Mask-Conditioned Stochastic Interpolants for Patch Inpainting on CIFAR-10 (Dogs)

Abstract

We study mask-conditioned image inpainting on CIFAR-10 (class "dog") using stochastic interpolants and flow matching. At each training step, we sample a binary mask to designate visible pixels and replace masked pixels with Gaussian noise to form the starting state. A time-dependent velocity field $b_\theta(x, t, m)$ is learned along a designed linear interpolant from the masked-noisy state to the ground-truth image. At generation time, we integrate the Probability Flow ODE and finally overwrite the visible pixels with their original values so that only masked regions change. We document data, conditioning, equations, architecture, training, sampling, dimensions, and parameter counts.

1 Data and preprocessing

Dataset. CIFAR-10 (train split), filtered to class 5 ("dog") only, yielding 5000 images (out of 50000). Images are in $\mathbb{R}^{3\times32\times32}$.

Normalization. Each image is converted with `ToTensor` and normalized using mean $(0.5, 0.5, 0.5)$ and std $(0.5, 0.5, 0.5)$. Values are centered near 0 and lie roughly in $[-1, 1]$.

Per-step sampling. At every training step:

- we sample a batch of size `batch_size` $= 32$ from the 5000 dogs (i.i.d. random indices),
- we sample a binary mask $m \in \{0, 1\}^{3\times32\times32}$ by placing `num_patches` $= 4$ square patches of size `patch_size` $= 8$ to zero (1 = visible, 0 = masked),
- we sample i.i.d. noise $\xi \sim \mathcal{N}(0, I)$ (same shape as the image).

The mask is identical across the 3 color channels (replicated by construction). For network conditioning, only the first mask channel is used as a binary map, see Sec. **??**.

2 Problem and variables

We denote $x_1 \sim p_{\text{data}}$ the target (dog) image, m the binary mask, and $\xi \sim \mathcal{N}(0, I)$ the i.i.d. noise. The inpainting initial state is

$$x_0 \;=\; m \odot x_1 \;+\; (1 - m) \odot \xi,$$

where $\odot$ denotes the Hadamard product. The time $t \sim \mathcal{U}([0, 1])$ is sampled independently.

3 Interpolant and target dynamics

We use a *one-sided linear* interpolant between x_0 and x_1:

$$x_t \;=\; (1-t)\,x_0 \;+\; t\,x_1, \qquad t \in [0,1],$$

whose pathwise derivative is constant:

$$\dot{x}_t \;=\; x_1 - x_0.$$

Flow matching learns a time-dependent velocity field $b_\theta(x,t,m)$ matching this target derivative along the interpolant. An auxiliary term η_ϕ is also trained (within `interflow`) for the shared stochastic interpolant.

4 Training objectives

We train two networks: b_θ (drift/velocity) and η_ϕ (auxiliary). The `interflow` losses are `one-sided-b` and `one-sided-eta` with a shared interpolant:

$$\begin{aligned}
\mathcal{L}_b(\theta) \;&=\; \mathbb{E}_{x_1,\xi,m,t}\Big[w(m)\,\big\|\,b_\theta(x_t,t,m) - (x_1 - x_0)\big\|_2^2\Big],\\
\mathcal{L}_\eta(\phi) \;&=\; \text{(auxiliary one-sided-eta term defined by the interpolant)}.
\end{aligned}$$

Mask reweighting emphasizes masked regions:

$$w(m) \;\propto\; m \;+\; \lambda\,(1-m), \qquad \lambda = \texttt{mask\_loss\_weight} = 10.0.$$

In practice, we apply the efficient approximation $\mathcal{L} \leftarrow \bar{w}\cdot\mathcal{L}$ where $\bar{w}$ is the per-batch mean of per-pixel weights.

5 Network architecture

Backbone. Two 2D U-Nets (same shape) with GroupNorm and base channels 64. Encoder path with 3 levels (stride 2), bottleneck, and symmetric decoder with skip connections.

Explicit conditioning. Each U-Net sees 5 channels:

$$\underbrace{3}_{\text{image}} \;+\; \underbrace{1}_{\text{time } t} \;+\; \underbrace{1}_{\text{mask } m} \;=\; 5.$$

The scalar time t is broadcast spatially to a constant 32×32 map. For the mask, only its first channel $[1 \times 32 \times 32]$ is concatenated (the mask image is built with 3 identical channels for convenience).

Wrappers and I/O. We operate in vector space with

$$d \;=\; 3 \times 32 \times 32 \;=\; 3072.$$

The wrappers reshape the flattened input to $x \in \mathbb{R}^{3\times32\times32}$, concatenate the t and m maps to form a $[5 \times 32 \times 32]$ tensor, and feed the U-Net. The output is $[3 \times 32 \times 32]$, flattened back to $\mathbb{R}^{3072}$.

Parameter counts. For `in_channels` $= 5$, `out_channels` $= 3$, `base_channels` $= 64$:

- b_θ network: $\approx$ 8,565,315 parameters,
- η_ϕ network: $\approx$ 8,565,315 parameters,
- total: $\approx$ 17,130,630 trainable parameters.

6 Training procedure

Loop. At each step:

1. sample a batch of dog images and a fresh mask per image;
2. form $x_0 = m \odot x_1 + (1 - m) \odot \xi$; flatten x_0, x_1 to $\mathbb{R}^{3072}$;
3. sample $t \sim \mathcal{U}([0, 1])$;
4. pass m into the wrappers (explicit conditioning) and compute $\mathcal{L}_b, \mathcal{L}_\eta$ (one-sided scheme);
5. apply $\bar{w}$ scaling (focus on masked pixels), backprop, optimizer step, scheduler step.

Optimization. Adam with `lr` $= 10^{-4}$. CosineAnnealingLR with $T_{\max}$ = `n_epochs` $= 5000$, $\eta_{\min} = 10^{-6}$. No gradient clipping is applied (we log gradient norms).

Logging and stopping. Losses, gradient norms, and LR are logged every `metrics_freq` $= 100$ iterations. Plots and checkpoints are saved every `plot_freq` $= 500$. Training stops after a fixed budget `n_epochs` $= 5000$ (no validation/early stopping).

7 Generation (inpainting)

We integrate the Probability Flow ODE:

$$\frac{dX_t}{dt} \;=\; b_\theta(X_t, t, m), \qquad X_0 = x_0, \quad t \in [0, 1],$$

using a `dopri5` solver with `n_step` $= 10$. The trajectory is deterministic conditional on (x_0, m); randomness stems from ξ and m. At the final time, we overwrite visible pixels from the original image:

$$\hat{x}_1 \;=\; m \odot x_1 \;+\; (1 - m) \odot X_1^{\text{gen}},$$

ensuring only masked regions are changed.

8 Shapes and dimensions

- Flattened image space: $d = 3072$.
- U-Net input: $[5, 32, 32]$ (3 image + 1 time + 1 mask).
- U-Net output: $[3, 32, 32]$ (flattened to [3072]).
- Wrapper inputs: either [3072] with t provided separately, or [3073] with t appended.

These dimensions do not depend on `num_patches`: it affects only the geometry of the binary mask, not the number of channels.

9 Hyperparameters and constants

Quantity	Value
Class	CIFAR-10 "dog" (id = 5)
Subset size	5000 images
Resolution	32×32, 3 channels
Normalization	mean $(0.5, 0.5, 0.5)$, std $(0.5, 0.5, 0.5)$
Batch size	32
patch size	8
num patches	4
mask loss weight λ	10.0
Interpolant	one-sided linear (path $x_t = (1-t)x_0 + tx_1$)
Losses	one-sided-b, one-sided-eta (shared)
Optimizer	Adam, `lr` $= 10^{-4}$
Scheduler	CosineAnnealingLR, $T_{\max} = 5000$, $\eta_{\min} = 10^{-6}$
ODE integrator	Probability Flow, `method=dopri5`, `n_step` $= 10$
Stopping	fixed number of steps: 5000
Params b_θ	$\approx$ 8,565,315
Params η_ϕ	$\approx$ 8,565,315

10 Remarks and limitations

- No validation split or early stopping: termination is by fixed budget.
- Masks are resampled at every step (online data augmentation); no per-image fixed mask.
- The model is *conditional flow matching* (not score-based diffusion, not a Schrödinger bridge). Generation is deterministic conditional on (x_0, m).
- Mean reweighting $\bar{w}$ is an efficient global approximation (instead of exact pixel-wise weighting in the sum).