COMPFLOW: COMPOSING VELOCITY FIELDS FOR MULTI-CONDITIONAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating samples that satisfy many conditions simultaneously can be achieved with a surprisingly simple operation: composing conditional velocity fields at inference time. We introduce *CompFlow*, a flow-matching framework for compositional inference without retraining, architectural changes, or specialized samplers. We show that weighted velocity residuals yield the score of a *Product-of-Experts* composition for positive weights, while negative weights introduce quotient factors to suppress selected concepts. Beyond this score identity, we derive the exact *compositional defect*, characterizing when the composed dynamics transport the intended density path. Computable from residuals already evaluated during sampling, this quantity connects transport compatibility to measurable interference between conditional contributions. On CLEVR, a single-object-conditioned model is composed at inference to *simultaneously* control shape, color, and position for up to five objects, achieving 99.1–86.5% per-object accuracy with $30\times$ fewer network evaluations than the evaluated baselines. With pretrained FLUX.1[dev], CompFlow combines up to five conditions in high-resolution images and improves non-spatial compositional alignment over single-prompt conditioning on T2I-CompBench. The same operator also enables controlled subtraction of contexts, objects, attributes, and styles. Together, these results connect practical inference-time concept arithmetic with a principled characterization of conditional interactions.

1 INTRODUCTION

Conditional generative models can render individual concepts faithfully, yet remain unreliable when several constraints must hold simultaneously. Training or fine-tuning a separate model for every combination is neither scalable nor modular (Kaplan et al., 2020). A useful compositional approach should instead reuse single-condition knowledge at inference time: it should add desired concepts, suppress undesired ones, and reveal when the requested conditions interfere. We therefore ask whether the conditional and deterministic velocity fields of a flow-matching (Lipman et al., 2023) model admit such arithmetic, and when the resulting sampler remains faithful to a well-defined composed distribution.

Compositional generation has long been explored through Mixture-of-Experts architectures (Jacobs et al., 1991), Energy-Based Models (EBMs) (LeCun et al., 2006; Du & Mordatch, 2019; Du et al., 2021), and Product-of-Experts (PoE) formulations (Hinton, 1999; Mayraz & Hinton, 2000; Hinton, 2002), with the latter two supporting direct combinations of energies or scores. Diffusion-based methods (Liu et al., 2022; Du et al., 2023; Yu & Gao, 2025) extend this principle by composing conditional scores at inference time. Composable Diffusion (Liu et al., 2022) introduced residual-based conjunction and negation but left open which distribution the sampler reaches. Existing answers add correction machinery, such as MCMC (Du & Mordatch, 2019; Nijkamp et al., 2020; Du et al., 2023) and related sampling schemes (Neal, 1996; Geffner & Domke, 2021; Duane et al., 1987; Neal, 1997) or sequential Monte Carlo reweighting (Skreta et al., 2025), at the cost of additional hyperparameters and computation. We instead characterize the uncorrected deterministic sampler exactly. Related work unifies flow merging and reward adaptation through density operators (De Santi et al., 2026), links compositional generalization to linear, orthogonal representations and multiplicative concept binding (Uselis et al., 2026a;b), and allocates game-inspired responsibility among diffusion models during denoising (Gupta et al., 2026).

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: **Zero-shot compositional generation with CompFlow.** Adding and subtracting conditioned velocity fields combines desired concepts, suppressing unwanted ones. For example, the first image illustrates addition for four separate prompts: [a cute sleeping dragon + a pillow-book + a mystical forest + blue-ish dreamlike atmosphere]. The last image shows effective concept suppression: [A wonderful landscape − green], successfully removing the green color from the landscape. Prompts and details are available in Appendix C.1.

In this work, we introduce **CompFlow**, a training-free compositional generative framework that performs signed arithmetic on conditional velocity residuals. For Gaussian probability paths, the residual between a conditional and unconditional velocity field is proportional to the gradient of the corresponding noised conditional likelihood. Consequently, a real-weighted sum of residuals defines a single compositional rule: positive weights multiply likelihood factors and provide an AND-like PoE composition, whereas negative weights divide by selected factors and suppress their associated concepts. Both operations use the same pretrained model and ODE sampler.

While the composed field has exactly the score associated with a product or quotient of likelihood factors at each time, those densities need not themselves form the probability path transported by the composed ODE. We therefore derive the exact continuity-equation defect of signed velocity composition. The defect characterizes when the simple sampler is exact, recovers the known inaccuracy of classifier-free guidance away from unit weight, and isolates the interactions between conditional experts. Because it depends only on residuals already evaluated during sampling, it also yields a plug-in diagnostic requiring no additional network evaluations.

Our contributions are:

1. **Signed velocity composition.** Starting from velocity-score identity for Gaussian paths, we show that a single weighted residual rule yields the score of a normalized product/quotient of conditional likelihood factors. Positive and negative weights support concept conjunction and subtraction, respectively.
2. **Transport characterization and diagnosis.** We derive the exact *continuity defect* for arbitrary signed weights, identifying conditions under which it vanishes, and obtaining an inference-time diagnostic from quantities already computed by the sampler.
3. **Efficient and effective concept control.** We demonstrate CompFlow strong conjunction results on CLEVR and pretrained text-to-image generation, together with systematic concept subtraction on `FLUX.1[dev]`, without retraining, architectural changes, rejection sampling, or a specialized integrator. On CLEVR, our diagnostic reveals spatial interference between conditional velocity residuals that leads to object interaction failure modes.

2 Additive Flow Composition

A conditional flow model may already represent the individual concepts of interest, even when their desired combination was absent during training. Composing them requires isolating how each condition alters the underlying generative dynamics. For Gaussian probability paths, this conditional contribution has a precise interpretation: the difference between conditional and unconditional velocities is affine to the condition's likelihood score. We develop **CompFlow** by isolating and composing these contributions along Gaussian probability paths. We then distinguish the target PoE density path from the evolution induced by the composed dynamics, quantify their mismatch through a continuity-equation defect, and identify when transport is exact.

Flow-Matching preliminaries. Flow Matching (Lipman et al., 2023) learns a time-dependent velocity field for the ODE $\dot{x}_t = v_t(x_t)$, initialized from $p_0 = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We use the Gaussian interpolant

$$x_t = t\, x_1 + \sigma_t x_0, \qquad \sigma_t = 1 - (1 - \sigma_{\min})t, \tag{1}$$

where $x_0 \sim p_0$ and $x_1 \sim p_{\text{data}}$ are independent, and $\sigma_{\min} \geq 0$. Its sample-level velocity is

$$u_t(x_0, x_1) = x_1 - (1 - \sigma_{\min})x_0. \tag{2}$$

The conditional flow-matching objective trains a neural predictor on this target; the trained field then averages over training pairs consistent with the current state. Under condition c, this field is $v_t^c(x) = \mathbb{E}[u_t(x_0, x_1) \mid x_t = x, c]$ and transports the conditional noised density $p_t(x \mid c)$. We will write v_t^0 for the unconditional field, which transports the prior density $p_t(x)$. A shared network with conditional inputs can approximate both fields, as in classifier-free guidance (Ho & Salimans, 2022; Zheng et al., 2023). When $\sigma_{\min} = 0$, the terminal distribution is p_{data}; otherwise, it retains Gaussian noise of standard deviation $\sigma_{\min}$.

We develop the composition rule for exact marginal fields, returning to their learned approximations at inference time. The identities hold for $t > 0$ with $\sigma_t > 0$; when $\sigma_{\min} = 0$, this restricts them to $(0, 1)$, with the zero-noise endpoint understood through limits.

Conditional residuals encode likelihood scores. Although flow-matching models predict velocities, the Gaussian interpolant makes their scores accessible through the same fields. Here, we recall the velocity–score identity of (Zheng et al., 2023), specialized to Equation (1).

Lemma 2.1 (Velocity–score identity)**.** *The conditional marginal velocity satisfies*

$$v_t^c(x) = \frac{x}{t} + \frac{\sigma_t}{t}\nabla_x \log p_t(x \mid c), \tag{3}$$

with the analogous identity for the unconditional field v_t^0.

For composition, the relevant quantity is the velocity residual induced by each concept c_i. Subtracting the unconditional field removes the shared drift x/t and, by Bayes' rule, isolates the likelihood score:

$$\begin{aligned} \delta_i(t, x) &:= v_t^{c_i}(x) - v_t^0(x) \\ &= \frac{\sigma_t}{t}\left[\nabla_x \log p_t(x \mid c_i) - \nabla_x \log p_t(x)\right] \\ &= \frac{\sigma_t}{t}\nabla_x \log p_t(c_i \mid x). \end{aligned} \tag{4}$$

Thus, δ_i captures the score change induced by conditioning on c_i relative to the unconditional distribution. It directly supplies the contribution needed for the score arithmetic used in compositional generation (Du et al., 2020), without requiring a separate likelihood model. The derivation is recalled in Appendix A.

Concept addition. When several conditions are specified, Equation (4) suggests combining their residuals around the same unconditional field. Summing residuals adds log-likelihood gradients, yielding the score of a product of likelihood experts and motivating the additive composition proposed below.

Proposition 2.2 (Concept addition and PoE scores)**.** *Let $p_t(x)$ and $p_t(x \mid c_i)$ be unconditional and conditional densities induced by Equation* (1)*, with marginal velocities v_t^0 and $v_t^{c_i}$ and residuals δ_i defined in Equation* (4)*. For fixed weights $\lambda_i \in \mathbb{R}$, assume the density and likelihood factors are positive and differentiable and that the following density has a finite, nonzero normalizing constant Z_t:*

$$\tilde{p}_t(x) = \frac{1}{Z_t}\, p_t(x) \prod_{i=1}^{m} p_t(c_i \mid x)^{\lambda_i}, \qquad Z_t = \int p_t(x) \prod_{i=1}^{m} p_t(c_i \mid x)^{\lambda_i}\, dx. \tag{5}$$

Then the composed velocity

$$v_t^{\text{comp}}(x) := v_t^0(x) + \sum_{i=1}^{m} \lambda_i \delta_i(t, x) \tag{6}$$

satisfies the pointwise identity

$$v_t^{\text{comp}}(x) = \frac{x}{t} + \frac{\sigma_t}{t}\nabla_x \log \tilde{p}_t(x). \tag{7}$$

Taking the log-gradient of Equation (5) and using Equation (4) proves the identity. Intuitively, each concept c_i contributes its residual δ_i, while λ_i controls its weight in the composed score. Subtracting the unconditional field from each expert avoids repeatedly counting the shared contribution; the residuals are then accumulated around a single unconditional field. With one concept, this recovers classifier-free guidance. With several, it specifies a PoE composition without a model trained on the joint condition.

Concept subtraction. A composition may also include concepts to suppress. The residual formulation handles them by reversing the corresponding likelihood-score contributions: a negative weight introduces an inverse likelihood factor into the PoE density. Thus, concept subtraction follows from the same composition rule.

Corollary 2.3 (Concept subtraction)**.** *Under the setup of Proposition 2.2, let $\mathcal{P}$ index desired conditions c_i and $\mathcal{N}$ index concepts $\bar{c}_j$ to be suppressed, with fixed weights $\lambda_i, \mu_j \geq 0$. Assuming positive, differentiable factors and $0 < Z_t^{\pm} < \infty$, define*

$$\tilde{p}_t^{\pm}(x) = \frac{1}{Z_t^{\pm}}\, p_t(x) \frac{\prod_{i\in\mathcal{P}} p_t(c_i \mid x)^{\lambda_i}}{\prod_{j\in\mathcal{N}} p_t(\bar{c}_j \mid x)^{\mu_j}}. \tag{8}$$

Writing $\bar{\delta}_j(t,x) := v_t^{\bar{c}_j}(x) - v_t^0(x)$ for the residual of each suppressed concept, the signed composition

$$v_t^{\pm}(x) := v_t^0(x) + \sum_{i\in\mathcal{P}} \lambda_i \delta_i(t,x) - \sum_{j\in\mathcal{N}} \mu_j \bar{\delta}_j(t,x) \tag{9}$$

satisfies

$$v_t^{\pm}(x) = \frac{x}{t} + \frac{\sigma_t}{t} \nabla_x \log \tilde{p}_t^{\pm}(x). \tag{10}$$

The corollary follows by assigning weight $-\mu_j$ to each suppressed concept in Proposition 2.2. For one desired and one suppressed concept with unit weights, the rule becomes $v_t^{\pm} = v_t^0 + \delta_1 - \bar{\delta}_1 = v_t^{c_1} - v_t^{\bar{c}_1} + v_t^0$. The density interpretation makes the effect explicit: subtraction downweights states where the suppressed concept has high likelihood, relative to the positive-only PoE. Intuitively, it expresses an inverse-likelihood preference, rather than conditioning on the logical complement of that concept. Further details are given in Appendix A.5.

The defect of naive composition. The preceding results identify the score encoded by the composed velocity at each time. Determining the distribution obtained by integrating it also requires tracking the evolution of the PoE density. Each expert follows the prescribed Gaussian interpolant, yet their normalized product need not remain compatible with the composed field. The mismatch may therefore persist despite exact component velocities and exact ODE integration. We characterize this intrinsic effect of composition through a scalar function of the conditional residuals δ_i defined in Equation (4).

Proposition 2.4 (Composition defect)**.** *Under the assumptions of Proposition 2.2, suppose the density and likelihood factors are C^2 in x and C^1 in t, and that p_{data} has bounded support. At times with $t > 0$ and $\sigma_t > 0$, the composition defect is characterized by*

$$D_t(x) := \frac{t^2}{\sigma_t^2}\left[\left\|\sum_{i=1}^m \lambda_i \delta_i(t,x)\right\|^2 - \sum_{i=1}^m \lambda_i \|\delta_i(t,x)\|^2\right]. \tag{11}$$

In particular, $\tilde{p}_t$ satisfies the continuity equation under v_t^{comp} if and only if D_t is spatially constant on its support at every time under consideration.

The criterion identifies spatial variation in D_t as the obstruction to transport of the intended density path. The explicit continuity residual and its derivation are deferred to Appendix A. To interpret the sources of this defect, we expand D_t, of the residuals:

$$D_t(x) = \frac{t^2}{\sigma_t^2}\left[\sum_i \lambda_i(\lambda_i - 1)\|\delta_i\|^2 + 2\sum_{i<j} \lambda_i \lambda_j \langle \delta_i, \delta_j \rangle\right]. \tag{12}$$

The first term depends on the individual guidance strengths, whereas the second couples different concepts through their velocity-residual inner products. At unit positive weights, only these pairwise

overlaps remain, making the geometry of the concept contributions particularly transparent. Signed weights affect both terms, so the same characterization covers subtraction as well as addition: each suppressed concept contributes its residual $\bar{\delta}_j$ with weight $-\mu_j$.

Corollary 2.5 (Exact cases)**.** *Under the assumptions of Proposition 2.4, the composed field satisfies the continuity equation for $\tilde{p}_t$ in each of the following cases: (i) $m = 1$ and $\lambda_1 = 1$; (ii) $\lambda_i = 1$ for all i and $\langle \delta_i(t,x), \delta_j(t,x) \rangle = 0$ for every $i \neq j$ and every x, t under consideration; (iii) $\lambda_i = 1$ for all i and $p_t(c_1, \ldots, c_m \mid x) = \prod_i p_t(c_i \mid x)$ for every x, t. In case (iii), $\tilde{p}_t = p_t(\cdot \mid c_1, \ldots, c_m)$ and the composed field equals the joint conditional marginal velocity. If the ODE induces a unique transport from the common initial density $\tilde{p}_0 = p_0$, its law is then $\tilde{p}_t$.*

Even a single expert can have a nonzero continuity defect:

$$D_t(x) = \frac{t^2}{\sigma_t^2} \lambda_1(\lambda_1 - 1) \|\delta_1(t,x)\|^2$$

varies spatially whenever $\lambda_1 \notin \{0, 1\}$ and the squared residual norm varies with x. The characterization thus includes the familiar limitation that classifier-free guidance does not generally transport the tempered density exactly (Bradley & Nakkiran, 2024; Skreta et al., 2025). For multiple concepts, it additionally exposes the contributions of their interactions. Because D_t is expressed entirely through velocity residuals, its spatial variation can be examined without additional model evaluations beyond those used to compute the composed field, providing a diagnostic of compatibility with the intended PoE path.

Inference. The residual formulation yields a sampling procedure for the pretrained model. We encode each desired or suppressed condition independently and replace the exact fields in Equations (6) and (9) with the learned network v_θ. A suppressed concept is described and encoded in the same way as a desired one; its role is determined by the negative sign of its weight. Starting from $x_0 \sim p_0$, we evaluate the unconditional and conditional velocities at the state and time requested by the ODE solver, subtract the shared unconditional field to obtain the residuals, and combine them with weights λ_i and $-\mu_j$. The solver then advances the sample using the resulting field until $t = 1$.

The procedure requires no additional training, correction, or rejection steps. Its PoE interpretation is subject to the composition defect characterized above, alongside approximation of the component fields and numerical integration error. If desired, the same residuals provide an estimate of D_t through Equation (11), with the exact-field quantity recovered when the learned velocities are exact.

For $m = |\mathcal{P}| + |\mathcal{N}|$ conditions, each composed-field evaluation requires $m + 1$ network evaluations: one unconditional evaluation, reused across all residuals, and one per condition. Hence, N Euler steps require $N(m + 1)$ network evaluations; for other solvers, the cost scales with the number of composed-field evaluations they request. The complete procedure is summarized in Algorithm 1.

Algorithm 1: CompFlow sampling by velocity addition

Require: Velocity network v_θ, encoder Enc;
conditions $\{c_i\}_{i=1}^m$, weights $\{\lambda_i\}_{i=1}^m$;
time grid $\{t_k\}_{k=0}^N$

$x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
$\boldsymbol{e}_i \leftarrow \mathrm{Enc}(c_i)$ for $i = 1, \ldots, m$
$\boldsymbol{e}_\varnothing \leftarrow$ null-condition embedding
for $k = 0, \ldots, N-1$ **do**
  $\boldsymbol{v}_\varnothing \leftarrow v_\theta(t_k, x_k, \boldsymbol{e}_\varnothing)$
  for $i = 1, \ldots, m$ **do**
    $\boldsymbol{v}_i \leftarrow v_\theta(t_k, x_k, \boldsymbol{e}_i)$
    $\delta_i \leftarrow \boldsymbol{v}_i - \boldsymbol{v}_\varnothing$
  end for
  $\boldsymbol{v}_{\mathrm{comp}} \leftarrow \boldsymbol{v}_\varnothing + \sum_{i=1}^m \lambda_i \delta_i$
  $x_{k+1} \leftarrow \text{ODE-STEP}(x_k, \boldsymbol{v}_{\mathrm{comp}}, t_k, t_{k+1})$
end for
return x_N

3 EXPERIMENTS

We evaluate whether composing conditional velocity residuals provides effective control over multiple concepts, first in a setting with explicit object attributes and then in open-domain text-to-image generation. CLEVR (Johnson et al., 2017) tests joint control of shape, color, and position for up to five objects; experiments with pretrained `FLUX.1[dev]` models (Labs et al., 2025; Labs, 2024) examine natural-language composition and concept subtraction. Architectures, hyperparameters, and additional experiments are provided in Appendices B and C.

3.1 CLEVR MULTI-OBJECT GENERATION

Setup. We train a single UNet-based flow-matching model on rendered CLEVR scenes, conditioning each example on the shape, color, and position of *one* object. At inference, CompFlow combines up to $m = 5$ such conditions through Algorithm 1. This setting tests whether control learned for individual objects extends to several independently specified objects without joint conditioning during training. Conditions are replaced by a learned null token with probability $p_{\text{drop}} = 0.15$, and sampling uses $N = 20$ Euler steps.

Evaluation protocol. Joint attribute control requires evaluating both object identity and placement. We therefore query GroundingDINO (Liu et al., 2024) with attribute descriptions such as *"a red cube"* and use Hungarian matching to associate detections with requested coordinates. A requested object is counted as correct when its matched, attribute-conditioned detection lies within $\tau = 0.1W$ of the target position, where W is the image width. This provides an automatic estimate of joint shape, color, and position accuracy. We also report mean pixel distance for matched detections and DINOv2 feature similarity to real CLEVR images as a proxy for visual realism. For comparison with position-based evaluations (Du et al., 2023; Yu & Gao, 2025), we additionally use SAM2 (Ravi et al., 2024) to assess whether requested positions are occupied, irrespective of object identity.

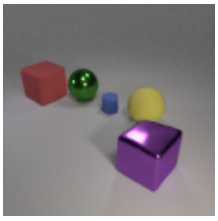

Figure 2: Five-object composition with joint shape, color, and position conditioning.

Main results. Table 2 shows that per-object accuracy on the joint shape–color–position task remains above 94% through $m = 4$ and reaches 86.5% at $m = 5$. DINOv2 similarity changes modestly, from 0.920 to 0.904, suggesting that the increased difficulty of satisfying all requested attributes is accompanied by a smaller change in this realism proxy. The five-object example in Figure 2 illustrates the level of control obtained from single-object conditioning. Sampling cost grows linearly as $N(m + 1)$, reaching 120 network evaluations at $m = 5$. For each number of conditions m, we select a shared per-condition weight $\lambda_i = w$ that maximizes per-object accuracy in the guidance sweep reported in Appendix B. Table 6 reports all metrics at the selected operating point for each m.

Under the position-only SAM2 protocol in Table 1, CompFlow exceeds all reported EBM baselines across the evaluated values of m, using at most 120 network evaluations compared with approximately $4{,}000$ for EBM (ULA) and EBM (U-HMC). CompLift and Cached CompLift achieve higher accuracy, but use a lift-score rejection criterion to select samples. CompFlow obtains its results along a single ODE trajectory with a fixed evaluation budget, offering a different accuracy–cost trade-off. Additional results appear in Appendix B, with an analysis of failure cases in Appendix B.2.

Table 1: Compositional generation on CLEVR, position-only task, verified via SAM2. Scene-level accuracy: fraction of samples in which all m requested objects are correctly placed. †Reproduced from Yu & Gao (2025). *CompLift applies rejection sampling: candidates failing the lift criterion are discarded and resampled; NFE counts only accepted samples and excludes rejected candidates. At m=5, CompFlow uses $N(m{+}1) = 120$ forward passes.

	Scene-level accuracy (%) ↑					
Method	m=1	m=2	m=3	m=4	m=5	NFE ↓
EBM (ULA)†	86.0	74.0	48.0	27.0	19.0	∼4000
EBM (U-HMC)†	82.0	66.0	34.0	24.0	11.0	∼4000
CompLift†*	100.0	100.0	99.9	97.5	90.3	rejection*
Cached CompLift†*	100.0	100.0	99.7	91.6	84.3	rejection*
CompFlow (ours)	**96.0±1.0**	**88.0±0.7**	**82.0**±1.1	**82.0±1.0**	**66.0±3.5**	≤ **120**

Compositional defect. By Proposition 2.4, transport along the PoE path fails whenever D_t varies across states. At unit weights, this variation arises solely from interactions among conditions. For two concepts, the self-terms vanish, yielding $D_t = 2(t/\sigma_t)^2\langle\delta_1, \delta_2\rangle$. Because $\langle\delta_1, \delta_2\rangle = \sum_p \delta_{1,p}\,\delta_{2,p}$ sums over pixels p, the pixelwise product $\delta_1 \odot \delta_2$ localizes where the two conditions interact in the image. Both quantities require only the residuals already evaluated by the sampler, substituting learned velocities for the exact fields.

Figure 3 visualizes this behavior graphically. We examine interaction maps along a composed trajectory at timestep $t = 0.5$. In this setup, the model is conditioned to generate

a green cylinder and a purple cube separated by either 11px or 70px. At the smaller separation, the residuals clearly overlap and interact together with opposite signs, as predicted by the corollary: the two conditions compete by pushing the trajectory in opposite directions. Conversely, when objects are well separated, the residuals concentrate around their respective objects, approximately satisfying the orthogonality condition in Corollary 2.5 (ii).

We next quantify this behavior beyond individual examples. For each condition set, we measure the variation of D_t during sampling as $\Delta = \int_0^1 \frac{\sigma_t}{t} \operatorname{Std}_k\left[D_t(X_t^{(k)})\right] dt$, which captures variation in defect interactions across the sampling trajectory. Across 768 two-object condition sets, Δ decreases from 110–127 at separations below 16 px, to values below 18 above 52 px. This trend helps explaining why larger CLEVR compositions are generally harder: as m increases, objects become more crowded, making close neighbors, and thus overlapping conditional contributions, more frequent. The defect therefore provides a zero-cost, sampler-intrinsic, diagnostic of where conditional contributions interfere.

Table 2: GroundingDINO scores of CompFlow on the full CLEVR compositional task.

	m (number of composed conditions)				
Metric	m=1	m=2	m=3	m=4	m=5
Position acc. (%) ↑	99.1	97.5	94.7	94.5	86.5
Mean dist. (px) ↓	4.80	5.32	6.57	6.18	9.07
DINOv2 sim. ↑	0.920	0.909	0.918	0.912	0.904
NFE	40	60	80	100	120

3.2 Text-to-Image Compositional Generation

So far, we described how an explicit-attributes model isolates composition residuals. Inspired by related work on compositional conditioning on diffusion models, we therefore ask next whether the same rule remains useful when concepts are specified in natural language and represented by a large pretrained model. For this experiments, we evaluate `FLUX.1[dev]` as a robust flow-matching-based, text-to-image generative baseline on T2I-CompBench (Huang et al., 2023b), combining quantitative evaluation with qualitative examples of concept addition and subtraction.

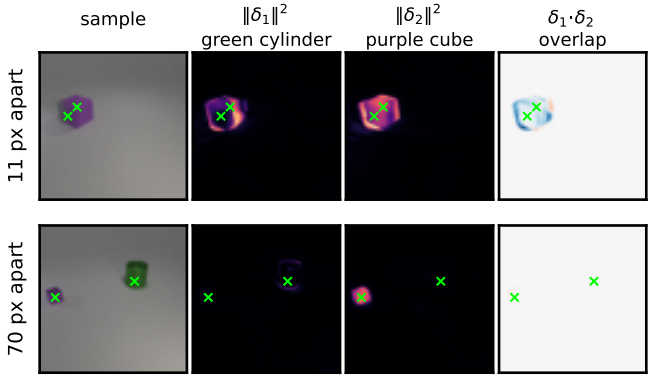

Figure 3: Residual interaction maps for a green cylinder and a purple cube separated by 11 px (top) and 70 px (bottom). Columns show the sample, squared residuals δ_1^2 and δ_2^2, and their product $\delta_1\delta_2$, whose pixelwise sum equals the cross term in Equation (12). Blue denotes opposing contributions, which concentre for nearby objects and nearly vanish for distant ones.

Setup. For each benchmark prompt c, an LLM extracts concept strings $\{c_i\}_{i=1}^m$, which are encoded independently by the `FLUX.1[dev]` text encoder. The decomposition instructions ask for a minimal set of conditions that preserves the original semantics; the full pipeline is given in Appendix C.3. We compose the resulting fields using Equation (6), with $\lambda_i = 1$ unless otherwise stated, and use $N = 28$ sampling steps. The single-prompt baseline conditions the same backbone on the complete prompt, using the same number of steps and backbone guidance setting. Each composed-field evaluation requires $m + 1$ network evaluations, so the comparison matches sampling steps rather than total computation. Neither method requires additional training or rejection sampling. Full settings and sampling times are reported in Appendices C.1 and C.3.

T2I-CompBench evaluation. Following T2I-CompBench++ (Huang et al., 2023a), we use ShareGPT4V-based Share-CoT scoring (Chen et al., 2023) as the primary metric for non-spatial relationships. CLIP (Radford et al., 2021; Hessel et al., 2021) and BLIP-CLIP (B-CLIP) (Chefer et al., 2023) provide complementary measures of image–text alignment.

All CompFlow images are evaluated against the original full prompt, so decomposition changes the conditioning procedure while preserving the evaluation target. We also include results reported in T2I-CompBench++ for comparison with other text-to-image models.

In Table 3, CompFlow achieves the highest Share-CoT score among the compared methods, increasing the `FLUX.1[dev]` single-prompt score from 0.7809 to 0.8021. Its CLIP score of 0.3201 is also competitive with reported compositional baselines, including Composable Diffusion (Liu et al., 2022). The Share-CoT improvement supports the use of separate concept conditions for satisfying the full prompt, with the gains obtained through additional inference-time evaluations of the same backbone.



Single prompt CompFlow Single prompt CompFlow

Figure 4: Single-prompt `FLUX.1[dev]` and CompFlow on dog–owner and woodcarver–sculpture compositions. Each pair compares full-prompt conditioning (left) with composition of separately conditioned velocity fields (right).

Qualitative analysis. Figure 4 illustrates two forms of non-spatial composition: *a dog interacting with its owner*, and *a woodcarver creating a bird sculpture*. In the latter example, separate conditions describe the artisan and the bird sculpture made from wood, and the CompFlow sample retains both as identifiable scene elements. These examples complement the aggregate scores by showing how separately specified concepts can coexist in the generated image.

Table 3: Compositional text-to-image evaluation using Share-CoT, CLIP, and B-CLIP metrics.

Model	Share-CoT	CLIP	B-CLIP
Stable v1-4 Rombach et al. (2021)	0.7487	0.3079	0.7565
Stable v2 Rombach et al. (2021)	0.7567	0.3127	0.7609
Composable v2 Liu et al. (2022)	0.6927	0.2980	0.7038
Structured v2 Feng et al. (2023)	0.7560	0.3111	0.7614
Attn-Exct v2 Chefer et al. (2023)	0.7593	0.3109	0.7607
GORS-unbiased Huang et al. (2023b)	0.7650	0.3158	**0.7641**
GORS Huang et al. (2023b)	0.7637	0.3193	0.7619
Stable XL Podell et al. (2024)	0.7673	0.3119	–
PixArt-α-ft Chen et al. (2024)	0.7747	0.3197	–
DALL·E 3 Betker et al. (2023)	0.7853	0.3003	–
Stable v3 Esser et al. (2024)	0.7782	0.3140	–
FLUX Labs et al. (2025)	0.7809	0.3127	0.7511
CompFlow (ours)	$\mathbf{0.8021 \pm 0.011}$	$\mathbf{0.3201 \pm 0.002}$	0.7613



a horse + grass field + pink sky + anime style + the big full moon

Figure 5: Progressive concept composition. Starting from *"a horse"*, each successive image adds the condition shown below it to the preceding set, combining subject, environment, color, and style.

Figure 5 further examines composition as the condition set grows: a horse is combined with a grass field, a pink sky, an anime style, and a full moon. Across this sequence, the subject remains recognizable while the scene incorporates the additional semantic and stylistic constraints. This illustrates the flexibility of the residual formulation across different concept types, without implying that their effects are independent. Further experiments on style mixing are reported in Appendix C.4.



Figure 6: **Style subtraction on FLUX.** Subtracting B from A with signed PoE ($\lambda = 1$) suppresses Van Gogh and watercolor styles while preserving the subject.

Concept subtraction. The same residual arithmetic can also remove a concept. Starting from a positive prompt A, we add the residual of a concept B with a negative weight $-\lambda$ (Equation (9)), keeping seeds and initial noise fixed so that any change in the sample is due to the subtraction alone. Figure 6 illustrates this on two styles. Asking for a portrait in Van Gogh style and subtracting the style itself yields a photographic portrait that keeps the original framing, while removing the watercolor medium from an illustrated castle produces a realistic castle with the same towers and silhouette. Generated on its own, B gives an unrelated image, a starry landscape or a flower, which shows that what reaches the sample is not the content of B but the direction its residual pushes the trajectory. Increasing λ strengthens this push and eventually alters lighting and composition as well. The complete sweeps in Appendix C.2 extend these observations to context, object, and attribute subtraction across ten concept pairs and five seeds, and include controls in which the subtracted concept is absent from A.

4 Conclusion

We introduced **CompFlow**, a training-free framework for compositional generation through signed arithmetic on conditional velocity residuals. The same rule supports adding desired concepts and suppressing unwanted ones, using a pretrained model and a standard ODE sampler without architectural changes. For Gaussian probability paths, we showed that this composition yields the score of a product or quotient of likelihood factors. We then characterized the continuity defect of the composed field, identifying when this exact score identity also leads to exact transport of the intended density path. Experiments demonstrate complementary forms of compositional control. On CLEVR, CompFlow jointly controls shape, color, and position for up to five objects, using over $30\times$ fewer network evaluations than the evaluated EBM baselines. With `FLUX.1[dev]`, composing separately encoded conditions improves compositional alignment over single-prompt conditioning on T2I-CompBench, at the cost of additional model evaluations. Signed composition also enables suppression of contexts, objects, attributes, and styles, with a tunable strength and no retraining. The subtraction experiments illustrate both effective concept removal and the accompanying changes to other scene properties. Beyond these generation results, our defect characterization gives the theoretical defect measurable spatial interpretation. This connection between transport analysis and observable model behavior motivates adaptive corrections where conditional contributions overlap. Together, these results establish CompFlow as a practical framework for compositional, training-free model arithmetic at inference time, connecting flexible generative control with a principled understanding of conditional interactions.

AI use statement

In this work, we used generative AI tools to aid and polish the writing (including improving clarity, grammar, phrasing, and overall presentation), and to support routine coding, debugging and experiment scheduling. We used AI and LLMs as part of our methodology as detailed in the main text and appendices. We did not use generative AI tools to generate research ideas or make scientific claims. All AI-assisted text was carefully reviewed, revised, and verified by the authors to ensure accuracy and consistency with the underlying research. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

Ethics statement

This work does not involve human subjects, personally identifiable information, or the collection of sensitive data. The datasets used in our experiments are publicly available research benchmarks and are used in accordance with their respective licenses and intended research purposes. We are not aware of any significant ethical concerns arising from the proposed methodology or its experimental evaluation. As with other machine learning methods, downstream applications should nevertheless consider potential issues related to bias, fairness, privacy, and misuse. We take responsibility for ensuring that the research is conducted and reported in accordance with established standards of research integrity.

Reproducibility statement

The main paper provides a complete description of the proposed method and experimental setting, while additional architectural details, implementation choices, hyperparameters, dataset information, and experimental configurations are reported in the appendix. We also provide the details required to reproduce the reported baselines, ablations, and evaluation procedures. The source code and configuration files used to produce the experimental results will be publicly released upon acceptance.

References

J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, and Y. Guo et al. Improving image generation with better captions. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Arwen Bradley and Preetum Nakkiran. Classifier-free guidance is a predictor-corrector, 2024. URL https://arxiv.org/abs/2408.09000.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models, 2023. URL https://arxiv.org/abs/2301.13826.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T. 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 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=eAKmQPe3m1.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023. URL https://arxiv.org/abs/2311.12793.

Riccardo De Santi, Malte Franke, Ya-Ping Hsieh, and Andreas Krause. A unified density operator view of flow control and merging, 2026. URL https://arxiv.org/abs/2602.08012.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3603–3613, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/378a063b8fdb1db941e34f4bde584c7d-Abstract.html.

Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/49856ed476ad01fcff881d57e161d73f-Abstract.html.

Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models, 2021. URL https://arxiv.org/abs/2012.01316.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 2023. URL https://proceedings.mlr.press/v202/du23a.html.

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(87)91197-X. URL https://www.sciencedirect.com/science/article/pii/037026938791197X.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017. URL https://arxiv.org/abs/1702.03118.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun R. Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=PUIqjT4rzq7.

Tomas Geffner and Justin Domke. MCMC variational inference via uncorrected hamiltonian annealing. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 639–651, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/05f971b5ec196b8c65b75d2ef8267331-Abstract.html.

Abhi Gupta, Polina Barabanshchikova, Vikas Garg, Samuel Kaski, and Tommi Jaakkola. Divide-and-denoise: A game-theoretic method for fairly composing diffusion models, 2026. URL https://arxiv.org/abs/2606.14756.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595.

Geoffrey E. Hinton. Products of experts, 1999. URL https://api.semanticscholar.org/CorpusID:15059668.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, 2002. ISSN 0899-7667. doi: 10.1162/089976602760128018. URL https://doi.org/10.1162/089976602760128018.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation, 2023a. URL https://arxiv.org/abs/2307.06350.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/f8ad010cdd9143dbb0e9308c093aff24-Abstract-Datasets_and_Benchmarks.html.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3:79–87, 1991. URL https://api.semanticscholar.org/CorpusID:572361.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2017. doi: 10.1109/CVPR.2017.215.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

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

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

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A tutorial on energy-based learning. In Gökhan Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander Smola, and Ben Taskar (eds.), *Predicting Structured Data*. MIT Press, Cambridge, MA, 2006. URL http://yann.lecun.com. v1.0, August 19, 2006.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=PqvMRDCJT9t.

Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models, 2022. URL https://arxiv.org/abs/2206.01714.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chuang Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, 2024.

Guy Mayraz and Geoffrey E. Hinton. Recognizing hand-written digits using hierarchical products of experts. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp (eds.), *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pp. 953–959. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/2000/hash/1f1baa5b8edac74eb4eaa329f14a0361-Abstract.html.

Radford M. Neal. *Monte Carlo Implementation*, pp. 55–98. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_3. URL https://doi.org/10.1007/978-1-4612-0745-0_3.

Radford M. Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification, 1997. URL https://arxiv.org/abs/physics/9701026.

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5272–5280. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5973.

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 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=di52zR8xgf.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

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

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. URL https://arxiv.org/abs/2112.10752.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts, 2025. URL https://arxiv.org/abs/2503.02819.

Arnas Uselis, Andrea Dittadi, and Seong Joon Oh. Compositional generalization requires linear, orthogonal representations in vision embedding models, 2026a. URL https://arxiv.org/abs/2602.24264.

Arnas Uselis, Darina Koishigarina, and Seong Joon Oh. How can embedding models bind concepts?, 2026b. URL https://arxiv.org/abs/2605.31503.

Chenning Yu and Sicun Gao. Improving compositional generation with diffusion models using lift scores, 2025. URL https://arxiv.org/abs/2505.13740.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided flows for generative modeling and decision making, 2023. URL https://arxiv.org/abs/2311.13443.

A PROOFS AND DERIVATIONS

The velocity–score identity for Gaussian probability paths was established in full generality by Zheng et al. (2023) and is restated in Lemma 2.1. We do not reprove that result here. Instead, we specialize it to the Gaussian interpolant used in this work, derive the conditional residual identity, and use these two ingredients to prove Proposition 2.2 (the pointwise Product-of-Experts score identity), Proposition 2.4 (the continuity defect of the composed field), and Corollary 2.5 (the cases in which the defect vanishes).

A.1 SPECIALIZATION TO THE GAUSSIAN INTERPOLANT

For the interpolant in Equation (1), the schedule is

$$\alpha_t = t, \qquad \sigma_t = 1 - (1 - \sigma_{\min})\, t, \qquad \dot{\alpha}_t = 1, \qquad \dot{\sigma}_t = -(1 - \sigma_{\min}).$$

Step 1: compute a_t. Directly from the definition,

$$a_t \;=\; \frac{\dot{\alpha}_t}{\alpha_t} \;=\; \frac{1}{t}.$$

Step 2: compute b_t. First note that $t(1 - \sigma_{\min}) = 1 - \sigma_t$ by the definition of σ_t. Then

$$\begin{aligned} b_t &= \frac{\big(\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t\big)\sigma_t}{\alpha_t} \\ &= \frac{\big(\sigma_t + t(1 - \sigma_{\min})\big)\sigma_t}{t} \\ &= \frac{\big(\sigma_t + 1 - \sigma_t\big)\sigma_t}{t} \;=\; \frac{\sigma_t}{t}. \end{aligned}$$

Step 3: assemble the velocity–score identity. Substituting $a_t = 1/t$ and $b_t = \sigma_t/t$ into $v_t^c(x) = a_t x + b_t\, s_t^c(x)$ yields

$$v_t^c(x) \;=\; \underbrace{\tfrac{1}{t}\, x}_{\text{drift}} \;+\; \tfrac{\sigma_t}{t} \underbrace{\nabla_x \log p_t(x \mid c)}_{\text{conditional score}}. \tag{13}$$

Sampling is initialized from $x_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, so the $1/t$ singularity at $t = 0$ is never evaluated.

A.2 CONDITIONAL VELOCITY RESIDUALS ARE LIKELIHOOD GRADIENTS

We compute the residual between the conditional and unconditional velocities. Recall

$$s_t^c(x) = \nabla_x \log p_t(x \mid c), \qquad s_t^0(x) = \nabla_x \log p_t(x).$$

Step 1: drift cancels in the residual. Both v_t^c and v_t^0 share the drift $\frac{1}{t}x$, which vanishes upon subtraction:

$$v_t^c(x) - v_t^0(x) \;=\; \tfrac{\sigma_t}{t}\big(\nabla_x \log p_t(x \mid c) - \nabla_x \log p_t(x)\big) \;=\; \tfrac{\sigma_t}{t}\, \nabla_x \log \frac{p_t(x \mid c)}{p_t(x)}.$$

Step 2: apply Bayes' rule. Substituting $p_t(x \mid c) = p_t(c \mid x)\, p_t(x) \,/\, p_t(c)$,

$$\nabla_x \log \frac{p_t(x \mid c)}{p_t(x)} \;=\; \nabla_x \log \frac{p_t(c \mid x)}{p_t(c)} \;=\; \nabla_x \log p_t(c \mid x),$$

where the last equality uses $\nabla_x \log p_t(c) = 0$.

Conclusion.

$$\boxed{v_t^c(x) - v_t^0(x) \;=\; \tfrac{\sigma_t}{t}\, \nabla_x \log p_t(c \mid x)} \tag{14}$$

The conditional velocity residual is exactly the (rescaled) gradient of the noised log-likelihood of the condition. □

A.3 Proof of Proposition 2.2: velocity addition gives a Product of Experts

Let $c_1, \dots, c_m$ be conditions to compose, with weights $\lambda_1, \dots, \lambda_m \in \mathbb{R}$. The composed velocity field is defined as the unconditional velocity plus a weighted sum of conditional residuals,

$$v_t^{\text{comp}}(x) \;:=\; v_t^0(x) \;+\; \sum_{i=1}^{m} \lambda_i \big[v_t^{c_i}(x) - v_t^0(x) \big], \tag{15}$$

and we define the tempered Product-of-Experts density

$$\tilde{p}_t(x) \;\propto\; p_t(x) \prod_{i=1}^{m} p_t(c_i \mid x)^{\lambda_i}. \tag{16}$$

Step 1: rewrite each residual via the score identity. Substituting Equation (14),

$$v_t^{\text{comp}}(x) \;=\; v_t^0(x) \;+\; \frac{\sigma_t}{t} \sum_{i=1}^{m} \lambda_i \, \nabla_x \log p_t(c_i \mid x).$$

Step 2: expand the unconditional velocity. Using Equation (13) with no conditioning, $v_t^0(x) = \frac{1}{t} x + \frac{\sigma_t}{t} \nabla_x \log p_t(x)$, so

$$v_t^{\text{comp}}(x) \;=\; \tfrac{1}{t}\, x \;+\; \tfrac{\sigma_t}{t} \left[\nabla_x \log p_t(x) \;+\; \sum_{i=1}^{m} \lambda_i \nabla_x \log p_t(c_i \mid x) \right].$$

Step 3: collect into a single log-gradient. The bracketed expression is the gradient of a sum of logs, hence the log of a product:

$$\nabla_x \log p_t(x) + \sum_{i=1}^{m} \lambda_i \nabla_x \log p_t(c_i \mid x) \;=\; \nabla_x \log \left[p_t(x) \prod_{i=1}^{m} p_t(c_i \mid x)^{\lambda_i} \right] \;=\; \nabla_x \log \tilde{p}_t(x),$$

where the last equality uses Equation (16) and the fact that the proportionality constant in $\tilde{p}_t$ is independent of x.

Conclusion.

$$\boxed{v_t^{\text{comp}}(x) \;=\; \tfrac{1}{t}\, x \;+\; \tfrac{\sigma_t}{t}\, \nabla_x \log \tilde{p}_t(x)} \tag{17}$$

The linearly composed field therefore obeys the pointwise score identity for the normalized tempered Product-of-Experts density $\tilde{p}_t$, whenever this density is well-defined. Whether v_t^{comp} actually transports $\tilde{p}_t$ is settled by Proposition 2.4, proved next. □

A.4 Proof of Proposition 2.4: continuity defect of the composed field

Setting. Let us begin defining the following notation: $s := \nabla \log p_t$ (the score), $r_i := \nabla \log p_t(c_i \mid x)$ (the conditional score), $\Lambda := \sum_i \lambda_i$ (the sum of the experts weights), $\rho_t := p_t \prod_i p_t(c_i \mid x)^{\lambda_i}$, $Z_t := \int \rho_t$, $\tilde{p}_t := \rho_t / Z_t$. By Bayes' rule,

$$\log \rho_t = (1 - \Lambda) \log p_t + \sum_i \lambda_i \log p_t(x \mid c_i) + \text{const}, \qquad \nabla \log \rho_t = s + \sum_i \lambda_i r_i. \tag{18}$$

Lemma A.1. *For any positive density q and $v_q := \frac{x}{t} + \frac{\sigma_t}{t} \nabla \log q$,*

$$\partial_t q + \nabla \cdot (q\, v_q) = q \Big[L[\log q] + \frac{\sigma_t}{t} \|\nabla \log q\|^2 \Big], \qquad L[f] := \partial_t f + \frac{1}{t} \big(d + x \cdot \nabla f \big) + \frac{\sigma_t}{t} \nabla^2 f. \tag{19}$$

Proof. Since $\nabla q = q \nabla \log q$ and $\partial_t q = q\, \partial_t \log q$,

$$\nabla \cdot (x\, q) = d\, q + x \cdot \nabla q = q\, (d + x \cdot \nabla \log q),$$
$$\nabla \cdot (q \nabla \log q) = \nabla \cdot \nabla q = \nabla^2 q = \sum_j \partial_j \big(q\, \partial_j \log q \big) = q \big(\|\nabla \log q\|^2 + \nabla^2 \log q \big).$$

Hence $\partial_t q + \nabla \cdot (q\, v_q) = q \big[\partial_t \log q + \frac{1}{t} (d + x \cdot \nabla \log q) + \frac{\sigma_t}{t} \nabla^2 \log q + \frac{\sigma_t}{t} \|\nabla \log q\|^2 \big]$. □

L is affine: $L[\alpha f + \beta g] = \alpha L[f] + \beta L[g]$ whenever $\alpha + \beta = 1$.

Theorem A.2. *With* $D_t := \big\| \sum_i \lambda_i r_i \big\|^2 - \sum_i \lambda_i \|r_i\|^2$,

$$\partial_t \tilde{p}_t + \nabla \cdot (\tilde{p}_t v_t^{\mathrm{comp}}) = \frac{\sigma_t}{t}\, \tilde{p}_t \big(D_t - \mathbb{E}_{\tilde{p}_t}[D_t]\big). \tag{20}$$

Proof. Step 1. Gaussian paths are transported by their own score field (Lemma 2.1), so the bracket in (19) vanishes for $q = p_t$ and $q = p_t(\cdot \mid c_i)$:

$$L[\log p_t] = -\frac{\sigma_t}{t}\|s\|^2, \qquad L[\log p_t(\cdot \mid c_i)] = -\frac{\sigma_t}{t}\|s + r_i\|^2. \tag{21}$$

Step 2. Combine (21) with coefficients $(1 - \Lambda), \lambda_1, \ldots, \lambda_m$, which sum to one, and use (18) and affinity of L:

$$L[\log \rho_t] = -\frac{\sigma_t}{t}\Big[(1 - \Lambda)\|s\|^2 + \sum_i \lambda_i \|s + r_i\|^2\Big]. \tag{22}$$

Step 3. $v_t^{\mathrm{comp}} = v_{\rho_t}$ (Proposition 2.2), so (19) applied to ρ_t together with (22) gives

$$\partial_t \rho_t + \nabla \cdot (\rho_t v_t^{\mathrm{comp}}) = \frac{\sigma_t}{t}\rho_t \Big[\Big\| s + \sum_i \lambda_i r_i \Big\|^2 - (1-\Lambda)\|s\|^2 - \sum_i \lambda_i \|s + r_i\|^2\Big] = \frac{\sigma_t}{t}\rho_t D_t, \tag{23}$$

since expanding the squares cancels every term containing s.

So far, we built a statement about ρ_t, but we want to actually compute the defect on the whole normalized probability density $\tilde{p}_t = \rho_t / Z_t$, and its transported velocity fieldv_t^{comp}. In general, Z_t were constant in t, it would be possible to divide (23) by it. However, $Z_t = \int p_t \prod_i p_t(c_i \mid x)^{\lambda_i}\, dx$ changes with t because the likelihoods $p_t(c_i \mid x)$ change as noise is removed.

Step 4a: the normalizer. Integrate (23) over $x \in \mathbb{R}^d$:

$$\int \partial_t \rho_t \, dx + \int \nabla \cdot (\rho_t v_t^{\mathrm{comp}})\, dx = \frac{\sigma_t}{t} \int \rho_t D_t \, dx. \tag{24}$$

The second integral vanishes (Gaussian tails), while the first integral is $\frac{d}{dt}\int \rho_t \, dx = \dot{Z}_t$ (differentiation under the integral). On the right side, $\rho_t = Z_t \tilde{p}_t$, so $\int \rho_t D_t \, dx = Z_t \, \mathbb{E}_{\tilde{p}_t}[D_t]$. Therefore

$$\dot{Z}_t = \frac{\sigma_t}{t}\, Z_t \, \mathbb{E}_{\tilde{p}_t}[D_t]. \tag{25}$$

Step 4b: quotient rule. Since $\tilde{p}_t = \rho_t / Z_t$ and Z_t does not depend on x,

$$\partial_t \tilde{p}_t = \frac{\partial_t \rho_t}{Z_t} - \frac{\rho_t \, \dot{Z}_t}{Z_t^2}, \qquad \nabla \cdot (\tilde{p}_t v_t^{\mathrm{comp}}) = \frac{\nabla \cdot (\rho_t v_t^{\mathrm{comp}})}{Z_t}. \tag{26}$$

Adding the two and using $\rho_t / Z_t = \tilde{p}_t$,

$$\partial_t \tilde{p}_t + \nabla \cdot (\tilde{p}_t v_t^{\mathrm{comp}}) = \frac{\partial_t \rho_t + \nabla \cdot (\rho_t v_t^{\mathrm{comp}})}{Z_t} - \tilde{p}_t \, \frac{\dot{Z}_t}{Z_t}. \tag{27}$$

By inserting (23) in the first term and (25) in the second, we obtain:

$$\partial_t \tilde{p}_t + \nabla \cdot (\tilde{p}_t v_t^{\mathrm{comp}}) = \frac{\sigma_t}{t}\, \tilde{p}_t \, D_t - \frac{\sigma_t}{t}\, \tilde{p}_t \, \mathbb{E}_{\tilde{p}_t}[D_t] = \frac{\sigma_t}{t}\, \tilde{p}_t \big(D_t - \mathbb{E}_{\tilde{p}_t}[D_t]\big). \tag{28}$$

□

A.5 Derivation of concept subtraction

Fix $t \in (0, 1)$, where the Gaussian noise is nonzero. Assume that the unconditional density and all likelihood factors in Equation (8) are positive and differentiable in x, and that $0 < Z_t^{\pm} < \infty$. The same argument applies at $t = 1$ if these regularity conditions and the velocity–score identity extend to that endpoint.

Step 1: differentiate the signed PoE. Taking logarithms of Equation (8) gives

$$\log \tilde{p}_t^{\pm}(x) = \log p_t(x) - \log Z_t^{\pm} + \sum_{i \in \mathcal{P}} \lambda_i \log p_t(c_i \mid x) - \sum_{j \in \mathcal{N}} \mu_j \log p_t(\bar{c}_j \mid x). \tag{29}$$

Since $Z_t^{\pm}$ does not depend on x,

$$\nabla_x \log \tilde{p}_t^{\pm} = \nabla_x \log p_t + \sum_{i \in \mathcal{P}} \lambda_i \nabla_x \log p_t(c_i \mid x) - \sum_{j \in \mathcal{N}} \mu_j \nabla_x \log p_t(\bar{c}_j \mid x). \tag{30}$$

Step 2: convert likelihood gradients to velocity residuals. Multiply Equation (30) by σ_t/t and add x/t. The unconditional terms give v_t^0, while each likelihood gradient gives the corresponding residual by Equation (14). This yields Equation (9). No conditional-independence assumption is needed for this algebraic identity. Equivalently, the coefficients of the full conditional and unconditional fields are

$$v_t^{\pm} = \left(1 - \sum_{i \in \mathcal{P}} \lambda_i + \sum_{j \in \mathcal{N}} \mu_j\right) v_t^0 \quad + \sum_{i \in \mathcal{P}} \lambda_i v_t^{c_i} - \sum_{j \in \mathcal{N}} \mu_j v_t^{\bar{c}_j}. \tag{31}$$

Their sum is one, so the common Gaussian drift is retained exactly. □

Scope of the density interpretation. Bayes' rule also gives

$$\tilde{p}_t^{\pm}(x) \propto p_t(x)^{1 - \sum_i \lambda_i + \sum_j \mu_j} \frac{\prod_{i \in \mathcal{P}} p_t(x \mid c_i)^{\lambda_i}}{\prod_{j \in \mathcal{N}} p_t(x \mid \bar{c}_j)^{\mu_j}}. \tag{32}$$

The unconditional factor is therefore part of the density-ratio construction. Inverse likelihood weighting favors states with less evidence for a negative concept, but does not implement the probability of its complement. Moreover, a quotient of Gaussian-noised densities need not itself follow the Gaussian probability path of any fixed terminal distribution. The transport question is settled by Proposition 2.4. The signed sampler transports $\tilde{p}_t^{\pm}$ up to the defect Equation (20), with $D_t = \left\|\sum_{i \in \mathcal{P}} \lambda_i r_i - \sum_{j \in \mathcal{N}} \mu_j \bar{r}_j\right\|^2 - \sum_{i \in \mathcal{P}} \lambda_i \|r_i\|^2 + \sum_{j \in \mathcal{N}} \mu_j \|\bar{r}_j\|^2$, where $\bar{r}_j = \nabla_x \log p_t(\bar{c}_j \mid x)$.

B Additional details on CLEVR experiments

UNet backbone. The velocity network v_θ is a UNet operating at 128×128 RGB resolution with base channel width $C = 128$ and channel multipliers $(1, 2, 2, 3)$, giving feature widths $(128, 256, 256, 384)$ across the four spatial levels. Each level uses two residual blocks with GroupNorm (32 groups) and SiLU activations (Elfwing et al., 2017); downsampling halves the spatial resolution with a strided 3×3 convolution. Multi-head self-attention with four heads is applied only at the lowest resolution (16×16, downsampling factor 8), which we found sufficient for CLEVR and keeps per-step cost tractable under the $N(m{+}1)$ scaling of compositional sampling. Skip connections concatenate matching-resolution features across the encoder and decoder. Timestep $t \in (0, 1]$ is embedded by sinusoidal features of dimension C followed by a two-layer MLP to dimension $4C = 512$ and injected into every residual block via scale–shift modulation. The output head applies a zero-init 3×3 convolution, so the learned velocity field starts at zero and is updated only by the training signal. The backbone contains 54.7M parameters.

Condition encoder. Each object is described by an attribute vector $c \in \mathbb{R}^{d_c}$ stacking normalized spatial coordinates and the integer-encoded shape and color classes ($d_c = 4$ in our runs). A two-layer MLP with SiLU activation projects c to a $4C$-dimensional embedding. This embedding is added to the timestep embedding before being fed to each residual block, sharing the same scale–shift modulation path; no cross-attention is used. A single learned null token $e_\varnothing \in \mathbb{R}^{4C}$, initialized from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, replaces the embedding with probability $p_{\text{drop}} = 0.15$ during training and provides the unconditional velocity v_t^0 required by Algorithm 1 at inference.

Table 5: Compute configuration used for training and evaluation.

Component	Specification
Operating system	Ubuntu 24.04.3 LTS
Kernel	Linux 6.8.0-83-generic
CPU	2× AMD EPYC 9274F 24-Core Processor
CPU cores / threads	48 physical cores / 96 hardware threads
System memory	1.5 TiB RAM
GPU	4× NVIDIA L40S
GPU memory	48 GB per GPU, 192 GB total
Training precision	TF32 matmul precision

Training dataset. We train on CLEVR (Johnson et al., 2017) rendered scenes containing between one and five objects. We generated the images for training according to the original CLEVR reported on the CLEVR webpage and GitHub repository. The training corpus aggregates all available relation shards and totals $185{,}803$ scene images, each paired with the scene JSON that enumerates every object together with its world-frame coordinates, shape, color, and projected pixel coordinates. At training time we draw a single random object per scene as the conditioning target, yielding single-object supervision exclusively. Images are resized to 128×128 and normalized to $[-1, 1]$. We split the dataset $90{:}5{:}5$ (train/val/test) with a fixed seed.

Optimizer and schedule. We optimize with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$), weight decay 10^{-5}, and base learning rate 10^{-4}. The schedule is a linear warmup over the first $10{,}000$ steps followed by a linear decay to 10^{-8} over training. Gradients are clipped at a global norm of 1.0. The per-device batch size is 64 with gradient accumulation to an effective batch of 256. Training uses TF32 matmul precision. A dropout of 0.15 is applied inside residual blocks. Early stopping monitors the validation CFM loss with patience 30 epochs; the selected checkpoint reaches validation loss 0.0079 after 103 epochs. Each training epoch required ≈ 15 minutes of training on our machines. This corresponds to approximately 25.75 wall-clock hours, or 103 GPU-hours, for the selected training run. Our compute configuration is detailed in Table 5.

Table 4: Wall-clock sampling time vs. number of composed conditions m on CLEVR (128×128, Euler solver, 20 steps, single NVIDIA GPU). Mean $\pm$ std over 5 runs after 1 warmup pass; $m{=}1$ uses single-condition generation, $m{\geq}2$ uses velocity composition. Cost scales roughly linearly in m, with one model forward per condition plus one null-token pass per step.

m	Time (s)	Min (s)	Max (s)	ms/step
1	0.5402 ± 0.0276	0.4860	0.5639	24.01
2	0.9084 ± 0.0026	0.9040	0.9119	45.42
3	1.1540 ± 0.0114	1.1325	1.1636	57.70
4	1.4081 ± 0.0143	1.3798	1.4173	70.40
5	1.6568 ± 0.0097	1.6440	1.6724	82.84

ODE solver and sampling. At inference we integrate the composed velocity field from Equation (6) with a fixed-step Euler solver and $N = 20$ steps, which empirically matches adaptive higher-order solvers (Heun, Dopri5, Tsit5 as implemented by `torchode`) on sample quality at a fraction of the per-step cost. The initial condition is $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and composition uses per-condition weights $\lambda_i = w$ with w set by the guidance-weight ablation below. Under this configuration the NFE per sample is $N(m + 1)$, i.e. 40, 60, 80, 100, and 120 for $m = 1, \ldots, 5$ respectively. We additionally report sampling times in Table 4

B.1 Guidance-Weight Ablation For CLEVR experiments

Protocol. We sweep a single guidance weight w applied uniformly to every conditioning slot ($\lambda_i = w$ for all i), jointly over $m \in \{1, 2, 3, 4, 5\}$ and $w \in \{1, 2, 3, 4, 5, 10\}$. For every (m, w) we generate 200 scenes whose per-object conditions are sampled from real CLEVR ground truth (matching the m-object budget) and evaluate them under the GroundingDINO protocol of Section 3.1: per-object accuracy at threshold τ, mean matched-detection pixel distance, and DINOv2 top-1 similarity to a real-CLEVR reference pool. All entries are produced by the same checkpoint, differing only in w. We provide additional qualitative samples in Figure 10

Table 6: Guidance-weight sweep on CLEVR (GroundingDINO protocol, 200 scenes per cell). For each (m, w) we report per-object accuracy at threshold τ (higher is better) and mean matched-detection pixel distance (lower is better).

m	metric	w=1	w=2	w=3	w=4	w=5	w=10
1	acc @ τ	**0.991**	0.963	0.943	0.942	0.942	0.923
	dist (px)	**4.80**	6.71	5.51	5.19	4.94	5.87
2	acc @ τ	0.921	**0.975**	0.944	0.947	0.961	0.931
	dist (px)	7.39	**5.32**	6.09	6.05	6.15	6.66
3	acc @ τ	0.887	0.945	0.907	**0.947**	0.927	0.945
	dist (px)	7.89	6.86	6.77	**6.57**	6.78	6.67
4	acc @ τ	0.875	0.905	0.898	**0.945**	0.945	0.870
	dist (px)	9.00	8.11	7.62	**6.18**	6.81	8.57
5	acc @ τ	0.804	0.860	**0.865**	0.822	0.832	0.737
	dist (px)	11.66	9.49	**9.07**	9.42	10.59	14.21

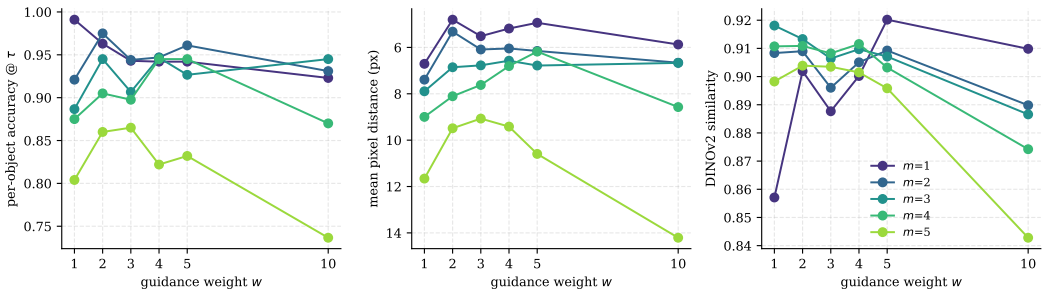

Figure 8: Guidance-weight ablation under the GroundingDINO protocol. One curve per m. *Left:* per-object accuracy at threshold τ. *Middle:* mean matched-detection pixel distance (the y-axis is inverted so that upward is better). *Right:* DINOv2 top-1 similarity to the real-CLEVR reference pool. The best operating point for each m shifts from low w at m=1 (saturation) to $w \in \{2, 3, 4\}$ as compositional load grows; very large w hurts consistently by $m \geq 4$. Data from Table 6;

Findings. Table 6 shows that the preferred guidance strength depends on the number of composed conditions. Accuracy is already near-perfect at $w = 1$ for $m = 1$, whereas larger compositions benefit from moderate guidance. Increasing w further does not consistently improve adherence: at $w = 10$, accuracy drops noticeably for $m = 4$ and $m = 5$, while remaining close to its peak for $m = 3$. The main-text results therefore report the best-performing weight separately for each m, with all metrics evaluated at the selected operating point. Figure 8 plots the full sweep, and Figure 9 summarizes the best-per-m results.

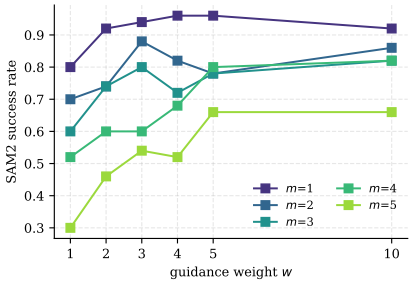

Figure 7: SAM2 position-only success rate as a function of guidance weight w, one curve per m. Same sweep grid as Table 6.

SAM2 cross-check. Under the SAM2 position-only protocol, the preferred guidance weight also depends on composition size: success peaks at $w = 4$ for $m = 1$ (96%) and at $w \in \{3, 10\}$ for larger m (Figure 7). Strong guidance remains more competitive under this metric, which checks whether requested positions are occupied without assessing object identity. The different operating points therefore reflect complementary evaluation criteria: positional occupancy for SAM2, and joint shape–color–position adherence for GroundingDINO.

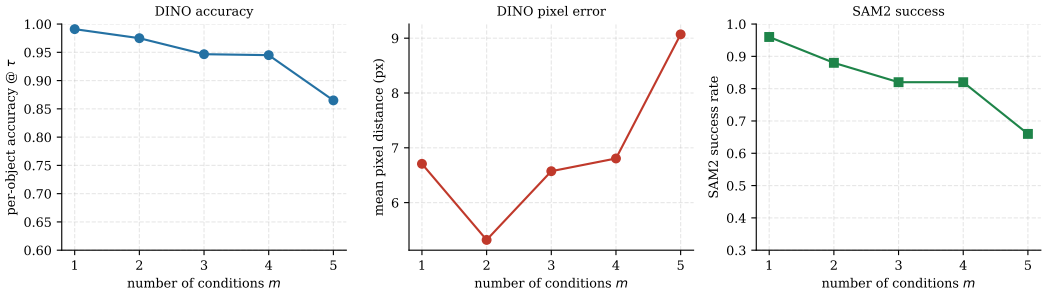

Figure 9: Best-per-m operating points: accuracy and pixel error under GroundingDINO, and success rate under SAM2, as a function of m. The w selected for each point is the per-m argmax of the corresponding metric from Table 6 and the SAM2 sweep.

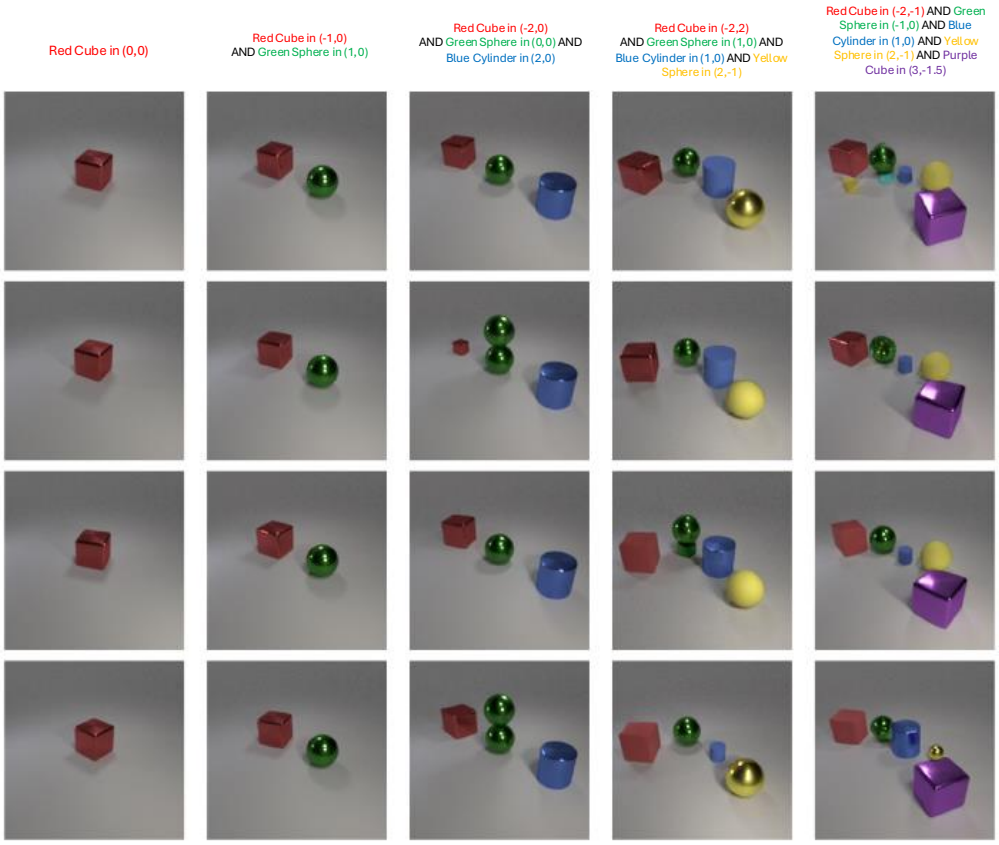

Figure 10: **Additional CLEVR samples** generated with CompFlow under increasing compositional load. Each column corresponds to a different set of composed object-level conditions, from a single red cube to five simultaneous shape–color–position constraints. Rows show independent samples generated from different random seeds for the same column conditions. Across seeds, the model consistently preserves the requested object attributes and approximate spatial layout, despite being trained only with single-object conditioning.

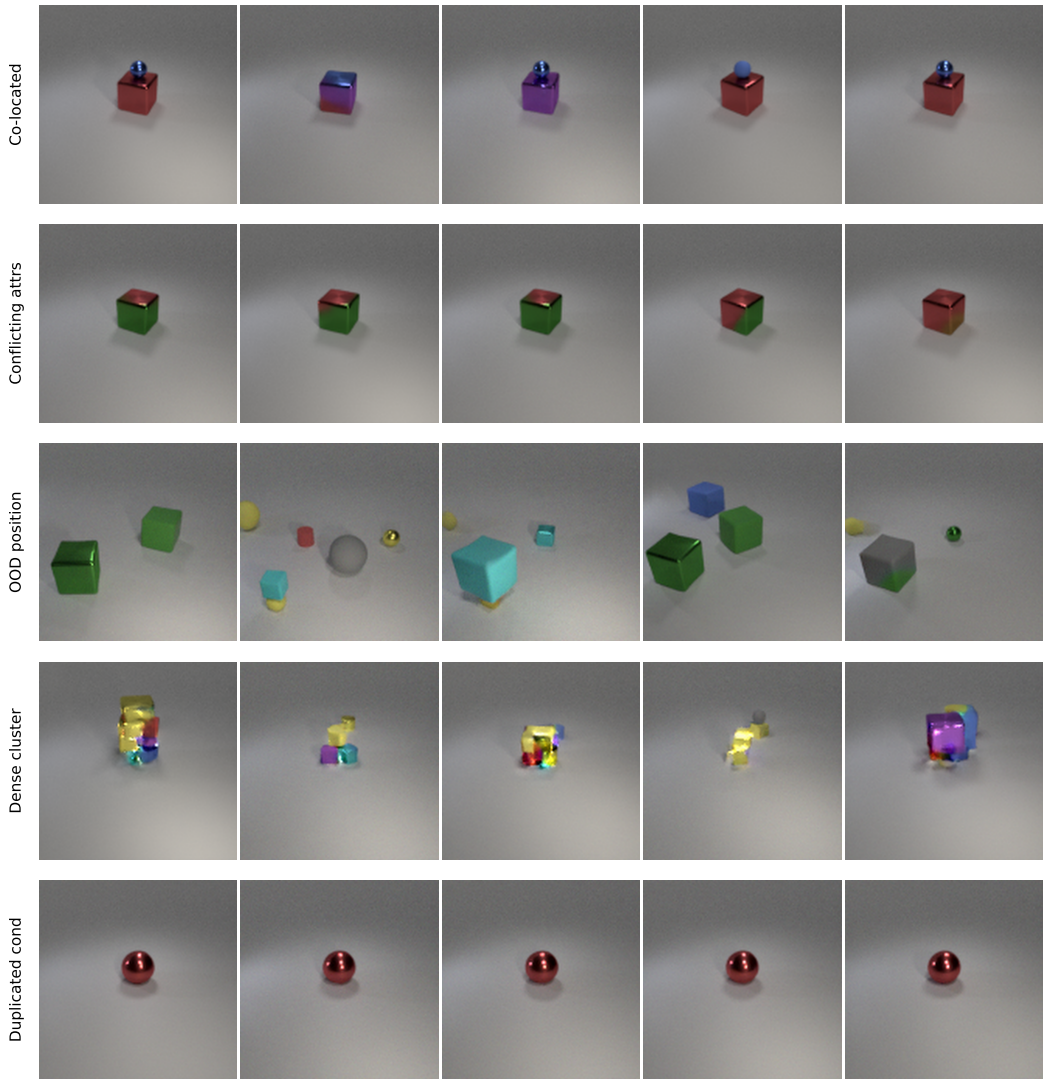

Figure 11: **CompFlow failure modes.** Five rows probe regimes the training distribution does not cover. Each row shows five independent samples at w=4, N=20, requested conditions are detailed in Table 7

B.2 Discussion of failure modes in CLEVR.

Figure 11 characterizes the regimes in which CompFlow's composition mechanism breaks down, complementing the in-distribution results of Section 3.1. Interestingly, three of the five regimes are informative about the method itself.

Co-located (row 1) is the most surprising result: when two objects are requested at the same (x, y), the model resolves the conflict geometrically by stacking or producing physically plausible configurations despite the absence of stacking in the single-object training distribution. Moreover, it is possible to observe collapsed modalities towards style merging. We attribute this to the composed velocity field locally satisfying both position constraints in the only way the rendering geometry permits and reveals interesting Out-of-Distribution (OOD) results.

Conflicting attributes (row 2) shows the predicted behavior of weighted velocity addition when two experts disagree on a single attribute at a single location: the composed score becomes the average of two incompatible velocity fields, and the resulting samples exhibit color-bleed artifacts rather than a clean choice between experts. In other words, asking the model to generate at the same time "a blue sphere" and "a red sphere" in the same position results in dynamics induced style mixing, producing a purple object. This is consistent with the Product-of-Experts interpretation of Proposition 2.2.

Table 7: Failure-mode scene specifications for CLEVR.

Scenario	**Objects (shape, color, position)**
Co-located	red cube @ $(0.0, 0.0)$ blue sphere @ $(0.0, 0.0)$
Conflicting attrs	red cube @ $(0.0, 0.0)$ green cube @ $(0.0, 0.0)$
OOD position	yellow sphere @ $(-5.0, -3.0)$
Dense cluster	red cube @ $(-0.3, -0.3)$ green sphere @ $(0.3, -0.3)$ blue cylinder @ $(0.3, 0.3)$ yellow cube @ $(-0.3, 0.3)$
Duplicated cond	red sphere @ $(0.0, 0.0)$ red sphere @ $(0.0, 0.0)$ red sphere @ $(0.0, 0.0)$

OOD position (row 3) probes coordinates far outside the training range, where the model falls back to in-distribution placements, often hallucinating extra objects.

Dense cluster (row 4) packs four objects into a region smaller than the typical inter-object spacing in CLEVR; here attribute identity degrades into an overlapping mass, though object count is approximately preserved. Both failures would likely be mitigated by training on a wider positional range and denser scenes, and neither contradicts the PoE analysis. Overall, the behavior of CompFlow under stress is consistent with its theoretical interpretation: it composes constraints faithfully where the backbone supports them, and degrades predictably where it does not.

Duplicated condition (row 5) provides a useful sanity check: repeating the same condition three times does *not* produce three objects but a single one, since the composed density $\tilde{p}_t(x) \propto p_t(x) \prod_i p_t(c|x)^{\lambda_i}$ increases around the same mode rather than instantiating multiple objects. This confirms that CompFlow composes *constraints on a sample*, not *independent objects in a scene*, and clarifies the semantics of multi-object generation: scene multiplicity must be encoded through distinct conditions, not through repetition.

B.3 Understanding the compositional defect on CLEVR

The cross term in Proposition 2.4 links compositional transport to interactions between conditional velocity contributions. Here we examine what these interactions look like in a trained flow matching model. On CLEVR, their spatial structure provides a naturally concrete interpretation; in fact, residuals associated with nearby objects can induce opposing changes in the same image region, whereas those associated with well-separated objects overlap much less. In this additional analysis, we will adopt the following approach: first, we will quantify this dependence on scene geometry, after that, we will examine how the defect changes with the number of concepts and develops along the sampling trajectory.

Measuring the defect from velocity residuals. The sampler already evaluates the unconditional and conditional velocities, giving the residuals $\delta_i = v_t^{c_i} - v_t^0$ at each step. Under the assumptions of

Proposition 2.4, these residuals determine the defect through their Gram matrix, $G_{ij} = \langle \delta_i, \delta_j \rangle$:

$$b_t D_t = \frac{t}{\sigma_t}\Big(\lambda^\top G \lambda - \sum_i \lambda_i G_{ii}\Big), \qquad b_t = \frac{\sigma_t}{t}. \tag{33}$$

The bracket separates into the self term $\sum_i \lambda_i(\lambda_i - 1)G_{ii}$ and the cross term $\sum_{i \neq j} \lambda_i \lambda_j G_{ij}$ of Equation (12). Substituting the learned velocities is straightforward and gives an empirical measure without additional network evaluations.

The continuity residual in Proposition 2.4 is $b_t(D_t - \mathbb{E}_{\tilde{p}_t}[D_t])$ in log-density units. Its defining feature is therefore the variation of D_t across states: a spatially constant field contributes with no residual. To capture this variation along the sampling trajectory, we decided to introduce two complementary quantities. The *defect magnitude*

$$\Delta = \int_0^1 b_t \, \mathrm{Std}[D_t(X_t)] \, dt \tag{34}$$

integrates the instantaneous spread across repeated samples of a fixed condition set. The *integrated defect*

$$L = \int_0^1 b_t \, D_t(X_t) \, dt \tag{35}$$

instead records the signed contribution along each trajectory, allowing contributions at different times to cancel. All empirical metrics are computed over trajectories of the composed sampler. In the exact cases of Corollary 2.5, $\Delta = 0$ and L is constant across trajectories of the same condition set, and their empirical variation will provide two ways to examine departures from these cases.

Evaluation protocol. We will use the same setup described in Section 3.1, with $N = 20$ Euler steps and unit weight on v_t^0. Each condition set is sampled $K = 8$ times from independent initial noise. We evaluate accuracy with the GroundingDINO protocol of Section 3.1, using the position tolerance $\tau = 0.1W$; Table 9 also considers $\tau/W \in \{0.10, 0.20\}$. We consider a scene correct only when **all** requested objects are correct. We specify that these diagnostic runs are separate from those in Table 2.

Spatial separation weakens residual interference. Two-object scenes at unit weights isolate the interaction between concepts: the self term vanishes, leaving

$$D_t = 2\frac{t^2}{\sigma_t^2}\langle \delta_1, \delta_2 \rangle. \tag{36}$$

We examine this interaction across six bins of requested object distance, with 128 condition sets per bin and eight samples per set: 768 scenes and 6,144 samples in total. The two requested objects always differ in shape or colour.

The defect decreases substantially as the objects separate (Table 8, Figure 13a). Between the 8–16 px and $\geq$52 px bins, Δ falls from 127.2 to 17.7, a 7.2$\times$ reduction with non-overlapping confidence intervals. Over the same range, the mean residual cosine changes from -0.31 to -0.08, indicating less opposition between the conditional contributions.

The maps in Figure 12 make this interaction visible. For nearby objects, the residuals overlap around the requested locations and predominantly oppose one another. With greater separation, the displayed overlap becomes much weaker. Since the pixelwise products sum to $\langle \delta_1, \delta_2 \rangle$, these maps give the theoretical cross term a direct spatial interpretation in the trained model. The reduction in overlap is consistent with the orthogonality condition in case (ii) of Corollary 2.5, although the measured defect remains nonzero even in the most distant bin.

Relation to compositional accuracy. The strong distance dependence of the defect is accompanied by a narrower range of scene accuracies, from 75% to 87% across bins. This distinction matters: the defect characterizes transport of the PoE path, whereas accuracy measures whether the requested objects appear. We therefore examine their association directly. Across condition sets, Δ has a modest positive correlation with the scene failure rate (Spearman $\rho = 0.12$, 95% CI $[0.04, 0.19]$). The association remains positive at all three detection tolerances and when correlations are computed within distance bins (Table 9). These results connect greater defect variation with more frequent compositional failures, while showing that the defect alone is not a strong predictor of task accuracy.

Table 8: Defect vs. requested distance (two objects, unit weights, 128 scenes per bin, 8 samples each). Means with 95% bootstrap CIs over scenes.

distance (px)	scenes	$\cos(\delta_1, \delta_2)$	defect Δ
< 8	128	-0.255 [-0.276, -0.232]	109.5 [100.9, 118.0]
8–16	128	-0.309 [-0.326, -0.290]	127.2 [118.5, 136.1]
16–24	128	-0.215 [-0.241, -0.187]	73.0 [63.6, 82.3]
24–36	128	-0.095 [-0.126, -0.065]	40.0 [31.2, 49.3]
36–52	128	-0.046 [-0.073, -0.021]	21.9 [15.5, 28.8]
≥ 52	128	-0.077 [-0.101, -0.053]	17.7 [13.1, 22.5]

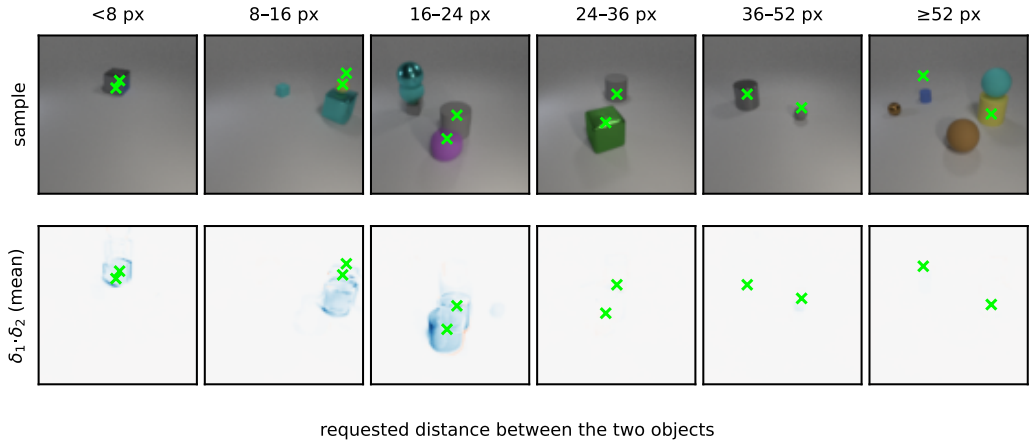

Figure 12: **Spatial structure of residual interactions on CLEVR.** Each column shows one sample from the median-overlap scene in a requested-distance bin (top) and its pixelwise residual overlap $\delta_1 \cdot \delta_2$ at $t = 0.5$, averaged over eight samples (bottom). Colours share the same scale; blue indicates opposing residuals. Crosses mark the requested positions. Nearby concepts exhibit concentrated negative overlap, which weakens with separation.

More concepts create more close neighbours. The spatial interpretation also suggests a mechanism contributing to the difficulty of larger compositions. Under the random-scene protocol of Table 2, the median pairwise distance remains near 34 px as m increases. The distance to each object's *nearest* neighbour, however, falls from 35.6 px at $m = 2$ to 16.6 px at $m = 5$ (Table 10, Figure 14b). Correspondingly, the fraction of objects with a neighbour within 16 px rises from 13% to 47%. Larger compositions therefore contain more objects in the distance range where the two-object experiment exhibits the strongest interference. This trend is consistent with crowding contributing to the decline in accuracy, alongside the growing number of pairwise interactions.

Integrated defect across composition sizes. We next examine how these interactions combine along complete sampling trajectories. Using the random-scene protocol, we consider $m \in \{2, 3, 4, 5\}$ concepts with equal weights $\lambda_i = w \in \{1, 2\}$. For each m, we sample 100 condition sets eight times each, using the same initial noise for both weights, and record L together with its self and cross contributions.

The within-set spread of L increases steadily with composition size (Table 11, Figure 13b). At $w = 1$, its median rises from 3.4 for two concepts to 125.5 for five. At $w = 2$, the median spread reaches 307.4 for five concepts. The decomposition clarifies the role of guidance: at unit weights, all variation comes from cross interactions, whereas at $w = 2$ the self contribution also varies substantially across trajectories. For five concepts, the median standard deviations of the self and cross contributions are 232.7 and 222.2, respectively. These spreads describe the two contributions separately; they do not add to the total spread because the contributions can be correlated.

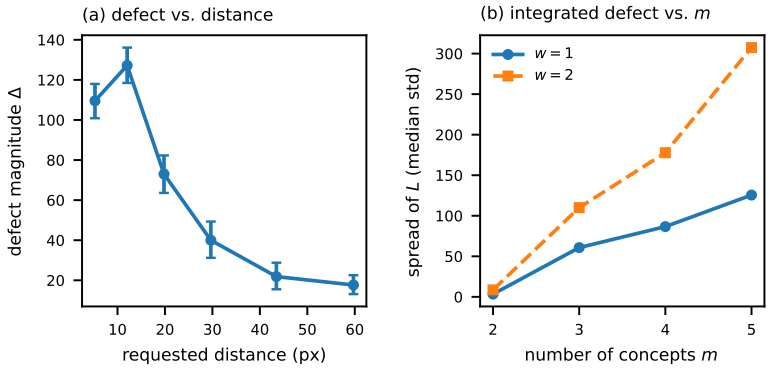

Figure 13: **(a)** Defect magnitude Δ vs. requested distance (two objects, unit weights; 95% CIs). **(b)** Within-set spread of the integrated defect L vs. the number of concepts (random-scene protocol).

Table 9: Association between the defect Δ and scene failure (768 scenes) at three detection thresholds. "Within bins" averages the Spearman correlation over the six distance bins.

τ/W	scene acc. (%)	Spearman(Δ, fail) [95% CI]	within bins
0.10	69.0	$+0.105$ [$+0.031$, $+0.179$]	$+0.091$
0.15	83.2	$+0.116$ [$+0.041$, $+0.190$]	$+0.119$
0.20	89.9	$+0.153$ [$+0.083$, $+0.228$]	$+0.135$

The integrated spread develops late in sampling. The temporal profiles in Figure 14c show a common pattern across composition sizes and both weights: the spread of the cumulative integral develops predominantly in the second half of sampling. This identifies late sampling as a relevant interval for examining the interactions responsible for the integrated defect. The observation is compatible with interference during refinement, although the temporal statistic alone does not establish which image features are being formed at each stage.

The experiments give the theoretical cross term a concrete interpretation in a trained flow matching model: it measures interactions between conditional velocity contributions whose overlap can be localized in the image. These interactions weaken with spatial separation, while larger compositions create more opportunities for close neighbours and exhibit greater variation in the integrated defect.

C Details on Text2Image generation task

C.1 Experimental setup

Backbone and pipeline. All text-to-image experiments use the publicly released `black-forest-labs/FLUX.1[dev]`. We rely on the `diffusers` `FluxPipeline` (≥ 0.30) and reuse all native components (text encoders, VAE, and the `FlowMatchEulerDiscreteScheduler`) without modification: there is no finetuning, no LoRA adaptation, and no change to either the transformer architecture or the scheduler. CompFlow is implemented purely as a sampling-time wrapper around the stock pipeline. Unless stated otherwise, all FLUX experiments use the configuration summarized in Table 12.

Velocity composition. At every ODE step the transformer is queried once per conditional subprompt c_i plus once on the unconditional prompt $c_0 = \varnothing$. The predicted velocities are combined according to Equation (6) which exactly recovers Algorithm 1 of the main paper for $\lambda_\varnothing = 1$. The composed velocity v_{comp} is then passed to the native scheduler step. We keep the FLUX guidance scalar fixed across all $m + 1$ forward passes within a step (see below), so that velocity composition acts strictly on top of the backbone's native prediction rather than re-entangling with classifier-free guidance.

Table 10: Crowding of requested objects under the random-scene protocol of Table 2 (500 scenes per m). Distances in pixels. The last two columns give the share of objects whose nearest requested neighbour is closer than 16 or 24 px.

m	median pairwise dist.	median nearest-neighbour dist.	< 16 px	< 24 px
2	35.6	35.6	13%	27%
3	33.3	22.5	30%	53%
4	33.8	18.6	40%	65%
5	33.3	16.6	47%	74%

Table 11: Integrated defect at unit weights ($w = 1$) under the random-scene protocol (100 condition sets per row, eight samples each). Spreads are medians over condition sets of the within-set standard deviations. At unit weights, the self term in Equation (12) vanishes and the integrated defect is entirely due to cross interactions.

m	std$[L]$	std$[L_{\text{self}}]$	std$[L_{\text{cross}}]$
2	3.4	0.0	3.4
3	60.7	0.0	60.7
4	86.6	0.0	86.6
5	125.5	0.0	125.5

C.2 Concept subtraction in FLUX

Setup. We evaluate the negative-concept extension of Equation (9) with separately encoded positive and negative text conditions. For each pair, A describes the desired scene and B names the concept to remove. The positive expert has unit weight and the negative expert has weight $-\lambda$, so that

$$v_t^{A-B}(x) = v_t^0(x) + \big[v_t^A(x) - v_t^0(x)\big] - \lambda\big[v_t^B(x) - v_t^0(x)\big], \tag{37}$$

where λ is the negative magnitude μ_1 of the method section. The weights are constant throughout sampling. All conditions are evaluated at the same state and time, and the signed field is passed unchanged to the native FLUX scheduler.

Figures 15 to 24 report the complete inventory: ten concept pairs, five seeds each, one grid per pair. Within each row all tiles share the seed and initial noise, so differences across columns are due only to the composition. Columns show the positive-only CompFlow sample A, the standalone generation of the negative concept B, and the four subtraction strengths. The standalone B column illustrates what the negative condition generates on its own; the subtraction uses its conditional velocity residual along the composed trajectory, not this image. The main-text figure selects one seed per category from this inventory.

Context subtraction. For *"a cabin in a forest" − "forest"* (Figure 15), the forest thins at $\lambda = 0.5$ and is gone at $\lambda = 1$ in all five seeds, while the cabin, with its lit windows, porch, and roofline, is re-situated on a city street, by a lake, or in a stone courtyard. For *"a dog on a beach" − "beach"*

Table 12: Default hyperparameters for CompFlow on `FLUX.1[dev]`. The "Source" column distinguishes parameters inherited from the stock `FLUX.1[dev]` pipeline from those introduced by CompFlow.

Component	**Value**	**Source**
Backbone	`black-forest-labs/FLUX.1[dev]`	FLUX
Scheduler	`FlowMatchEulerDiscreteScheduler`	FLUX
Resolution	1024×1024 (mult. of 16)	FLUX
ODE steps N	28	ours
FLUX-distilled guidance	1	ours
Numerical precision	`bfloat16`	ours
Unconditional prompt	empty string `""`	ours
$\lambda_\varnothing$ (uncond. weight)	1.0	ours

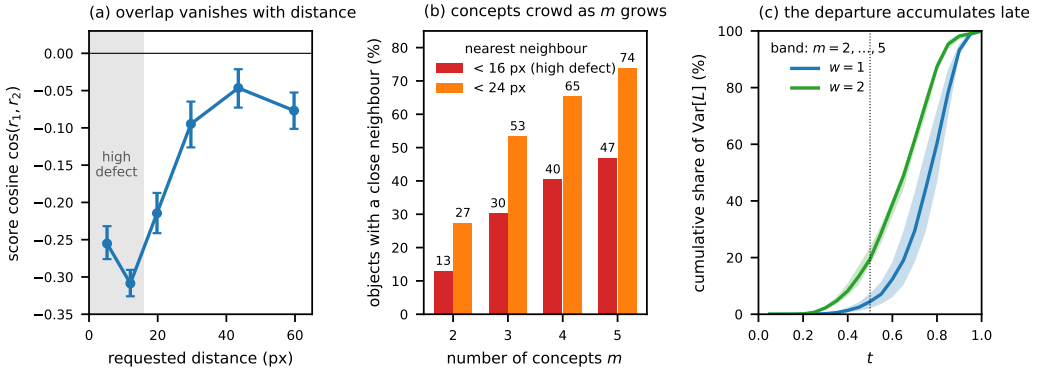

Figure 14: **Geometry and timing of compositional interactions.** **(a)** Residual cosine versus requested distance, with 95% confidence intervals. The shaded band marks distances below 16 px, where the two-object experiment shows the strongest interference. **(b)** Fraction of objects with a nearest neighbour within 16 px (red) or 24 px (orange) under the random-scene protocol. **(c)** Cumulative variance profile of the integrated defect (mean and range over $m = 2, \ldots, 5$), showing that its spread develops predominantly after $t = 0.5$.

(Figure 16), the dog, including breed, pose, and collar, is preserved in every composed sample up to $\lambda = 2$, while sand, surf, and sky give way to neutral, indoor, and sunset backgrounds. Context subtraction therefore rewrites the setting around the object named in A.

Object subtraction. For *"a table with a red apple" − "apple"* (Figure 17), the apple is absent at $\lambda = 1$ in all five seeds and the table, its cloth, and the room lighting remain; the freed spot is re-sampled with content consistent with the prompt, such as a cup, a glass, a slice of toast, or a person reading at the table. For *"a street with a bicycle" − "bicycle"* (Figure 18), the cyclist shrinks to a distant figure at $\lambda = 0.5$ and has left the street at $\lambda = 1$ in every seed, while the avenue, facades, parked cars, and lane markings of A persist; four seeds keep a coherent street up to $\lambda = 1.5$ and two up to $\lambda = 2$. Object removal is a re-composition of the scene rather than an empty region.

Attribute subtraction. For *"a snowy mountain landscape" − "snow"* (Figure 19), snow cover decreases monotonically with λ in all five seeds, exposing rock, conifers, lakes, and green valleys, and the mountains persist in all twenty composed tiles; beyond $\lambda = 1$ the scenes are relit at sunset. For *"a red sports car" − "red"* (Figure 20), the car changes color at $\lambda = 1$ in every seed and stays a sports car through $\lambda = 2$, while the background shifts to cyan, the complement of the subtracted hue. Both pairs show that a graded attribute is controlled continuously by λ, and that the negative expert moves the sample away from the subtracted concept in a predictable direction.

Style subtraction. For *"a castle in watercolor style" − "watercolor"* (Figure 21), all five illustrations become photographic castles on wooded hills at $\lambda = 1$, preserving towers, flags, and composition; four seeds keep the castle at $\lambda = 1.5$ and three at $\lambda = 2$, where the scene turns to dusk and moonlight. For *"a portrait in Van Gogh style" − "Van Gogh style"* (Figure 22), four of five seeds turn into photographic portraits at $\lambda = 1$, with brushwork, palette, and swirling background removed and the head-and-shoulders framing kept; three seeds remain portraits up to $\lambda = 2$. A medium and an artist style are thus removed while the depicted subject is retained.

Irrelevant-concept controls. In *"a cat indoors" − "ocean"* (Figure 23) and *"a red car" − "snow"* (Figure 24) the negative concept is absent from the positive prompt. All five seeds remain indoor cats through $\lambda = 1.5$ and cats at $\lambda = 2$, and every composed car sample up to $\lambda = 2$ is a red car; the subtraction changes pose, framing, vehicle model, and palette, which drifts toward warm tones as the sample moves away from the cold, white subtracted concept. The subject named by A is preserved when the subtracted concept has nothing to remove, which shows that the operator targets the named concept rather than perturbing the sample arbitrarily.

Operating range. Across all ten pairs and fifty sweeps the weight traces the same progression: attenuation of the target at $\lambda = 0.5$, removal with preservation of the complement at $\lambda = 1$ in nearly every seed, and, at larger weights, an increasing influence of the negative expert on the remaining scene. This progression follows directly from Equation (37). The negative expert contributes a negatively weighted conditional velocity residual to the entire composed dynamics, so at small λ it modulates the sample and at large λ it competes with the positive residual for the direction of the trajectory. The point at which this competition begins to alter the complement depends on the concept pair and on the seed, and lies at or above $\lambda = 1$ for every sweep in the inventory; for the beach, snow, red, and both control pairs it lies beyond $\lambda = 2$. Within this range the signed PoE removes contexts, objects, attributes, media, and styles with a single scalar per concept and no change to the model or the sampler.

C.3 Prompt decomposition and why it is non-trivial

CompFlow generates images by composing a set of conditional velocity fields from the `FLUX.1[dev]` backbone, one per condition c_i, where each condition is a natural-language sub-prompt describing a scene constraint. For this reason, evaluating CompFlow on the non-spatial, test-set category of T2I-CompBench requires decomposing each benchmark prompt, i.e. a single sentence describing a complex scene, into a list of semantically *disjoint* sub-prompts that serve as independent conditions. This decomposition step is necessary and non-trivial. In fact, the benchmark prompts were designed to be natural and descriptive rather than structurally separable, and they exhibit patterns that poorly fit with naive splitting. The non-spatial test-set split of T2I-CompBench contains 300 prompts of the form:

> *A person is wearing a hat and sunglasses while fishing.*
> *A woman is holding a camera and taking photos of a beautiful landscape.*
> *A group of activists are marching in protest, chanting slogans and waving signs.*

We concluded that a simple string-level decomposition is insufficient for these prompts. In particular, splitting on conjunctions such as *"and"* may produce fragments that are either incomplete, redundant, or semantically dependent on missing context. For example, in *"a person is wearing a hat and sunglasses while fishing"*, the phrase *"sunglasses while fishing"* is not a standalone condition unless the subject *"person"* is recovered from the first conjunct. Similarly, in *"holding a camera and taking photos of a beautiful landscape"*, the two clauses describe a coupled action involving the same agent and object, rather than two independent visual entities. Other prompts contain coordinated verb phrases, nested noun phrases, or conjunctions inside a single concept, such as *"chanting slogans and waving signs"*. A regex-based splitter would therefore either *over-segment* a single coherent event or under-specify the resulting conditions, producing condition sub-prompts that do not preserve the meaning of the original benchmark prompt. This makes prompt decomposition a semantic parsing problem rather than a purely syntactic preprocessing step.

Decomposition strategy. The design principle behind our chosen strategy is that **sub-prompts must not share actions or visual objects**, guaranteeing fairness (i.e.there is no a single prompt describing the whole scene that has to be generated). Each sub-prompt is a grammatically complete and standalone sentence describing a single action or attribute of the subject, containing the minimum information needed to ground that fragment in a Text-to-image generative setting. The decomposition preserves the subject across sub-prompts while assigning each sub-prompt its own *non-overlapping* predicate: the visual content described by any one sub-prompt does not appear in any other. Table 13 illustrates this strategy on randomly sampled benchmark prompts.

LLM-based automatic decomposition. Given the diversity of syntactic patterns, we automated the decomposition using an LLM. Each of the 300 non-spatial prompts was sent as a separate user message; the model was instructed to return only a JSON array with no additional text. The system prompt is reported below.

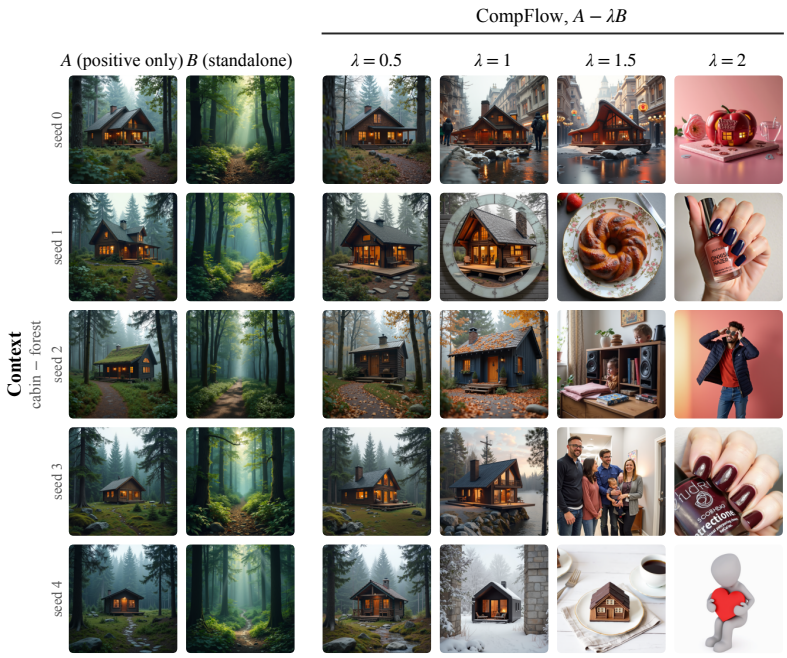

Figure 15: **Context: *"a cabin in a forest"* − *"forest"*, five seeds.** At $\lambda = 1$ the forest is gone in every seed and the cabin, with its lit windows, porch, and roofline, is re-situated in a new setting: a city street, a lakeside, a stone courtyard. The subtraction rewrites the context around the object rather than erasing the object. From $\lambda = 1.5$ the negative residual dominates and the scene leaves the prompt.

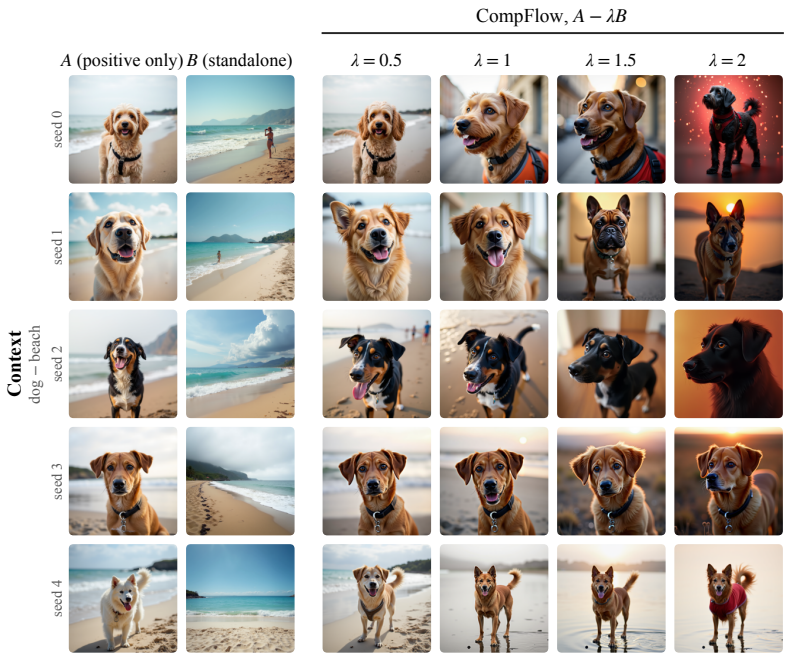

Figure 16: **Context: *"a dog on a beach"* − *"beach"*, five seeds.** The dog is preserved in all 25 composed samples, including breed, pose, and collar, while sand, surf, and sky are progressively replaced by neutral, indoor, and sunset backgrounds. This pair shows the most stable behavior in the inventory: the whole sweep up to $\lambda = 2$ stays on the positive subject.

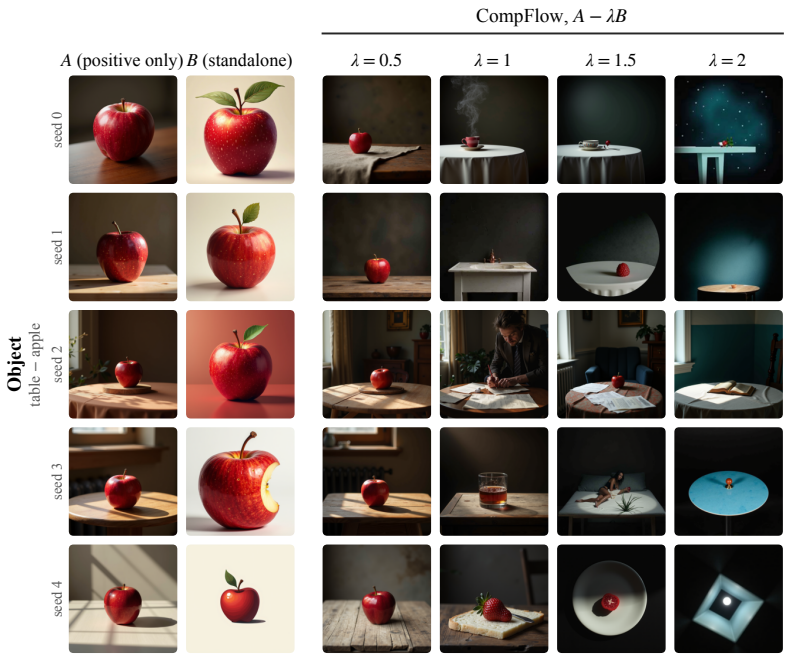

Figure 17: **Object: *"a table with a red apple"* − *"apple"*, five seeds.** At $\lambda = 1$ the apple is absent from all five seeds and the table, with its cloth, lighting, and room, remains; the freed spot is re-sampled with content consistent with the rest of the prompt (a cup, a glass, a slice of toast, an empty tabletop, a person reading at the table). Object removal is thus a re-composition of the scene, not an inpainting hole.



Figure 18: **Object: *"a street with a bicycle"* − *"bicycle"*, five seeds.** The cyclist shrinks at $\lambda = 0.5$ and has left the street at $\lambda = 1$ in every seed, while the tree-lined avenue, facades, parked cars, and lane markings of A persist; four seeds keep a coherent street up to $\lambda = 1.5$ and two up to $\lambda = 2$.

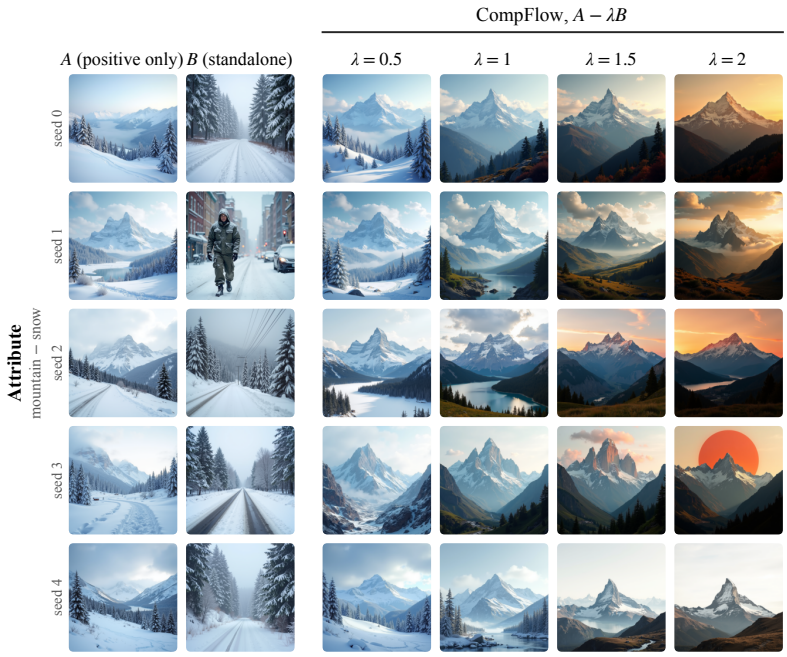

Figure 19: **Attribute: *"a snowy mountain landscape"* − *"snow"*, five seeds.** Snow cover decreases monotonically with λ in all five seeds, exposing rock, conifers, lakes, and green valleys, and the mountains persist in all 20 composed tiles. Beyond $\lambda = 1$ the scenes are relit at sunset, a consistent side effect of pushing away from the cold white palette of the subtracted attribute. This is graded attribute control with one scalar.



Figure 20: **Attribute: *"a red sports car"* − *"red"*, five seeds.** At $\lambda = 1$ the car changes color in every seed (silver, yellow, blue, purple) while remaining a sports car, and the background shifts to cyan, the complement of the subtracted hue: the negative expert pushes the sample away from red along the color axis in a predictable direction. The car is retained through $\lambda = 2$ in all seeds.



Figure 21: **Medium: *"a castle in watercolor style"* − *"watercolor"*, five seeds.** At $\lambda = 1$ all five watercolor illustrations become photographic castles on wooded hills, preserving towers, flags, and composition; four seeds keep the castle at $\lambda = 1.5$ and three at $\lambda = 2$, where the scene turns to dusk and moonlight. The medium is removed while the depicted subject is kept.

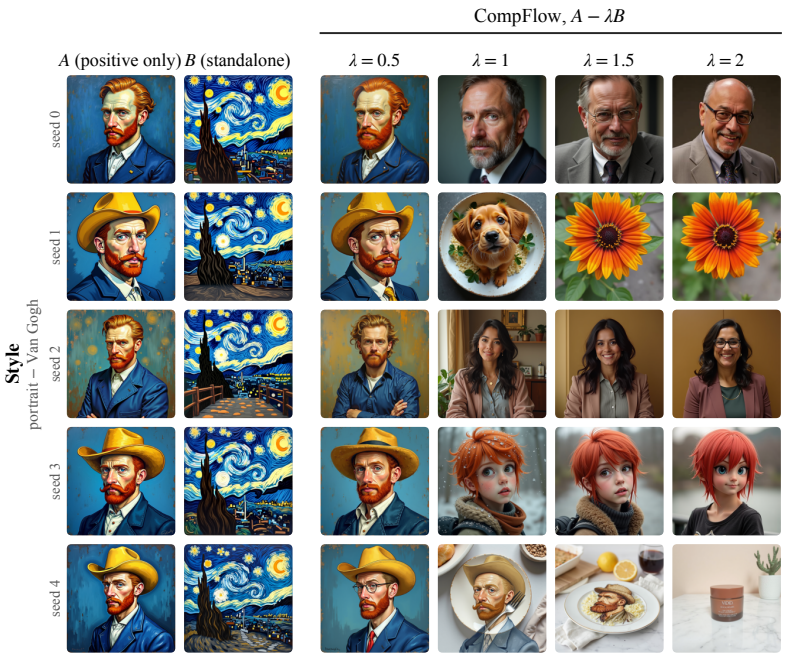

Figure 22: **Style: *"a portrait in Van Gogh style"* − *"Van Gogh style"*, five seeds.** Subtracting the artist style turns the painted portrait into a photographic portrait at $\lambda = 1$ in four of five seeds, removing the brushwork, palette, and swirling background while keeping the head-and- shoulders framing; three seeds remain portraits up to $\lambda = 2$.



Figure 23: **Control: *"a cat indoors"* − *"ocean"*, five seeds.** The negative concept is absent from A. All five seeds remain indoor cats through $\lambda = 1.5$ and cats at $\lambda = 2$; the subtraction shifts pose, framing, and palette toward warm tones and leaves the subject in place. The operator acts on the named concept, and a concept that is not present has nothing to remove.



Figure 24: **Control: *"a red car"* − *"snow"*, five seeds.** Second control with an absent negative concept. Every composed sample up to $\lambda = 2$ is a red car; the model of the car and the setting change, and the scenes drift toward warm sunset lighting, the opposite of the subtracted cold, white concept. Together with Figure 23, this confirms subject preservation under irrelevant subtraction across seeds.

Table 13: Prompt decomposition examples from the non-spatial T2I-CompBench split. Each compound benchmark prompt is split into sub-prompts with non-overlapping predicates, used as independent conditions in CompFlow.

Benchmark prompt	**Decomposition (ours)**
A person is wearing a hat and sunglasses while fishing.	`["A person is wearing a hat", "A person is wearing sunglasses", "A person is fishing with a rod"]`
A man is sitting on a park bench feeding pigeons.	`["A man is sitting on a park bench", "A man is feeding pigeons"]`
A dog is wagging its tail and playing fetch with its owner.	`["A dog is wagging its tail", "A dog is playing fetch with its owner"]`
A person is standing on a ladder and fixing a leaky roof.	`["A person is standing on a ladder", "A person is fixing a leaky roof"]`
A man is holding a basketball and practicing his jump shot.	`["A man is holding a basketball", "A man is practicing his jump shot"]`

System prompt for LLM-based decomposition

```
Your task is to decompose a natural-language image prompt into a
list of non-overlapping sub-prompts, one per independent visual
concept, suitable as separate conditioning signals for a compositional
generative model.

Rules:

    1. Each sub-prompt must be a complete, self-contained sentence
       that could stand alone as an image-generation prompt.

    2. Preserve the subject across sub-prompts.  Assign each
       sub-prompt a single, distinct action or attribute -- no action
       or object should appear in more than one sub-prompt.

    3. If the prompt describes a single atomic scene with one action,
       return a one-element list containing the original prompt
       unchanged.

    4. Return only a JSON array of strings.  No explanation, no
       preamble, no markdown fences.

Examples:
Input:  "A person is wearing a hat and sunglasses while fishing."
Output:  ["A person is wearing a hat", "A person is wearing
sunglasses",          "A person is fishing with a rod"]

Input:  "A man is sitting on a park bench feeding pigeons."
Output:  ["A man is sitting on a park bench", "A man is feeding
pigeons"]
```

The LLM output was then post-processed by parsing and validating the JSON array and fed to the compositional sampler.

Prompts used for generating images in Figure 1 and Figure 4

- `"a cute sleeping dragon" + "a pillow-book" + "a mystical forest" + "blue-ish dreamlike atmosphere"`
- `"A red fox in a snowy forest with trees", "film photo style, soft lighting, detailed, intricate, sharp focus", "amber atmosphere, cinematic, volumetric lighting, 85mm lens"`
- `"a lone traveler wearing a wool cloak" + "walking through the Scottish Highlands" + "epic cinematic landscape photography" + "muted colors"`
- `"a wonderful landscape" − "color green"`
- `"a dog playing in the park" + "a dog and its owner"`
- `"the woodcarver crafting a sculpture" + "a bird sculpture"`

T2I-CompBench evaluation. For the non-spatial test-set split, each benchmark prompt is first decomposed into sub-prompts as described in Appendix C.3; the resulting list is then fed to the sampler with uniform weights $\lambda_i = \lambda$ and $\lambda_\varnothing = 1$. All metrics are computed against the original, un-decomposed benchmark prompt, so that CompFlow is judged on its ability to reconstruct the full benchmark scene from disjoint sub-prompts rather than against the easier per-fragment task.

For each of the 300 prompts in the T2I-CompBench non-spatial split, we generate five images using independent random seeds, yielding $5 \times 300 = 1500$ images per hyperparameter configuration. We sweep the guidance weight over

$$\lambda \in \{0.5, 1, 2, 3\}$$

and the number of sampling steps over

$$N \in \{20, \ldots, 40\}.$$

We report the results obtained by the best aggregate configuration in this sweep. The best-performing setting was $\lambda = 1$; unless stated otherwise with N=28 sampling steps.

Computational cost. Figure 25 illustrates the computational cost above with a direct measurement of the composition sampler's wall-clock time as a function of the number of conditions $m \in \{1, \ldots, 6\}$.

The benchmark was run on a single NVIDIA L40S (48 GB, `bfloat16`) at 512×512 resolution with $N = 20$ ODE steps. The measured curve is essentially linear in m, in close agreement with the predicted $N \cdot (m+1)$ forward-pass count: the slope corresponds to the cost of one additional conditional forward pass per ODE step, while the intercept absorbs the unconditional pass and the constant per-step scheduler overhead. Trial-to-trial variance stays within a few percent and does not grow with m, indicating that at the resolutions we use no allocator pressure or memory thrashing kicks in, and that deployment-time inference cost is fully predictable from m alone. All the experiments were run on a single GPU under the configuration detailed in Table 5

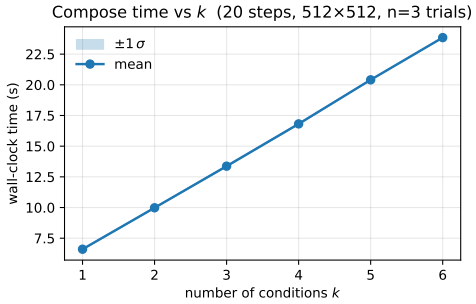

Figure 25: Wall-clock time of velocity composition as a function of the number of conditions m, measured on a single NVIDIA L40S.

C.4 Dynamics-induced style mixing

The velocity-composition operator in Eq. (6) acts pointwise along a single latent trajectory and is symmetric under permutation of the (c_i, λ_i) pairs. This structure naturally encourages *style mixing*, reflecting the attribute-blending behavior observed in Figure 11, rows 1–2, but in a more coherent and

visually graceful form. When several conditions provide compatible but distinct semantic, geometric, or stylistic directions, the composed trajectory does not necessarily instantiate each condition as a separate object. Instead, it may follow an averaged dynamical path that blends their visual attributes into a single coherent sample. We illustrate this behavior in Figure 26.



horse + sheep | sphere + cube | glossy chair + lamp | green apple + bear

Figure 26: Dynamics-induced style mixing. When conditions are composed along the same latent trajectory, CompFlow often produces coherent visual blends rather than discrete object conjunctions. The resulting images combine semantic, geometric, material, and stylistic attributes from multiple experts into a single sample.

Style and attribute fusion. When two or more sub-prompts describe visually compatible concepts of comparable salience, the composed velocity field can interpolate between their associated directions instead of selecting one condition or creating multiple separate entities. For example, composing "a horse" and "a sheep" produces an animal with attributes of both species; composing "a sphere" and "a cube" yields a rounded cubic object; and composing "a glossy chair" with "a lamp" produces an object that mixes shape, material, illumination, and function. These samples are not degenerate: they are coherent images produced by following a single composed dynamical system. The behavior reflects the fact that CompFlow performs composition in velocity space, so compatible experts can jointly deform the trajectory toward an intermediate visual mode.

Implications for compositional control. This behavior is useful when the intended prompt involves global style, material, texture, color, or aesthetic modifiers, since the velocity addition can smoothly combine such factors during sampling. However, it also reveals a limitation for prompts requiring hard binding, counting, or explicit separation of instances. If the target semantics require "a horse and a sheep" as two distinct entities, the pointwise additive dynamics alone does not specify that the two concepts should occupy different spatial regions or correspond to different object slots. In such cases, the method may instead produce a single blended concept. This suggests that velocity composition is especially well suited to style-level and attribute-level composition, while precise multi-object binding may require an additional mechanism, such as spatial conditioning, object slots, masks, or region-specific experts. In our FLUX experiments, additive composition alone does not reliably preserve instance separation or spatial binding. Additional structure, such as spatial conditioning, object slots, or region-specific experts, may improve control in these settings.

Eq. 6. Lifting them likely requires either a non-symmetric composition strategy or an explicit spatial prior on top of the per-condition velocities; we leave both directions to future work.

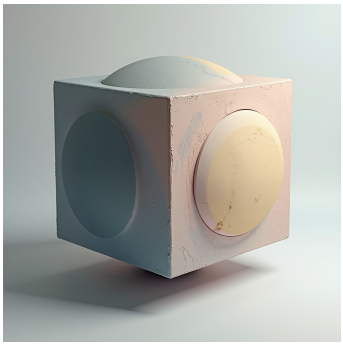

Figure 27: CompFlow output for the prompts *"a cube on the left"*The left/right assignment is destroyed and the generation collapse into a single hybrid solid.

Spatial-relational failures. The same averaging dynamics that enables the style mixing above breaks down when the composed conditions encode *spatial-relational* constraints. Figure 27 reports a representative case obtained while attempting to reproduce a CLEVR-style scene with CompFlow: each sub-prompt anchors a single primitive to a specific half of the canvas, and the FLUX backbone handles the side assignment in isolation. Under additive composition, however, the

dynamics discards the spatial binding, while the description of two simple primitives drives the same blending behavior described above; the result is a chimeric solid in which sphere-shaped patches appear painted onto the faces of a cube, rather than two distinct objects in their requested halves. We were unable to recover correct binding by varying λ, the seed, the resolution, or the number of ODE steps. The same mechanism is responsible for analogous failures on counting prompts (e.g. "a horse" composed with "a horse" or with "two horses" as a third condition does not yield two instances) and, more broadly, for any prompt whose intended semantics rely on a structured assignment of conditions to image regions or object slots. These failure regimes mark the boundary of what an additive operation can encode: spatial-relational composition appears to require an explicit spatial prior on top of the single-condition velocities, both of which fall outside the scope of this work, and are left as future development.