. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. GPT-5 System Card, 2025a. URL https://openai.com/index/gpt-5-system-card/.

OpenAI. Introducing our latest image generation model in the API, 2025b. URL https://openai.com/index/image-generation-api/.

OpenAI. The new ChatGPT Images is here, 2025c. URL https://openai.com/index/new-chatgpt-images-is-here/.

OpenAI. GPT-6 Astra System Card. Technical report, OpenAI, 2026. URL https://deploymentsafety.openai.com/gpt-6-astra.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. URL https://arxiv.org/abs/2304.07193.

Yuandong Pu, Le Zhuo, Songhao Han, Jinbo Xing, Kaiwen Zhu, Shuo Cao, Bin Fu, Si Liu, Hongsheng Li, Yu Qiao, Wenlong Zhang, Xi Chen, and Yihao Liu. Picabench: How far are we from physical realistic image editing? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AWxI5xnuZB.

Qwen Team. Qwen-Image-Edit-2511, 2025. URL https://huggingface.co/Qwen/Qwen-Image-Edit-2511. Hugging Face model card.

Naina Raisinghani. Introducing Nano Banana Pro, 2025. URL https://blog.google/innovation-and-ai/products/nano-banana-pro/.

Noam Rotstein, Gal Yona, Daniel Silver, Roy Velich, David Bensaid, and Ron Kimmel. Pathways on the image manifold: Image editing via video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7857–7866, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Rotstein_Pathways_on_the_Image_Manifold_Image_Editing_via_Video_Generation_CVPR_2025_paper.html.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Team Seedream. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025. URL https://arxiv.org/abs/2509.20427.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020. URL https://arxiv.org/abs/2003.12039.

Siqi Wan, Jingwen Chen, Yehao Li, Yingwei Pan, Ting Yao, and Tao Mei. PhysEdit: Physically consistent image editing via causal enforcement. In *Computer Vision – ECCV 2026*, pp. 430–447. Springer, 2026. doi: 10.1007/978-3-032-37023-5_24.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, et al. Wan: Open and advanced large-scale video generative models, 2025. URL https://arxiv.org/abs/2503.20314.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12030–12040, June 2026.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. Sim-cot: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, volume 2026, pp. 56721–56742, 2026.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun

Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025a. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, Ziyi Xia, Ze Liu, Chaofan Li, Haoge Deng, Kun Luo, Bo Zhang, Jiajun Zhang, Dong Liu, Defu Lian, Xinlong Wang, Zhongyuan Wang, Tiejun Huang, and Zheng Liu. Omnigen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21964–21975, June 2026.

Jay Zhangjie Wu, Xuanchi Ren, Tianchang Shen, Tianshi Cao, Kai He, Yifan Lu, Ruiyuan Gao, Enze Xie, Shiyi Lan, Jose M. Alvarez, Jun Gao, Sanja Fidler, Zian Wang, and Huan Ling. Chronoedit: Towards temporal reasoning for image editing and world simulation, 2025b. URL https://arxiv.org/abs/2510.04290.

Yi Xin, Qi Qin, Siqi Luo, et al. Lumina-dimoo: An omni diffusion large language model for multi-modal generation and understanding. *arXiv preprint arXiv:2510.06308*, 2025. URL https://arxiv.org/abs/2510.06308.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4387. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/bec783f119b969772c35243b1c801090-Abstract-Datasets_and_Benchmarks_Track.html.

Chun-Hsiao Yeh, Yilin Wang, Nanxuan Zhao, Richard Zhang, Yuheng Li, Yi Ma, and Krishna Kumar Singh. Beyond simple edits: X-planner for complex instruction-based image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11991–11999, 2026.

Fukun Yin, Shiyu Liu, Yucheng Han, Zhibo Wang, Peng Xing, Rui Wang, Wei Cheng, Yingming Wang, Aojie Li, Zixin Yin, Pengtao Chen, Xianfang Zeng, Gang Yu, and Daxin Jiang. Reasonedit: Towards reasoning-enhanced image editing models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Zechuan Zhang, Zhenyuan Chen, Zongxin Yang, and Yi Yang. Are image-to-video models good zero-shot image editors? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2090–2103, June 2026.

Liangbing Zhao, Le Zhuo, Sayak Paul, Hongsheng Li, and Mohamed Elhoseiny. From statics to dynamics: Physics-aware image editing with latent transition priors. *arXiv preprint arXiv:2602.21778*, 2026. URL https://arxiv.org/abs/2602.21778.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, Hua Yang, Xue Yang, and Haodong Duan. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fe79898dcf078ec54b6feeea10ebb751-Abstract-Datasets_and_Benchmarks_Track.html.

Xiaoyu Zhu, Xinke Deng, Suresh Taddewadikar, Arnab Kumar Mondal, Zhongyu Jiang, Ian Fasel, and Joerg Liebelt. Beyond visual cot: Internalized visual thinking for proactive video reasoning. *arXiv preprint arXiv:2608.15869*, 2026. URL https://arxiv.org/abs/2608.15869.

# A PHYSTRACE CONSTRUCTION

## A.1 CONSTRUCTION PIPELINE

Figure 5 illustrates the video generation, verification, annotation, and intermediate-frame sampling steps described in Section 3.2.

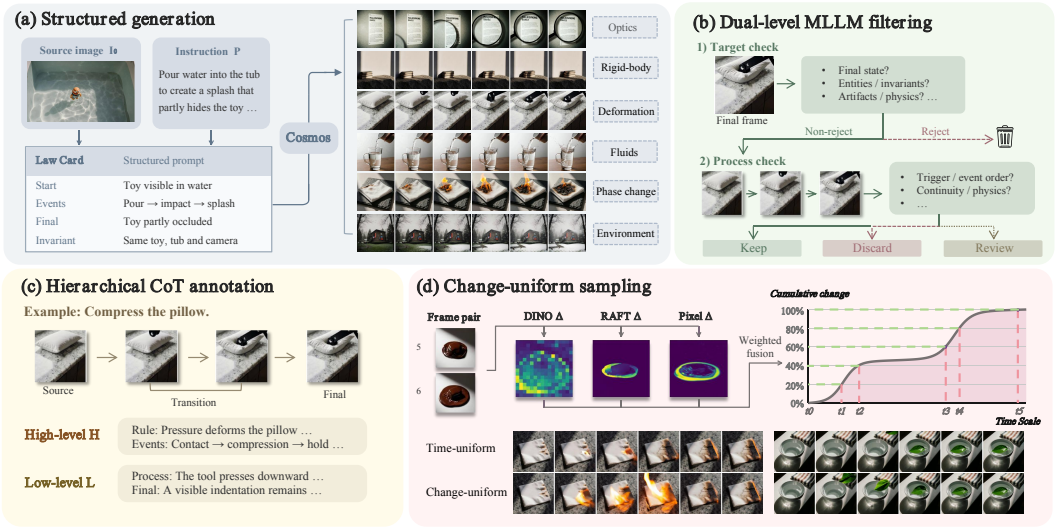



Figure 5: **PhysTrace construction pipeline.** Cosmos 3 Super (NVIDIA, 2026) generates videos from source images and detailed prompts. Qwen2.5-VL-72B-Instruct (Bai et al., 2025) verifies the generated videos and produces hierarchical reasoning annotations. Change-uniform sampling selects the intermediate visual states.

## A.2 CHANGE-UNIFORM FRAME SAMPLING

Physical changes in generated videos are often unevenly distributed over time: some intervals contain substantial transformations, while others show little visible change. Uniform temporal sampling may therefore overrepresent low-change intervals and miss important transitions. To provide more informative visual supervision, we select four intermediate frames that divide the cumulative visual change from the source to the final frame into five approximately equal intervals. For each pair of adjacent video frames, we compute DINOv2 feature change, RAFT motion, and pixel change. Their weighted sum gives the adjacent-frame score in Eq. 2.

For DINOv2 (Oquab et al., 2023), let $\delta_j$ denote the cosine distances between corresponding patch features in adjacent frames. We define the raw feature-change score as a weighted average of the mean distance over all patches and the mean over the 20% of patches with the largest distances:

$$\tilde{d}_j^{\mathrm{D}} = 0.75\ \mathrm{MeanTop}_{20\%}(\delta_j) + 0.25\ \mathrm{Mean}(\delta_j). \tag{11}$$

For RAFT (Teed & Deng, 2020), we subtract the spatial median of each optical-flow component to reduce the effect of global translation. Let $r_j$ denote the magnitude of the residual flow at each pixel, and let $D$ be the frame diagonal in pixels. We define

$$\tilde{d}_j^{\mathrm{R}} = \frac{Q_{0.9}(r_j)}{D} + 0.15\ \mathrm{Mean}\big(\mathbf{1}[r_j > 0.5]\big), \tag{12}$$

where $Q_{0.9}$ denotes the 90th percentile across pixels.

For pixel change, we convert adjacent frames to Lab color space, scale the channel values to $[0, 1]$, and apply Gaussian smoothing. Let $\Delta_j$ denote the absolute difference between the smoothed frames, averaged over the three channels at each pixel. We define

$$\tilde{d}_j^{\mathrm{P}} = Q_{0.9}(\Delta_j). \tag{13}$$

Table 6: Quantitative comparison on ImgEdit-Bench (Ye et al., 2025). Higher scores are better in all categories.

| Model | Add | Adjust | Extract | Replace | Remove | BackG | Style | Hybrid | Action | Overall |
|---|---|---|---|---|---|---|---|---|---|---|
| MagicBrush | 2.84 | 1.58 | 1.51 | 1.97 | 1.58 | 1.75 | 2.38 | 1.62 | 1.22 | 1.89 |
| Instruct-Pix2Pix | 2.45 | 1.83 | 1.44 | 2.01 | 1.50 | 1.44 | 3.55 | 1.20 | 1.46 | 1.98 |
| AnyEdit | 3.18 | 2.95 | 1.88 | 2.47 | 2.23 | 2.24 | 2.85 | 1.56 | 2.65 | 2.50 |
| Step1X-Edit | 3.88 | 3.14 | 1.76 | 3.40 | 2.41 | 3.16 | 4.63 | 2.64 | 2.52 | 3.12 |
| BAGEL | 3.56 | 3.31 | 1.70 | 3.30 | 2.62 | 3.24 | 4.49 | 2.38 | 4.17 | 3.17 |
| UniWorld-V1 | 3.82 | 3.64 | 2.27 | 3.47 | 3.24 | 2.99 | 4.21 | 2.96 | 2.74 | 3.31 |
| OmniGen2 | 3.57 | 3.06 | 1.77 | 3.74 | 3.20 | 3.57 | 4.81 | 2.52 | 4.68 | 3.39 |
| FLUX.1 Kontext | 3.76 | 3.45 | 2.15 | 3.98 | 2.94 | 3.78 | 4.38 | 2.96 | 4.26 | 3.48 |
| ChronoEdit | 4.31 | 4.02 | 3.25 | 4.03 | 4.32 | **4.30** | 4.68 | 3.73 | 4.83 | 4.13 |
| Qwen-Image-Edit-2511 | **4.45** | 4.23 | **4.16** | 4.64 | 4.41 | 4.24 | 4.78 | **3.76** | 4.83 | **4.41** |
| PhysFlow (Ours) | 4.26 | **4.33** | 3.91 | **4.66** | **4.66** | 4.25 | **4.85** | 3.26 | **4.85** | 4.39 |

We normalize each sequence of raw scores to sum to one over the video, yielding $d_j^{\mathrm{D}}$, $d_j^{\mathrm{R}}$, and $d_j^{\mathrm{P}}$. We combine these normalized scores according to Eq. 2 and compute their cumulative sum from the source frame onward. We then select four intermediate frames at cumulative-change fractions of approximately 20%, 40%, 60%, and 80%, providing more informative training supervision.

# B Implementation Details

**Optimization.** We train on 11,448 PhysTrace examples using eight GPUs, a per-device batch size of 1, and 8 gradient-accumulation steps. The effective batch size is 64. We use AdamW with a weight decay of 0.01, cosine learning-rate decay, and a 10% warm-up for each training run. All stages use BF16 mixed precision. In Stage I, we set the main flow, explicit CoT, and visual loss weights to 0.1, 0.1, and 0.25, respectively. Stage II keeps the Stage I loss weights. As each text or intermediate visual state is compressed into a latent token, we remove its direct supervision; the remaining explicit states keep the same loss weights. The teacher visual encoder uses an EMA decay of $\mu = 0.999$ in Stages I and II. In Stage III, we retain only the flow-matching loss, with a weight of 1.0.

**Trainable Modules and Learning Rates.** We apply rank-32 LoRA to the language backbone of the Qwen-VL planner and rank-512 LoRA to MMDiT. In Stages I and II, we train the Qwen-VL LoRA and our visual projector, and update the final layers of the planner's visual encoder. The learning rates are $10^{-5}$ for the Qwen-VL LoRA, $5 \times 10^{-6}$ for the trainable visual components, and $10^{-4}$ for our visual projector. MMDiT and the VAE remain frozen in these stages. In Stage III, we freeze the entire Qwen-VL planner and visual projector, and train only the MMDiT LoRA at $10^{-4}$.

# C Additional Experimental Results

## C.1 Results on ImgEdit-Bench

In addition to PICABench (Pu et al., 2026) and RISEBench (Zhao et al., 2025), we evaluate PhysFlow on ImgEdit-Bench (Ye et al., 2025) to assess whether it retains general image-editing capability. Following PhysEdit (Wan et al., 2026), we use GPT-4.1 as the judge for this evaluation. As shown in Table 6, PhysFlow improves performance across six of the nine categories while maintaining highly comparable performance on the rest. Crucially, the overall score shows only a marginal decrease compared to the base model (4.39 vs. 4.41). These results confirm that PhysFlow significantly enhances physical plausibility without compromising general image-editing capabilities.

## C.2 PICABench Results with GPT-5

Because PhysTrace construction and the PhysFlow planner both use Qwen-family models, we further evaluate PICABench-Superficial with GPT-5 (OpenAI, 2025a) to rule out potential evaluator bias. As shown in Table 7, PhysFlow reaches 66.43% overall accuracy, 1.44 percentage points above its Qwen-

Table 7: PICABench-Superficial (Pu et al., 2026) accuracy (%) evaluated by GPT-5 (OpenAI, 2025a). LP, LSE, RFL, RFR, DFM, CSL, GST, and LST denote Light Propagation, Light Source Effects, Reflection, Refraction, Deformation, Causality, Global State Transition, and Local State Transition, respectively.

| Model | LP↑ | LSE↑ | RFL↑ | RFR↑ | DFM↑ | CSL↑ | GST↑ | LST↑ | Overall↑ |
|---|---|---|---|---|---|---|---|---|---|
| **Proprietary Models** | | | | | | | | | |
| Nano Banana | 60.29 | 59.30 | 65.94 | 53.95 | 59.90 | 55.27 | 60.60 | 59.88 | 59.87 |
| GPT-Image-1 | 61.26 | 66.04 | 62.39 | 59.21 | 59.66 | 52.88 | 70.75 | 59.04 | 61.08 |
| Seedream 4.0 | 62.71 | 65.50 | 65.77 | 53.51 | 59.17 | 53.45 | 65.12 | 66.11 | 61.91 |
| Nano Banana Pro | 60.53 | 70.62 | **70.32** | 57.02 | 64.79 | **58.65** | 72.74 | 70.27 | 66.16 |
| GPT-Image-1.5 | **62.95** | **73.15** | 68.13 | **62.28** | **65.53** | 58.51 | **74.39** | **71.52** | **67.05** |
| **Open-Source Models** | | | | | | | | | |
| DiMOO | 46.00 | 29.38 | 43.68 | 35.53 | 39.36 | 36.71 | 22.52 | 40.54 | 35.66 |
| Uniworld-V1 | 42.37 | 34.50 | 46.04 | 46.05 | 40.10 | 39.52 | 22.85 | 39.50 | 37.68 |
| Bagel | 46.97 | 39.35 | 49.41 | 42.54 | 44.25 | 39.24 | 46.80 | 49.27 | 45.07 |
| Bagel-Think | 49.88 | 50.40 | 47.05 | 43.42 | 49.88 | 38.68 | 45.70 | 50.94 | 46.48 |
| OmniGen2 | 49.64 | 48.79 | 56.49 | 39.04 | 44.74 | 39.80 | 51.10 | 39.09 | 46.79 |
| Hidream-E1.1 | 49.15 | 48.25 | 49.07 | 46.49 | 44.50 | 40.51 | 56.40 | 40.33 | 47.90 |
| Step1X-Edit | 45.04 | 47.44 | 53.46 | 34.21 | 45.72 | 42.90 | 55.85 | 46.57 | 48.23 |
| Flux.1 Kontext | 54.96 | 57.41 | 57.50 | 36.40 | 51.83 | 38.12 | 48.79 | 47.61 | 48.93 |
| ChronoEdit | 59.45 | 67.53 | 57.34 | 56.72 | 56.64 | 51.64 | 64.01 | 55.77 | 58.61 |
| Qwen-Image-Edit-2509 | 62.95 | 61.19 | 62.90 | 55.26 | 48.66 | 48.95 | 67.33 | 54.89 | 58.29 |
| Qwen-Image-Edit-2511 | 63.98 | 74.87 | 71.79 | 63.21 | 60.01 | 53.82 | 70.20 | 62.19 | 64.99 |
| **PhysFlow (Ours)** | **64.09** | **78.10** | **72.69** | **63.70** | **60.22** | **55.39** | **73.23** | **62.50** | **66.43** |

Image-Edit-2511 backbone. PhysFlow improves on its backbone across all eight physical categories and achieves the highest score among the compared open-source models in every category. The largest gains occur in light-source effects (+3.23 points) and global state transitions (+3.03 points). Both tasks require changes beyond the directly edited object, such as updating the surrounding illumination or the broader scene state. This pattern is consistent with PhysFlow's emphasis on the consequences of physical changes. PhysFlow closely rivals the leading closed-source model, GPT-Image-1.5, trailing it by a narrow margin of just 0.62 points.

### C.3 RISEBench Results with GPT-4o

Following prior practice (Zhao et al., 2025), we also evaluate on RISEBench with the closed-source model GPT-4o (OpenAI, 2024). As shown in Table 8, PhysFlow achieves the highest overall strict-completion accuracy of 21.94% among all the models, compared with 20.28% for its Qwen-Image-Edit-2511 backbone. It improves over the backbone in all four reasoning categories. These results confirm that PhysFlow achieves consistent performance gains across different judges, aligning well with the main-text evaluation using Qwen2.5-VL-72B-Instruct.

Table 8: Quantitative comparison on RISEBench (Zhao et al., 2025) evaluated by GPT-4o (OpenAI, 2024).

| Model | Temporal↑ | Causal↑ | Spatial↑ | Logical↑ | Overall↑ |
|---|---|---|---|---|---|
| FLUX.1 Kontext | 4.71 | 2.22 | 17.00 | 2.35 | 6.94 |
| OmniGen2 | 2.35 | 2.22 | 7.00 | 1.18 | 3.33 |
| ChronoEdit | 2.35 | 5.56 | 8.00 | 2.35 | 4.72 |
| BAGEL | 0.00 | 8.89 | 13.00 | 4.76 | 6.96 |
| Step1X-Edit | 9.41 | **23.33** | 17.00 | 7.06 | 14.44 |
| Qwen-Image-Edit-2511 | 14.12 | 22.22 | 35.00 | 7.06 | 20.28 |
| **PhysFlow (Ours)** | **15.29** | **23.33** | **36.00** | **10.59** | **21.94** |

### C.4 Training Dataset Comparison

Table 9 provides the category-level breakdown of the dataset comparison summarized in Table 4. All three models use the Qwen-Image-Edit-2511 backbone. Both PhysicTran38K and PICA100K-SFT contain more training examples than PhysTrace. For PhysicTran38K, we convert its textual CoT

Table 9: Training dataset comparison on PICABench-Superficial (Pu et al., 2026) using the Qwen-Image-Edit-2511 backbone. Scores are editing accuracy (%).

| Configuration | LP↑ | LSE↑ | RFL↑ | RFR↑ | DFM↑ | CSL↑ | GST↑ | LST↑ | Overall↑ |
|---|---|---|---|---|---|---|---|---|---|
| PICA100K-SFT | **55.52** | 60.30 | 57.69 | 53.04 | 41.46 | 45.91 | 59.13 | 52.12 | 53.38 |
| PhysicTran38K | 54.14 | 62.16 | 54.55 | 57.60 | 44.03 | 45.19 | 61.26 | 52.20 | 53.82 |
| PhysTrace (Ours) | 54.45 | **65.19** | **58.52** | **57.99** | **44.91** | **47.31** | **64.32** | **56.43** | **56.32** |

and videos into the same supervision format as PhysTrace, with high- and low-level reasoning and four intermediate visual states. We then train PhysFlow using the same architecture, curriculum, and optimization settings described in Section B. Its larger training data results in 4,632 optimization steps, compared with 1,790 for PhysTrace. PICA100K-SFT has no videos, so we fine-tune it on source–target image pairs for three epochs (4,926 optimization steps). Training with our curated PhysTrace achieves the highest score in seven of the eight categories and outperforms PhysicTran38K in every category. Its largest gains over PhysicTran38K occur in local state transitions (+4.23 points), reflection (+3.97 points), and global state transitions (+3.06 points). PICA100K-SFT holds only a marginal lead in light propagation, scoring 55.52% versus 54.45% for PhysTrace. These comparisons demonstrate that PhysTrace provides substantially more effective supervision, achieving superior performance despite using fewer training samples.

### C.5 Qualitative Results on RISEBench

Figure 6 presents four RISEBench examples involving temporal change, shape change, and spatial movement. In the first two examples, several baselines change the surrounding scene while leaving the main object relatively unchanged: the bench has little accumulated snow, and the underwater boat shows limited deterioration after 50 years. PhysFlow covers the bench with snow and adds visible growth and wear to the boat itself. In the balloon example, Step1X-Edit inflates the balloon bodies but leaves a visible break between the rounded bodies and their narrow necks, making them appear disconnected. Other baselines leave some balloons partly inflated. PhysFlow produces fully inflated balloons with continuous shapes while preserving their colors and arrangement. In the final example, PhysFlow makes the bird larger and turns it toward the viewer, conveying both its approach and its direction of flight.

### C.6 Probing Physical Reasoning through Latent Interventions

We evaluate 64 PICABench-Superficial (Pu et al., 2026) inputs, 8 per physical category, under the original condition and three interventions. These interventions mask all four intermediate visual tokens (Mask Mid), reverse their contents (Reverse Mid), or permute their contents as $(3, 1, 4, 2)$ (Shuffle Mid). We apply these interventions to the intermediate visual tokens after planning, leaving the text conditioning and final visual tokens unchanged. We employ Qwen2.5-VL-72B-Instruct (Bai et al., 2025) as the evaluator to answer physical validation questions on the edited images, computing accuracy first within each sample and then averaging across the entire dataset. Figure 7 shows that reversing or shuffling the intermediate visual tokens reduces accuracy by 3.42 percentage points on the initial 64 inputs. This result suggests that the intermediate visual latents carry order-sensitive information about the progression of visual states, which the generator uses to guide physical editing on these inputs.

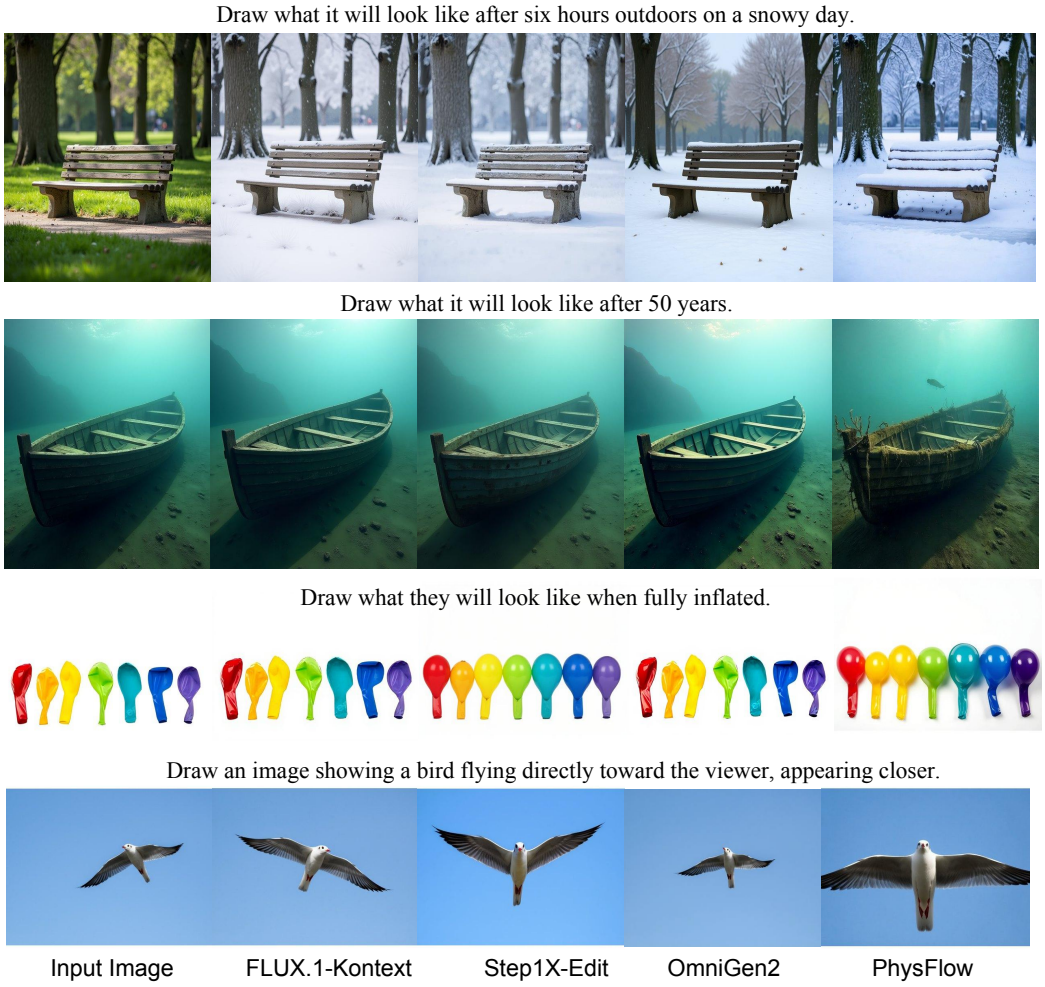



Figure 6: Additional qualitative comparisons on RISEBench (Zhao et al., 2025).

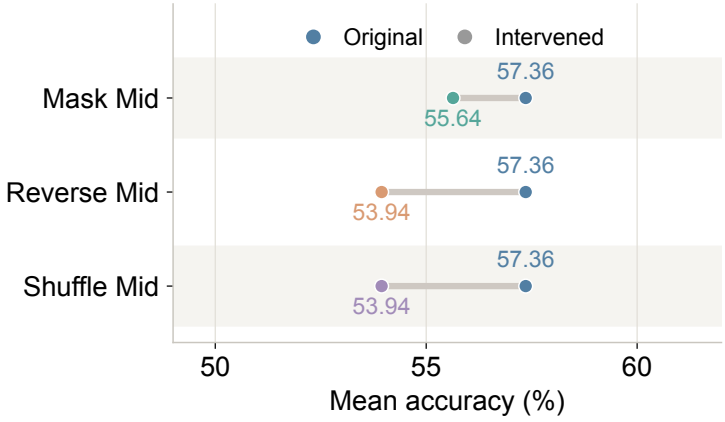



Figure 7: **Effect of intermediate-token interventions on physical editing.** Mean physical question-answering accuracy on 64 PICABench-Superficial inputs (Pu et al., 2026), comparing the original condition with masking, reversing, and shuffling intermediate visual tokens.