# VeCoR — Velocity Contrastive Regularization for Flow Matching

Supplementary Material

## A. More Implementation Details

This section elucidates more details about the concrete augmentation-like perturbation in Sec. 4.2.

### A.1. Augmentation-like Perturbation Details

Given a batch input $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, which may represent training images ($\hat{I}_+$), latents ($\hat{x}_+$), or velocities ($\hat{v}_+$), our goal is to apply augmentation-like perturbations to obtain negative samples $z_-$.

**Random Channel Shuffle** We apply a per-sample cyclic channel shift to ensure that no channel remains in its original position. Given $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, a shift $k \in \{1,\dots,C-1\}$ is sampled and applied via modular indexing, producing the perturbed output $z_-$.

**Random Crop and Resize** For a batch input $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, we uniformly sample the target area ratio and aspect ratio:

$$\alpha \sim \mathcal{U}(\text{scale}_{\min}, \text{scale}_{\max}), \qquad r \sim \mathcal{U}(\text{ar}_{\min}, \text{ar}_{\max}),$$

with default $\alpha \in [0.9, 0.95]$ and $r \in [0.95, 1.05]$.

The target crop area is

$$A_{\text{crop}} = \alpha(HW),$$

and the crop dimensions are

$$h = \sqrt{\frac{A_{\text{crop}}}{r}}, \qquad w = \sqrt{A_{\text{crop}}\, r},$$

rounded to integers and clamped to valid ranges. If the sampled dimensions fall below a threshold, we fall back to a larger crop (e.g., $0.9H \times 0.9W$). A valid crop location is sampled uniformly, and the cropped region is resized back to $(H, W)$, resulting in $z_-$.

**CutMix** Given a batch $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, we first construct a derangement permutation

$$\pi : \{1,\dots,B\} \to \{1,\dots,B\}$$

that satisfies $\pi(i) \neq i$ for every index $i$, ensuring that no sample is mixed with itself.

For each sample $z_+^{(i)}$, we draw a mixing coefficient

$$\lambda^{(i)} \sim \text{Beta}(\alpha, \alpha), \qquad \alpha = 1,$$

and compute the CutMix region scale

$$r^{(i)} = \sqrt{1-\lambda^{(i)}}.$$

The corresponding box width and height are

$$w^{(i)} = r^{(i)}W, \qquad h^{(i)} = r^{(i)}H.$$

A box center $(c_x, c_y)$ is sampled uniformly over the spatial domain. The bounding coordinates are then clipped to valid image ranges:

$$x_1 = \text{clip}\left(c_x - \frac{w^{(i)}}{2},\, 0,\, W\right),$$

$$x_2 = \text{clip}\left(c_x + \frac{w^{(i)}}{2},\, 0,\, W\right),$$

$$y_1 = \text{clip}\left(c_y - \frac{h^{(i)}}{2},\, 0,\, H\right),$$

$$y_2 = \text{clip}\left(c_y + \frac{h^{(i)}}{2},\, 0,\, H\right).$$

Finally, the rectangular region of $z_+^{(i)}$ within $(x_1 : x_2,\, y_1 : y_2)$ is replaced by the corresponding patch from the paired sample $z_+^{(\pi(i))}$, yielding the CutMix-perturbed output $z_-^{(i)}$.

**Gaussian Blur** Given a batch input $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, we apply a per-channel Gaussian blur with kernel size $k$ (odd) and standard deviation $\sigma \geq 1$. We use $k = 5$ and $\sigma = 1$.

The kernel is

$$G(u,v) = \exp\left(-\frac{u^2+v^2}{2\sigma^2}\right), \qquad u,v \in \left[-\tfrac{k-1}{2},\, \tfrac{k-1}{2}\right],$$

normalized so that $\sum_{u,v} G(u,v) = 1$.

We replicate the kernel across channels:

$$K \in \mathbb{R}^{C\times 1\times k\times k}, \qquad K_c = G,$$

and apply depthwise convolution with reflection padding

$$p = \left\lfloor \frac{k}{2} \right\rfloor,$$

which prevents artificial dark borders or edge artifacts that would otherwise arise from zero-padding during Gaussian smoothing. Finally, the blurred output defines $z_-$.

**Gaussian Noise** Given a batch input $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, we compute a noise scale for each individual sample $z_+^{(i)} \in \mathbb{R}^{C\times H\times W}$.

For each sample, we first measure its per-sample standard deviation:

$$\sigma^{(i)} = \mathrm{std}\Big(z_+^{(i)}\Big), \qquad \tilde{\sigma}^{(i)} = \frac{\sigma^{(i)}}{\sigma_{\max}},$$

where $\sigma_{\max}$ is the maximum standard deviation observed within the batch.

We then define the noise magnitude as

$$\gamma^{(i)} = \text{base\_scale}\,\big(1 - \tilde{\sigma}^{(i)}\big),$$

where base_scale is set to 255 in image space and to 1 in both latent and velocity spaces.

Finally, we inject Gaussian noise into each sample to get $z_-^{(i)}$:

$$z_-^{(i)} = z_+^{(i)} + \gamma^{(i)}\,\varepsilon^{(i)}, \qquad \varepsilon^{(i)} \sim \mathcal{N}(0,1).$$

**Color Jitter** Given a batch $z_+ \in \mathbb{R}^{B\times C\times H\times W}$, we apply per-sample color jitter composed of brightness, contrast, and saturation adjustments. We first normalize the input to obtain $z'$, and independently sample the jitter factors from

$$\lambda_\mathrm{b},\, \lambda_\mathrm{c},\, \lambda_\mathrm{s} \sim \mathcal{U}(1-\delta,\, 1+\delta), \qquad \delta = 0.2.$$

**Brightness.**

$$z' \leftarrow \lambda_\mathrm{b}\, z_+.$$

**Contrast.** Let $\mu = \mathrm{mean}(z')$ denote the global mean intensity:

$$z' \leftarrow (z' - \mu)\lambda_\mathrm{c} + \mu.$$

**Saturation.** Let $g = \mathrm{mean}_c(z')$ be the per-pixel channel average:

$$z' \leftarrow (z' - g)\lambda_\mathrm{s} + g.$$

These three operators are applied in a random order. The final result is clamped to $[0, 1]$ and rescaled as needed to obtain $z_-$.

# B. More Results

In this section, we provide additional quantitative and qualitative results.

## B.1. ImageNet-1K Results with ODE Sampling

Tab. 7 reports the ImageNet-1K results under ODE sampling. Across all SiT backbones, integrating VeCoR yields clear improvements in the main quality metrics, achieving lower FID and higher IS under the same sampling budget (50 NFEs, Heun2).

Table 7. Results on ImageNet-1K 256×256 using SiT backbones (same seed, 50 NFEs, Heun2).

| Model | FID ↓ | IS ↑ | sFID ↓ | Prec. ↑ | Rec. ↑ |
|---|---|---|---|---|---|
| SiT-S/2 [28] | 59.28 | 23.43 | 9.33 | 0.40 | **0.59** |
| **+VeCoR (Ours)** | **55.11** | **24.34** | **8.53** | **0.41** | **0.59** |
| SiT-B/2 [28] | 37.07 | 40.27 | **6.79** | 0.51 | **0.65** |
| **+VeCoR (Ours)** | **33.76** | **41.22** | 7.76 | **0.53** | 0.63 |
| SiT-L/2 [28] | 21.9 | 63.84 | **5.58** | 0.61 | **0.64** |
| **+VeCoR (Ours)** | **19.81** | **66.16** | 7.41 | **0.63** | 0.62 |
| SiT-XL/2 [28] | 18.59 | 72.05 | **5.30** | 0.63 | **0.64** |
| **+VeCoR (Ours)** | **16.55** | **76.17** | 7.21 | **0.65** | 0.62 |

Although FID, IS, and Precision improve, we observe mild decreases in sFID and Recall under certain configurations. A possible explanation is that, in a fully deterministic ODE setting, the additional signals from VeCoR about "where not to go" may guide the trajectory to remain closer to certain regions of the manifold, which could slightly limit the diversity of viable generation paths.

Overall, these shifts are small relative to the overall gains, and VeCoR remains beneficial under both SDE- and ODE-based sampling.

## B.2. Text-to-Image Qualitative Results

We provide the text-to-image visual comparisons in Fig. 7, which illustrate that, under identical sampling conditions, incorporating VeCoR leads to outputs with better color consistency and stronger semantic alignment to the input prompts.

# C. On the Effectiveness of the VeCoR Loss

For completeness, we provide the analytical form of our velocity contrastive regularization (VeCoR). Although its structure resembles the contrastive FM objective in ΔFM [40], the intent is fundamentally different: ΔFM leverages contrastive signals *across conditions* to enforce class-level separability, whereas our formulation applies contrastive supervision directly at the *dynamics level* to enhance trajectory stability and suppress off-manifold drift during sampling. Thus, despite superficial similarities, the contrastive role in VeCoR is intrinsically distinct.

We begin by expressing the VeCoR objective in its expectation form:

$$\mathcal{L}^{(\mathrm{VeCoR})}(\theta) = \mathbb{E}\Big[\|\mathbf{v}_\theta(\hat{x}_t, t) - \hat{\mathbf{v}}_+\|_2^2 - \lambda \sum_{k=1}^{K} \|\mathbf{v}_\theta(\hat{x}_t, t) - \hat{\mathbf{v}}_-^{(k)}\|_2^2\Big], \tag{8}$$

where the expectation is taken over timesteps, perturbed states $\hat{x}_t$, and injected noise.


Figure 7. **Qualitative comparison on text-to-image generation (MS-COCO).** We use classifier-free guidance with $w = 2.0$ and using (same seed, 50 NFEs, Euler–Maruyama).

**Step 1: Expand and collect quadratic terms.** Let $\mathbf{v}_\theta = \mathbf{v}_\theta(\hat{x}_t, t)$ for brevity. Expanding the squared terms and applying linearity of expectation yields

$$\mathcal{L}^{(\mathrm{VeCoR})}(\theta) = \mathbb{E}\Big[(1-\lambda K)\,\mathbf{v}_\theta^\top \mathbf{v}_\theta - 2\,\mathbf{v}_\theta^\top\Big(\hat{\mathbf{v}}_+ - \lambda \sum_{k=1}^{K} \hat{\mathbf{v}}_-^{(k)}\Big)\Big] + \text{const}, \tag{9}$$

where the constant aggregates all terms independent of $\mathbf{v}_\theta$.

**Step 2: Compute the minimizer.** Taking the gradient of (9) with respect to $\mathbf{v}_\theta$ and setting it to zero gives

$$2(1-\lambda K)\,\mathbf{v}_\theta^* = 2\,\mathbb{E}\Big[\hat{\mathbf{v}}_+ - \lambda \sum_{k=1}^{K} \hat{\mathbf{v}}_-^{(k)}\Big]. \tag{10}$$

Dividing both sides by $2(1-\lambda K)$ yields the closed-form solution:

$$\mathbf{v}_\theta^* = \frac{\mathbb{E}[\hat{\mathbf{v}}_+] - \lambda \sum_{k=1}^{K} \mathbb{E}[\hat{\mathbf{v}}_-^{(k)}]}{1-\lambda K}. \tag{11}$$

This derivation shows that the VeCoR objective preserves the FM fixed point while adding a contrastive correction that suppresses destabilizing dynamical alternatives. Unlike $\Delta$FM—whose contrastive term separates flows across conditioning labels—the negative velocities in VeCoR represent dynamical directions that would drive trajectories toward undesirable, off-manifold evolution. In this view, VeCoR acts as a corrective force that steers the predicted velocity away from off-manifold directions and reinforces stable, data-consistent trajectories. To maintain this behavior in a mathematically well-posed manner, the quadratic coefficient $(1-\lambda K)$ must remain positive, ensuring that the objective retains a proper minimization structure. This leads to the requirement

$$\lambda K < 1,$$

which prevents the loss from becoming ill-conditioned.