JOINT FLOW MATCHING: CONSISTENT GENERATIVE AND DISCRIMINATIVE INFERENCE FROM A SHARED FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

We study Joint Flow Matching (JFM), a flow-based framework for learning a joint distribution over paired variables such as images and class labels. JFM transports the variables in opposite directions: an image is transported from noise to data while its label is transported from data to noise. This provides a single probabilistic model in which generation and classification are represented as complementary conditional inferences. We formulate the joint distribution induced by this construction and characterise the relationship between its forward and reverse conditionals. We investigate the resulting model for classification and conditional generation, including confidence calibration and consistency between generated samples and classifier predictions. Experiments on conditional datasets demonstrate competitive classification accuracy, well-calibrated confidence without post-hoc calibration, and classifier-consistent generation. We further explore the shared joint distribution as a basis for interpreting the relationship between discriminative and generative behaviour.

1 INTRODUCTION

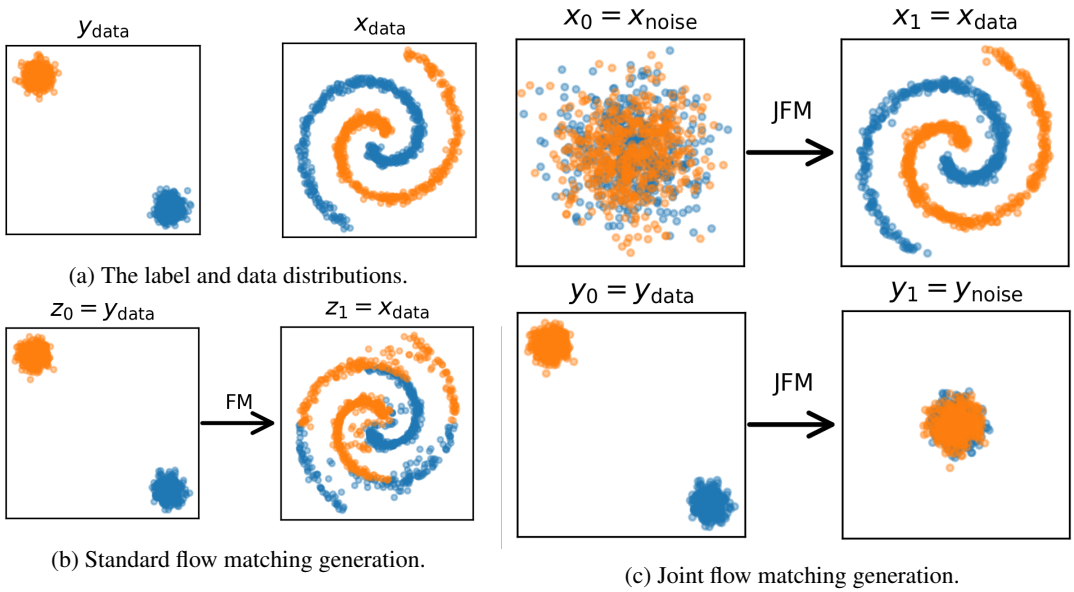
The goal of any generative model is to learn a mapping from a tractable noise distribution to a complex data distribution. Continuous normalising flows (CNFs) (Chen et al., 2018) accomplish this by learning a time-indexed velocity field that transports probability mass from noise at $t = 0$ to data at $t = 1$. Flow Matching (Lipman et al., 2023) has emerged as a prominent training paradigm for CNFs, enabling scalable training by forgoing the need for costly numerical integration at each epoch. Most generative models (including CNFs) treat generation as a one-directional process: noise samples are pushed forward to approximate a data distribution. When the data consists of paired variables (X, Y) (e.g., images and class labels), these models typically capture $p(X|Y)$. Sampling the conditional distribution $p(Y|X)$ requires either a separate discriminative model or costly approximate inference. There is no guarantee of consistency between the two directions.

We present *Joint Flow Matching* (JFM), a mechanism that addresses this asymmetry by assigning noise and data distributions to opposite temporal endpoints: $p_0(X, Y) = p_{\text{noise}}(X)p_{\text{data}}(Y)$ and $p_1(X, Y) = p_{\text{data}}(X)p_{\text{noise}}(Y)$. A single reversible flow between these endpoints supports two conditional directions:

- **Generation** ($t : 0 \to 1$): given a label $y_0 \sim p_{\text{data}}(Y)$, integrate forward to obtain image $x_1 \sim p_{\text{data}}(X)$ conditioned on the label.
- **Classification** ($t : 1 \to 0$): given an image $x_1 \sim p_{\text{data}}(X)$, integrate backward to obtain label y_0 conditioned on the image.
- **Consistency**: both conditional directions are induced by the same joint distribution and are therefore mutually consistent.

This reversible conditional flow matching mechanism was also proposed by Caetano et al. (2026) under the name *Symmetrical Flow Matching* (SFM). While SFM explores the generative and classification capabilities of the unified model, our work focuses on the joint distribution induced by the construction. We establish theoretical guarantees for the consistency of its generative and dis-

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.



(a) The label and data distributions.

(b) Standard flow matching generation.

(c) Joint flow matching generation.

Figure 1: Forward flow in FM and JFM trained to model $p(x|y)$ and $p(y|x)$ for the dataset shown in (a). Colour indicates the class conditioning, corresponding to the distinct arms of the spiral. FM can in principle map arbitrary source and target distributions but flowing y directly to x to enable bidirectional conditioning performs poorly for the spiral under the linear interpolation path. (b). JFM has no trouble recovering $p(x|y)$ under the same transport (c). Reverse flow for the two models is shown in Appendix A.

criminative conditionals. We then demonstrate the conditional-generation behaviour of JFM compared to standard FM using the spiral dataset (Figure 1). Finally, we show that the consistency between the generative and discriminative conditionals enables functionality beyond generation and classification on real image datasets, including calibrated predictions and new avenues for model interpretability.

This work is complementary to SFM, which established that a reverse-coupled flow can support unified generation and classification (Caetano et al., 2026). The motivation for JFM is not that a jointly trained model should outperform separately trained classifiers or generators, but that a shared joint model provides a relationship between generation and classification that is absent when the two are trained independently. By deriving both directions of inference from the same model, JFM allows generated samples to be evaluated by the corresponding classifier and class-conditioned transformations to be examined for their effect on the model's predictions. These capabilities provide a model-internal means of examining the learned relationships without requiring prior specification of features or failure modes.

Contributions

1. We formulate Joint Flow Matching as a joint probabilistic model over paired variables (X, Y), providing a formal probabilistic characterisation of the shared reversible flow underlying generative and discriminative inference.

2. We show theoretically that the flow construction yields consistency between the conditional distributions $p_{\text{JFM}}(X \mid Y)$ and $p_{\text{JFM}}(Y \mid X)$, without requiring additional consistency constraints or regularisation.

3. We empirically investigate the implications of this consistency for classification, conditional generation, and their relationship.

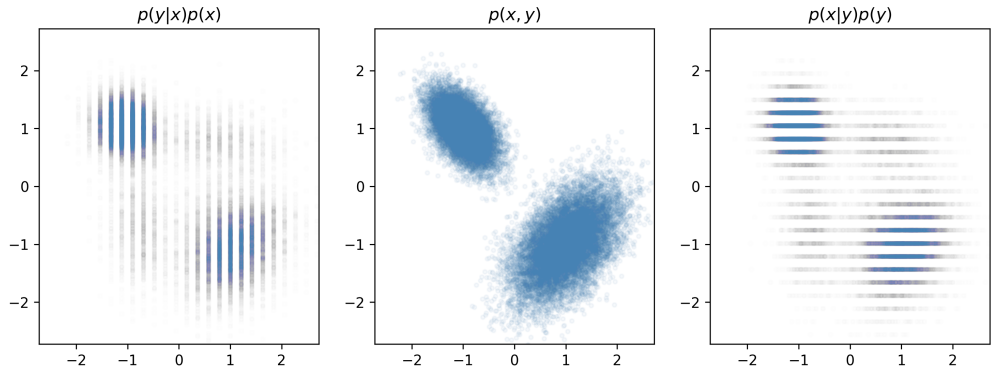


Figure 2: Illustration of the consistency of JFM. A flow is trained on the marginal distributions along the X and Y axes. The central plot shows the true joint distribution of the two variables. The left and right plots show the generated joint when conditioning on the X or Y variables respectively. Note the shadows in the upper right and lower left quadrants of the generated distributions; although not present in the true distribution they are consistent between the generated marginals – generated joints match. The bands in the joints are for visualisation purposes only to make it clear which axis has been sampled over.

2 Theory

2.1 Setup and Endpoint Distributions

Consider a joint space $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^{|\mathcal{X}|} \times \mathbb{R}^{|\mathcal{Y}|}$ with a temporal parameter $t \in [0, 1]$. Let $p_{\text{data}}(X)$ and $p_{\text{data}}(Y)$ denote the marginal data distributions, with tractable noise distributions $p_{\text{noise}}(X)$ and $p_{\text{noise}}(Y)$ defined over the same spaces. We define the endpoint distributions as:

$$p_0(x, y) = p_{\text{noise}}(x)\, p_{\text{data}}(y), \qquad p_1(x, y) = p_{\text{data}}(x)\, p_{\text{noise}}(y). \tag{1}$$

Both are valid joint distributions: at each endpoint, the two factors are independent by construction, since the noise distribution is chosen independently of the data distribution. Unlike standard CNFs, in which noise and data are assigned uniformly across variables at $t = 0$ and $t = 1$ respectively, here they are assigned in opposite roles for both X and Y (Caetano et al., 2026).

We learn a continuous normalising flow $\Phi_t : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ defined by a velocity field v:

$$\frac{\mathrm{d}}{\mathrm{d}t}\Phi_t(x, y) = v(t, \Phi_t(x, y)), \qquad \Phi_0(x, y) = (x, y). \tag{2}$$

Under regularity conditions on v, Φ_t is a diffeomorphism for each t, and in particular Φ_1 is a bijection. The flow is trained so that $(\Phi_1)_* p_0 = p_1$, i.e. Φ_1 pushes p_0 forward to p_1. By the change of variables formula,

$$p_1(x_1, y_1) = p_0\big(\Phi_1^{-1}(x_1, y_1)\big) \cdot \left|\det J_{\Phi_1^{-1}}(x_1, y_1)\right|, \tag{3}$$

where $J_{\Phi_1^{-1}}$ denotes the Jacobian of Φ_1^{-1}. Concretely, the transported sample $(x_1, y_1) = \Phi_1(x_0, y_0)$ is obtained by integrating the ODE from Equation 2:

$$(x_1, y_1) = (x_0, y_0) + \int_0^1 v(t, x_t, y_t)\, dt. \tag{4}$$

2.2 The Joint Distribution

Standard CNFs couple variables at a shared data endpoint ($t = 1$), making the joint $p_1(x, y) = p_{\text{data}}(x, y)$ explicit. In JFM the endpoint distributions are factored by design, so the learned coupling between X and Y is not directly visible at either end. It is instead encoded in the joint $p_{\text{JFM}}(X, Y)$, defined as the joint distribution of the pair (x_1, y_0): the $\mathcal{X}$-output of the forward flow paired with the $\mathcal{Y}$-input, or equivalently the $\mathcal{Y}$-output of the reverse flow paired with the $\mathcal{X}$-input.

Formally, we define the map $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathcal{X} \times \mathcal{Y}$ by

$$\Psi(x_0, y_0) = \big(\pi_x(\Phi_1(x_0, y_0)),\ y_0\big), \tag{5}$$

where π_x denotes the projection onto $\mathcal{X}$ by discarding the $\mathcal{Y}$ variable. The joint distribution is the pushforward $p_{\text{JFM}} := \Psi_* p_0$. We assume that for almost every $y_0 \in \mathcal{Y}$ the map $\pi_x \circ \Phi_1(\cdot, y_0) : \mathcal{X} \to \mathcal{X}$ is a diffeomorphism. Under this assumption, Ψ is itself a diffeomorphism, so the change of variables formula applies. Since Ψ transforms $x_0 \mapsto x_1$ while holding y_0 fixed, its Jacobian is block-triangular:

$$J_\Psi(x_0, y_0) = \begin{pmatrix} \partial x_1/\partial x_0 & \partial x_1/\partial y_0 \\ 0 & I \end{pmatrix},$$

so $|\det J_\Psi| = |\det \partial x_1/\partial x_0|$. Applying the change of variables formula:

$$p_{\text{JFM}}(x_1, y_0) = p_0(x_0, y_0) \left| \det \frac{\partial x_0}{\partial x_1}\bigg|_{y_0} \right| = p_{\text{noise}}(x_0) p_{\text{data}}(y_0) \left| \det \frac{\partial x_0}{\partial x_1}\big|_{y_0} \right|, \tag{6}$$

where $x_0 = \big(\pi_x \circ \Phi_1(\cdot, y_0)\big)^{-1}(x_1)$. The Jacobian factor is replaced by a partial Jacobian for fixed y_0. The same derivation can be made using the reverse flow starting with fixed x_1 and finding $y_1 \mapsto y_0$.

While Φ_1 is a diffeomorphism on the full product space $\mathcal{X} \times \mathcal{Y}$ by the regularity of v, this does not in general imply that the slice maps $\pi_x \circ \Phi_1(\cdot, y_0) : \mathcal{X} \to \mathcal{X}$ are diffeomorphisms. The slice diffeomorphism assumption is not derivable from the ODE regularity conditions alone, but it is consistent with the inductive bias of neural velocity fields: a violation would require two distinct noise inputs, at the same label y_0, to produce the same data output x_1 under the learned flow. However, we almost never (in the statistical sense) observe this in practical neural velocity fields.

2.3 Consistency of Generative and Discriminative Conditionals

The central claim of JFM is that forward and backward integration through the same trained flow yield samples from the two conditionals of a single joint distribution p_{JFM}, guaranteeing consistency between generation and classification.

- **Forward (generation):** draw $x_0 \sim p_{\text{noise}}(X)$ and $y_0 \sim p_{\text{data}}(Y)$, integrate forward to obtain $(x_1, y_1) = \Phi_1(x_0, y_0)$. The output x_1 is a sample from $p_{\text{JFM}}(x_1|y_0)$.
- **Backward (classification):** given $x_1 \sim p_{\text{data}}(X)$, draw $y_1 \sim p_{\text{noise}}(Y)$ and integrate backward to obtain $(x_0, y_0) = \Phi_1^{-1}(x_1, y_1)$. The output y_0 is a sample from $p_{\text{JFM}}(y_0|x_1)$.

Both claims follow directly from the definition of p_{JFM} as $\Psi_* p_0$. In the forward direction, $x_1 = \pi_x(\Phi_1(x_0, y_0))$, so for fixed y_0 the distribution of x_1 is exactly $p_{\text{JFM}}(x_1|y_0)$. In the backward direction, fixing x_1 and marginalising over $y_1 \sim p_{\text{noise}}(Y)$ via Φ_1^{-1} recovers the same conditional $p_{\text{JFM}}(y_0 \mid x_1)$, since Φ_1 is bijective and maps the prescribed source distribution to the prescribed target distribution. Bayes' theorem then gives the consistency relation:

$$p_{\text{JFM}}(x_1|y_0)\, p_{\text{data}}(y_0) = p_{\text{JFM}}(y_0|x_1)\, p_{\text{data}}(x_1). \tag{7}$$

This relationship holds for any flow Φ_1 satisfying the endpoint conditions in Equation 1. In standard CNFs both variables are transported from noise to data simultaneously, and there is no analogous backward pass that isolates $p(Y|X)$: fixing x_1 and integrating backward recovers $x_0 \sim p_{\text{noise}}(X)$ but gives no information about y. There is no way to decompose the combined data distribution, which JFM solves by isolating one variable at either end of the time domain.

While the theoretical consistency result is formally established only for continuous Y, our experiments suggest that the construction can be applied effectively to categorical variables in practice. A formal treatment of discrete Y, including continuous relaxations or simplex-based formulations, is left for future work.

2.4 Training Objective

Following standard Flow Matching (Lipman et al., 2023), we express the velocity field as a parametric model and train it by minimising a conditional flow matching loss. For each training pair $(x_1, y_0) \sim p_{\text{data}}(X, Y)$ we sample noise variables $x_0 \sim p_{\text{noise}}(X), y_1 \sim p_{\text{noise}}(Y)$ and time $t \sim \mathcal{U}(0, 1)$.

We then form the linearly interpolated intermediate states and their corresponding conditional target velocities:

$$x_t = (1-t)x_0 + tx_1, \qquad u_x = x_1 - x_0,$$
$$y_t = (1-t)y_0 + ty_1, \qquad u_y = y_1 - y_0.$$

Note that $u_x = x_1 - x_0$ is a positive-direction target velocity (noise to data), while $u_y = y_1 - y_0$ is a negative-direction target (data to noise). This is due to the reversed roles of the Y variable inherent to JFM. Using the conditional flow matching trick proposed by Lipman et al. (2023) and the optimal transport probability path to interpolate velocities across time we construct the training loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{\substack{t\sim\mathcal{U}[0,1] \\ (x_1,y_0)\sim p_{\text{data}}(X,Y) \\ x_0\sim p_{\text{noise}}(X), y_1\sim p_{\text{noise}}(Y)}} \|v_x^\theta(x_t, y_t, t) - u_x(x_t)\|^2 + \|v_y^\theta(x_t, y_t, t) - u_y(y_t)\|^2. \quad (8)$$

3 Related Work

There are many families of generative models. Generative adversarial networks (Goodfellow et al., 2014) train a generator and discriminator jointly via a minimax objective. While capable of high-quality synthesis, training a GAN is often unstable (Srivastava et al., 2017; Thanh-Tung & Tran, 2020), the generator and discriminator are architecturally separate, and there is no mechanism for consistent conditional inference. Variational autoencoders (Kingma et al., 2014) introduce an encoder-decoder pair trained by modelling the latent-space distribution, but are limited in both sample quality and density estimation accuracy. Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021; Yang et al., 2023) achieve state-of-the-art sample quality by learning to reverse a fixed forward-noising process and have been extensively studied for their generalisation properties (Kadkhodaie et al., 2024; Niedoba et al., 2025; Vastola, 2025).

Discrete normalising flows (Tabak & Turner, 2013; Rezende & Mohamed, 2015; Dinh et al., 2017; Kingma et al., 2016) construct a bijection from a tractable base distribution to the data distribution using a physically inspired velocity field. The bijective nature of normalising flows gives rise to exact likelihood computation, although discrete normalising flows require expensive computation for training and inference. Continuous normalising flows (CNFs) (Chen et al., 2018; Grathwohl et al., 2019) generalise discrete normalising flows to a continuous-time setting, parameterising the dynamics of a probability density over time and using a neural ODE to model them. Maximum-likelihood training of CNFs requires expensive simulation of the ODE during each gradient step; Flow Matching (Lipman et al., 2023) resolves this by forming a tractable conditional velocity field with equivalent gradient once marginalised.

3.1 Generation-Classification Duality

Several lines of work have sought to unify generation and classification in a single model. Energy-based models (Du & Mordatch, 2019; Song & Kingma, 2021) define a scalar energy $E(x, y)$ from which both $p(x)$ and $p(y|x)$ can be derived. However, inference requires expensive MCMC methods and uses an intractable partition function, making exact likelihoods impossible. Joint energy-based models such as JEM (Grathwohl et al., 2020) reinterpret a discriminative classifier as an energy-based generative model, but generation quality suffers compared to dedicated generative models and MCMC remains a bottleneck (although this was addressed in Grathwohl et al., 2021). Classifier-free guidance (Ho & Salimans, 2021) steers diffusion samples toward a desired class at inference time but fails to produce a consistent joint model.

Li et al. (2023), building on the work of Clark & Jaini (2023), construct a classifier from a diffusion generator by comparing the quality of multiple restorations of an image corrupted with noise, each time conditioning on a different candidate from a set of classification labels. Though re-purposing of the generator is a useful trick, it does not give actual consistency between generation and classification. CycleGAN Zhu et al. (2017) employs a similar training method to our FM setup in the

spiral example in Figure 1, learning a direct mapping from one conditional to another. Their work can be used to map from images conditioned on one label to those conditioned on another. JFM can achieve this, as illustrated in Figure 3, in which we switch the conditional y_0 while keeping the noise x_0 derived from an image, producing new images that swap the object retaining independent context. The consistency property of JFM means we use this visualisation to show what is correlated and what is not with the class membership of the model.

Classical saliency methods (e.g. Simonyan et al., 2014; Zeiler & Fergus, 2014, see Li et al., 2021, for a metastudy) compute input-gradient maps or feature activation visualisations for a fixed classifier, but these are separate from any generative model. GAN-based approaches (Goetschalckx et al., 2019) explore the latent geometry of the generator model but require a separately trained discriminator to assign meaning to generated images. Integrated gradients (Sundararajan et al., 2017) and Shapley-value methods (Lundberg & Lee, 2017) give semantic attributions to inputs for classifiers but are decoupled from any generative model. The joint distribution in JFM arises naturally from the coupling learned by the flow itself. The attributions are drawn from the same model that performs both generation and classification.

Caetano et al. (2026) proposed Symmetrical Flow Matching (SFM), which also uses opposite endpoint assignments for image and label variables to enable both generative and discriminative inference within a single flow model. SFM develops this construction as a unified framework for image generation, semantic segmentation, and classification, with an emphasis on task performance and efficient bidirectional inference. In contrast, JFM focuses on the probabilistic structure induced by the shared flow and formally characterises the consistency between the corresponding generative and discriminative conditionals. JFM also uses a simpler label-flow construction in which the label velocity is predicted from the bottleneck representation rather than propagated through the full image-processing network. With this formulation, JFM provides stable reverse classification across inference step counts without requiring label perturbation.

4 METHODOLOGY

The JFM framework can be implemented using any parametric velocity model $v(t, x_t, y_t)$ and trained using the flow matching objective set out in Equation 8. The exact form of the velocity model will almost certainly be determined by the application. For simple toy examples (see Section 5.1) we have employed a small multilayer-perceptron with X and Y concatenated for input and output. All training runs used AdamW optimiser with default weight decay of 0.01 and learning rate decayed with cosine annealing.

4.1 IMAGE GENERATION-CLASSIFICATION

For image generation-classification we used an ADM-style U-Net backbone (Dhariwal & Nichol, 2021), following prior diffusion and flow-matching work, with the key addition of a classification head attached at the bottleneck for label velocity output. A detailed schematic of the architecture and its dimensions is provided in Appendix B.

The U-Net takes as input the image $x_t \in \mathbb{R}^{C\times H\times W}$, label $y_t \in \mathbb{R}^K$, and time $t \in [0, 1]$, and produces output velocities $(v_x(t), v_y(t)) = v(x_t, y_t, t)$. The image is fed end-to-end with standard encoder and decoder made up of attention and residual blocks (He et al., 2015). Time t is sinusoidally embedded (Vaswani et al., 2017) and injected into the model along with the embedding of the label y_t via scale-shift normalisation (Dhariwal & Nichol, 2021). Image velocity $v_x(t)$ is taken at the output of the decoder network, which naturally has the same dimension as the input image.

We extract the label velocity $v_y(t)$ from the bottleneck of the U-Net after passing it through an MLP classifier head. The bottleneck encodes the most abstract, class-discriminative features of the image, making it the natural place to extract information about the label. This mirrors the intuition that classification relies on global, low-frequency image structure, which is most concentrated at the coarsest spatial scale.



Figure 3: CIFAR-10 images JFM-generated from the noise x_0 recovered via reverse flow from the original image from the class "cat" (left column) then forward flowed with different label conditioning in each following column.

4.2 Training

We trained U-Net to model the objective described in Section 2.4 using the loss defined in Equation 8 with earning rate of 1×10^{-4}. Images x_1 were standardised to zero mean and unit variance, the labels y_0 one-hot encoded. Noise samples x_0 and y_1 were sampled from the multivariate Gaussian distribution, $x_0 \sim \mathcal{N}(0, \sigma_x^2 I)$ and $y_1 \sim \mathcal{N}(0, \sigma_y^2 I)$, with $\sigma_x = 1.0$ and $\sigma_y = 0.1$.

The best velocity model weights were selected based on validation classification accuracy, evaluated after each epoch using a single 5-step reverse-flow pass over validation data with fixed noise initialisation of $y_1 = \mathbf{0}$.

4.3 Inference

Generation ($t : 0 \to 1$) To generate an image conditioned on label y_0, we sample image noise $x_0 \sim p_{\text{noise}}(X)$ and numerically integrate the ODE from Equation 2 forward from $t = 0$ to $t = 1$ using the Euler method:

$$x_{t+\Delta t} = x_t + \Delta t \cdot v_x^\theta(x_t, y_t, t), \qquad y_{t+\Delta t} = y_t + \Delta t \cdot v_y^\theta(x_t, y_t, t).$$

The output x_1 is the generated image; y_1 is discarded. Conditioning is implicit on the label y_0 which starts as a one-hot encoded class label.

Classification ($t : 1 \to 0$) To classify an image x_1, we initialise $y_1 \sim p_{\text{noise}}(Y)$ and integrate the ODE backward from $t = 1$ to $t = 0$ by the same process as above with negated velocity:

$$x_{t-\Delta t} = x_t - \Delta t \cdot v_x^\theta(x_t, y_t, t), \qquad y_{t-\Delta t} = y_t - \Delta t \cdot v_y^\theta(x_t, y_t, t).$$

The output y_0 is a real-valued score vector over the classes. The predicted class is taken to be $\hat{k} = \text{argmax}_k(y_0)_k$. We estimate classification confidence via Monte Carlo sampling, drawing multiple samples of y_1 with the same x_1, flowing it to y_0 and taking the most frequent predicted label; confidence is its empirical frequency.

5 Results

5.1 The Spiral

We generated this dataset from a bimodal distribution whose modes lie along two interleaving spiral curves. The labels correspond to the modes of the distribution one-hot encoded in $\mathbb{R}^2$. With some

added noise, these form $p_{\text{data}}(X)$ and $p_{\text{data}}(Y)$ respectively. Both standard flow matching and joint flow matching were trained with an MLP velocity model of four hidden layers of 128 SiLU units each. We trained over 200 epochs with the batch size of 1024 and the starting learning rate of 10^{-3}. ODE integration uses the Euler method with 64 function evaluations. The results are shown in Figures 1 and 5.

FM. The flow is trained to transport labels $z_0 \sim p_{\text{data}}(Y)$ to data $z_1 \sim p_{\text{data}}(X)$. For this, FM necessitates that both variables are embedded in the shared space and the labels are infused with some noise (as FM's bijective flow would not be able to map two point labels to more points on the spiral). Each class occupies a mode of a multimodal Gaussian at $t = 0$, and the flow must scatter these clusters onto the correct spiral arm at $t = 1$. Though not typical, FM does allow this. However, the spiral arms are tightly intertwined: straight-line paths from a label cluster to its target arm inevitably pass through the opposite arm. This could be solved with a flow over appropriate manifold, but we assume no such prior knowledge, in which case straight-line paths are preferred by the optimal transport assumptions (Chen & Lipman, 2024). As shown in Figure 1b, the resulting samples populate both arms regardless of the conditioning label: $p_{\text{data}}(X)$ is recovered, but $p_{\text{data}}(X \mid Y)$ is essentially uninformative.

JFM. Each data sample $x_1 \sim p_{\text{data}}(X)$ is paired with noise $x_0 \sim \mathcal{N}(0, I)$ and label sample $y_0 \sim p_{\text{data}}(Y)$ with $y_1 \sim \mathcal{N}(0, 0.1 \cdot I)$ for linear interpolation and trained jointly over $t \sim \mathcal{U}(0, 1)$. Because the label and data variables occupy separate spaces, the label signal is never corrupted by proximity to the data manifold, and remains informative throughout integration even as it is transported toward noise. The flow (in the forward direction) simultaneously routes x_0 to the correct spiral arm and transports y_0 to $p_{\text{noise}}(Y)$, decoupling class conditioning from intra-class variability. As shown in Figure 1c, JFM achieves near-perfect conditioning fidelity: samples conditioned on two class flow to appropriate arms of the spiral. In Figure 5c, the backward pass recovers the correct class label from a data point, confirming that a single model simultaneously performs generation and classification.

5.2 Image Data

We demonstrate implications of JFM's consistency using the CIFAR-10 dataset (Krizhevsky & Hinton, 2009). We use the modified U-Net architecture described in Section 4. Our model was trained on an NVIDIA L40 GPU for 500 epochs with standard data augmentation. We perform classification via reverse ODE integration over 20 steps. Confidence is defined as the empirical frequency of the modal prediction across 10 samples of label noise y_1, and calibration was evaluated using top-label Expected Calibration Error (ECE) with 10 equal-width bins. Resulting test accuracy was 0.907 with ECE of 0.058 (see Appendix C for a sensitivity analysis). On the generative side, using 200-step ODE sampling against the CIFAR-10 training distribution, we obtain an FID-50k score of 33.55. This generative performance is below that of specialised CIFAR-10 generative models, reflecting the trade-off involved in using a shared model for both generation and classification.

Figure 3 shows conditional reconstruction akin to CycleGAN Zhu et al. (2017), which takes an explainability aspect in JFM. We take a test image from CIFAR-10 dataset, and reverse flow with $y_1 = \mathbf{0}$ through JFM-trained U-Net (over 200 steps) to obtain the noise x_0 that specifies the class-independent variation. We then flow forward (over 200 steps) using the same x_0 while changing y_0 to generate respective images of different classes. Inspecting what changes and what stays the same allows us to interpret the class-related and class-independent aspects of an image, as determined by the model. For example, in the second row, the hand is mostly left intact across images of different class, just as is the orange colour in the bottom row. The model clearly does not "see" these aspects as class-related. On the other hand, the switch from uniform to horizon separated background in bottom row indicates that the "ship" and "truck" classification is strongly correlated with background. Consistency of generation and classification in JFM provides natural means of interpreting model's decision-making without post-hoc methods.

Figure 4 illustrates another potential interpretability avenue enabled by JFM. We generated images conditioned on a class label by sampling x_0 and forward flowing through JFM for 100 steps. Each generated image was then reverse flowed and classified 20 times, sampling a different y_1 on each pass to estimate the model's classification confidence. We binned the generated images by con-

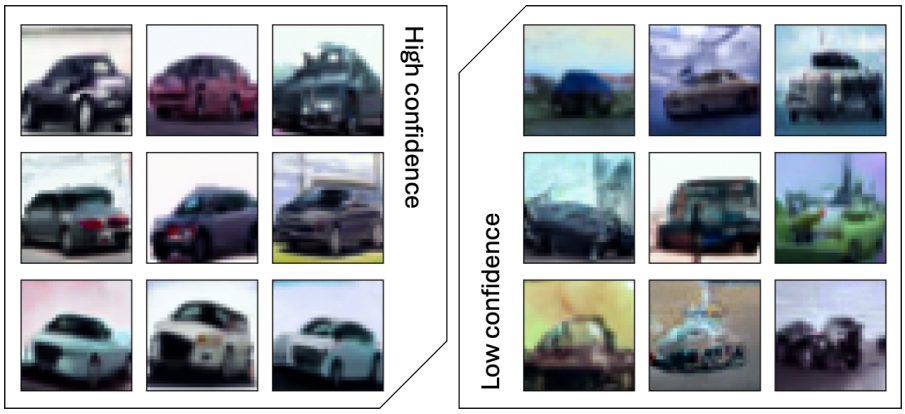


Figure 4: Sample of similar-looking images generated by a CIFAR-10-trained JFM conditioned on the "car" class, grouped by the confidence of the reverse-flow classification. Images on the left are classified as "car" with ≥ 0.9 confidence, while images on the right are classified as "car" with ≤ 0.1 confidence. The consistency property allows classification confidence to be used to examine visual patterns associated with confident and uncertain generations of the flow model.

fidence and inspected them for systematic visual patterns associated with confident and uncertain predictions. In Figure 4, for example, images classified with high confidence as cars consistently exhibit a prominent front and rear light clusters, while lower-confidence examples tend to lack these features. JFM therefore provides a model-driven route to investigating learned behaviour: rather than first formulating a hypothesis about which features the model uses and then designing an experiment to test it, we can generate examples and inspect the patterns that emerge across confidence levels. The model itself brings potential edge cases and discriminative patterns to our attention. While JFM does not guarantee that its internal representation aligns with the semantic meaning of a class label, its generation and classification capabilities provide a direct means of probing the relationship between what the model generates and what it recognises. More examples are provided in Appendix C.

6 Conclusion

Joint Flow Matching (JFM) is a training framework for continuous normalising flows that yields consistent generative and discriminative inference from a single reversible model. By assigning X and Y to opposite temporal endpoints, the endpoint distributions factorise by construction, and the learned flow implicitly defines a joint distribution $p_{\text{JFM}}(X, Y)$. No additional constraints, regularisation, or auxiliary networks are required to enforce this consistency.

Consistency does not guarantee a better underlying generative model. Well-tuned hyperparameters, representative training data, and sufficient training are all still required to produce a useful model under JFM. However, JFM's consistency enables model-driven analysis of its learned behaviour, allowing the model to generate samples and assess its own confidence in them without requiring specific hypotheses about what to investigate.

This work has limitations. The slice-diffeomorphism assumption required for Equation 6 may be violated by pathological datasets. The framework also extends naturally to continuous Y variables (e.g. regression targets or structured outputs), although the behaviour of the reversed conditional $p_{\text{JFM}}(Y|X)$ in such settings has not been characterised. Lastly, the training of a joint generative-discriminative model may be computationally more taxing than doing either separately.

Several extensions naturally follow from the JFM framework. The consistency guarantee holds for any choice of velocity model and noise distribution, so rectification (Liu et al., 2022), optimal-transport couplings (Tong et al., 2023), and Riemannian generalisations (Chen & Lipman, 2024) can all be applied directly. The joint distribution also offers a principled basis for model-internal interpretability: visualising $p_{\text{JFM}}(X|y_0)$ across different label values can expose which regions of the image distribution are associated with each class without post-hoc attribution methods.

AI USE STATEMENT

Large Language Models were used to assist with software implementation, provide initial explanations of flow matching concepts and potential applications, edit and polish manuscript text, and prepare a schematic figure of the model architecture. All AI-assisted code, text, and figures were reviewed and verified by the authors. The authors take full responsibility for the content of the manuscript, experiments, results, and all associated artifacts.

REPRODUCIBILITY STATEMENT

The source code accompanying this work is publicly available at: `https://github.com/4c4a394f/anon-4c4a394f`.

REFERENCES

Francisco Caetano, Christiaan Viviers, Peter H.N. De With, and Fons van der Sommen. Symmetrical flow matching: unified image generation, segmentation, and classification with score-based generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2026.

Ricky T. Q. Chen and Yaron Lipman. Flow Matching on General Geometries. In *International Conference on Learning Representations*, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Kevin Clark and Priyank Jaini. Text-to-image diffusion models are zero shot classifiers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 58921–58937. Curran Associates, Inc., 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Yilun Du and Igor Mordatch. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. GANalyze: Toward Visual Definitions of Cognitive Image Properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5744–5753, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. In *International Conference on Learning Representations*, 2019.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Learning the Stein Discrepancy for Training and Evaluating Energy-Based Models without Sampling. In *International Conference on Machine Learning*, pp. 3732–3747. PMLR, 2020.

Will Grathwohl, Jacob Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No MCMC for me: Amortized sampling for fast and stable training of energy-based models. In *International Conference on Learning Representations*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 770–778. IEEE, 2015.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.

Zahra Kadkhodaie, Florentin Guth, Eero P. Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *International Conference on Learning Representations*, 2024.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. *University of Toronto, ON, Canada*, 2009.

Alexander C. Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. Your diffusion model is secretly a zero-shot classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2206–2217, 2023.

Xiao-Hui Li, Yuhan Shi, Haoyang Li, Wei Bai, Caleb Chen Cao, and Lei Chen. An Experimental Study of Quantitative Evaluations on Saliency Methods. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3200–3208, New York, NY, USA, 2021. Association for Computing Machinery.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conferenceon Learning Representations*, 2022.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Matthew Niedoba, Berend Zwartsenberg, Kevin Murphy, and Frank Wood. Towards a Mechanistic Explanation of Diffusion Model Generalization. In *International Conference on Machine Learning*, 2025.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning*, volume 37, pp. 1530–1538. PMLR, 2015.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034*, 2014.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Diederik P. Kingma. How to Train Your Energy-Based Models. *arXiv preprint arXiv:2101.03288*, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

E. G. Tabak and Cristina V. Turner. A Family of Nonparametric Density Estimation Algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in GANs. In *International Joint Conference on Neural Networks*, 2020.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2023.

John J. Vastola. Generalization through variance: How noise shapes inductive biases in diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Computing Surveys*, 56(4), 2023.

Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision – ECCV 2014*, pp. 818–833, Cham, 2014. Springer International Publishing.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

A Reverse flow in the Spiral Example

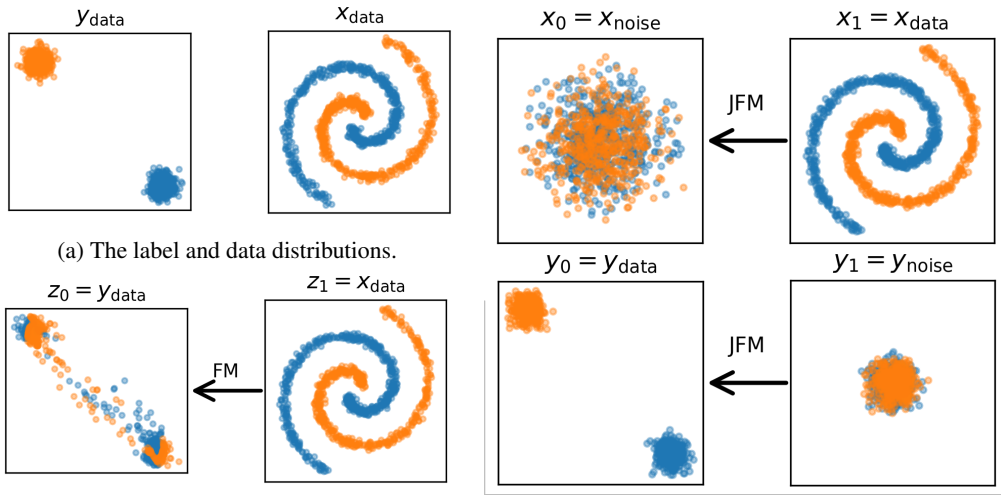
Figure 1 showed the forward/generative flow, from labels y_{data} to the spiral x_{data} of the trained FM and JFM. Figure 5 shows the reverse/discriminative flow from the spiral to the labels. Classification would be determined by which cluster a given point ends up at.

B CIFAR-10 U-Net architecture.

The CIFAR-10 model uses an ADM-style U-Net with input dimensions $3 \times 32 \times 32$ and a base channel width of 128. The four resolution levels use channel multipliers $(1, 2, 2, 4)$, giving channel widths of 128, 256, 256, and 512, respectively, with two residual blocks per level. Downsampling is performed between successive levels, giving a 4×4 bottleneck with 512 channels. Attention is applied at downsampling factors 1 and 2 using four attention heads. Residual blocks use GroupNorm with 32 groups, SiLU activations, scale-shift conditioning, and zero dropout.

The continuous label vector $y_t \in \mathbb{R}^{10}$ is embedded using a two-layer MLP with dimensions $10 \to 512 \to 512$ and combined with the corresponding time embedding. This conditioning is supplied throughout the U-Net. The bottleneck representation, of dimension $512 \times 4 \times 4 = 8192$, is flattened and passed to a classification head consisting of two hidden layers of 512 and 256 units, respectively. Each hidden layer uses LayerNorm and SiLU activation, with zero dropout, followed by a final linear layer producing the 10-dimensional label velocity v_y. The image velocity v_x is produced by the U-Net decoder output. Thus, the label vector y_t conditions the U-Net, while its velocity v_y is predicted from the bottleneck representation.

The model was trained for 500 epochs with batch size 256.



(a) The label and data distributions.

(b) Standard flow matching classification.

(c) Joint flow matching classification.

Figure 5: Reverse flow in FM and JFM trained to model $p(x|y)$ and $p(y|x)$ for the dataset shown in (a). Colour indicates the class conditioning, corresponding to the distinct arms of the spiral. In standard FM, consistent bidirectional conditioning requires x to reverse flow directly to y; this works poorly for the spiral under linear interpolation path (b). JFM has no trouble recovering the correct $p(y|x)$ under the same transport (c).

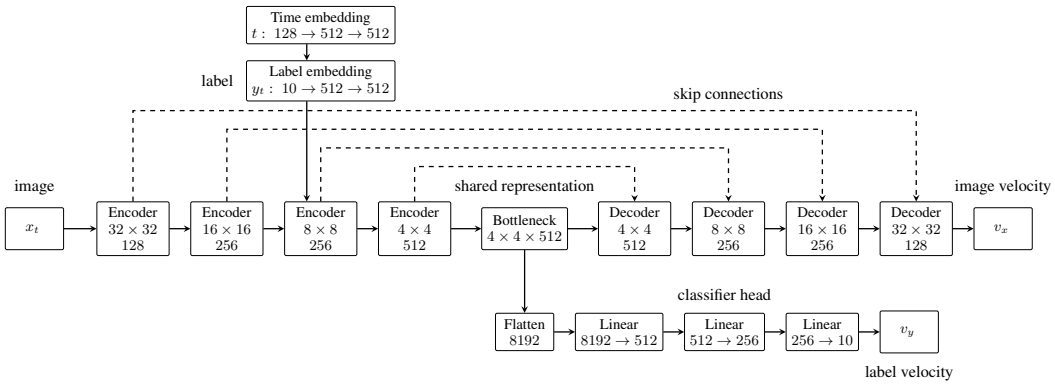


Figure 6: Architecture of the CIFAR-10 JFM model. The noisy image x_t is processed by an ADM-style U-Net with four resolution levels. The continuous label vector y_t and time t are embedded and used to condition the U-Net. The bottleneck representation is used both by the decoder to predict the image velocity v_x and by a separate MLP head to predict the label velocity v_y. The bottleneck has dimensions $4 \times 4 \times 512 = 8192$; the classification head maps this representation through dimensions $8192 \to 512 \to 256 \to 10$.

C CIFAR-10 Results

Figure 7 provides a visualisation of generator-consistent classification on CIFAR-10. The left side shows classification decisions and confidence on a sample of test data by the model that achieved 0.908 test accuracy. The right-hand side shows classification decisions and confidence on a sample of data generated and classified via the forward and reverse flow of the JFM model.

To assess sensitivity to the number of label-noise samples used to estimate classification confidence, we repeated the evaluation with 5, 10, and 100 samples. Accuracy was 0.906, 0.907, and 0.908, with corresponding ECE values of 0.061, 0.058, and 0.055, respectively. The small variation across sample counts indicates that the reported classification and calibration results are not strongly dependent on the number of samples used to estimate confidence.

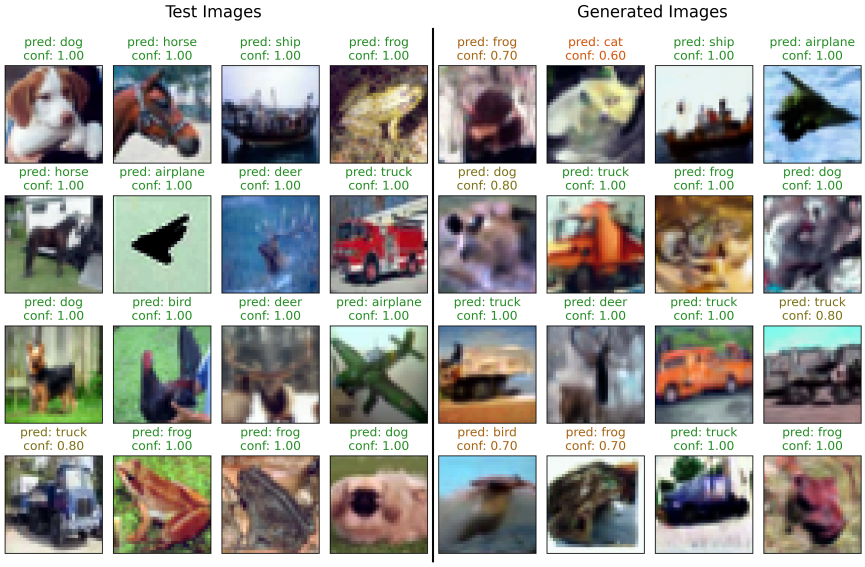


Figure 7: Classification decisions and confidence of the JFM trained on CIFAR-10 data evaluated on test (left side) and generated (right side) data. The colour of the label is colour-maped to green for high and red for low confidence.