RFMI: Estimating Mutual Information on Rectified Flow for Text-to-Image Alignment

Chao Wang[1,2], **Giulio Franzese**[1], **Alessandro Finamore**[2], **Pietro Michiardi**[1]
EURECOM[1], Huawei Technologies SASU, France[2]
[1]{chao.wang, giulio.franzeze, pietro.michiardi}@eurecom.fr
[2]{wang.chao3, alessandro.finamore}@huawei.com

Abstract

Rectified Flow (RF) models trained with a Flow matching framework have achieved state-of-the-art performance on Text-to-Image (T2I) conditional generation. Yet, multiple benchmarks show that synthetic images can still suffer from poor alignment with the prompt, i.e., images show wrong attribute binding, subject positioning, numeracy, etc. While the literature offers many methods to improve T2I alignment, they all consider only Diffusion Models, and require auxiliary datasets, scoring models, and linguistic analysis of the prompt. In this paper we aim to address these gaps. First, we introduce RFMI, a novel Mutual Information (MI) estimator for RF models that uses the pre-trained model itself for the MI estimation. Then, we investigate a self-supervised fine-tuning approach for T2I alignment based on RFMI that does not require auxiliary information other than the pre-trained model itself. Specifically, a fine-tuning set is constructed by selecting synthetic images generated from the pre-trained RF model and having high point-wise MI between images and prompts. Our experiments on MI estimation benchmarks demonstrate the validity of RFMI, and empirical fine-tuning on SD3.5-Medium confirms the effectiveness of RFMI for improving T2I alignment while maintaining image quality.

1 Introduction

Text-to-Image (T2I) generative models have reached an incredible popularity thanks to their high-quality image synthesis, ease of use, and integration across a variety of end-users services (e.g., image editing software, chat bots, smartphones apps, websites). T2I models are trained using large-scale datasets (LAION project, 2024) to generate images *semantically aligned* with a user text input. Yet, recent benchmarks (Huang et al., 2023; Wu et al., 2024) show that even Rectified Flow (RF) models, e.g., Stable Diffusion 3 (Esser et al., 2024) and FLUX (FLUX, 2023), despite achieving a new state of the art, still suffer from a variety of alignment issues (subjects in the images might be missing, or have the wrong attributes, such as numeracy, positioning, etc).

Many works in the literature propose methods to mitigate alignment issues at either inference-time – e.g., steering the generation guided by auxiliary text information or map-based objectives (Chefer et al., 2023; Feng et al., 2023; Shen et al., 2024) – or using model fine-tuning – e.g., updating the pre-trained model weights via supervised learning (Krojer et al., 2023) or reinforcement learning (Fan et al., 2023; Huang et al., 2023) to address a specific task. Despite their merits, available alignment methods require *complementary data* such as linguistic analysis of prompts – e.g., A&E (Chefer et al., 2023) steers U-Net cross-attention units by knowing which are the relevant tokens the prompt that need to appears in the image, auxiliary models – e.g., DPOK (Fan et al., 2023) uses a reward function based on a complementary model trained to capture human judgment of alignment and aesthetics, or datasets – e.g., HNITM (Krojer et al., 2023) relies on a contrastive learning approach creating negative prompt examples from a set of target prompts. In other words, these methods shift the alignment problem from the model to the users.

Differently from existing literature, in this work we argue for avoiding auxiliary input in favor of a *self-supervised approach* where the pre-trained model is used to compute a score signaling if the generated images align with the text prompt. In particular, we aim to use a pre-trained RF model as a neural estimator for the Mutual Information (MI) between the generated image and the prompt. In turn, this raises two research questions: how to estimate MI using an RF model? and how to use the MI estimates to improve T2I alignment?

While multiple MI neural estimators have been proposed, to the best of our knowledge no previous work considers RF models. For instance, discriminative approaches (McAllester & Stratos, 2020) focus on directly estimating the ratio between joint distribution and product of marginals but the sample average's variance scales exponentially with the ground-truth MI. Considering methods closer to our scope, generative approaches based on estimating the two marginal densities separately with generative models like normalizing flows (Anonymous, 2025; Dinh et al., 2017) are more scalable but face estimation accuracy challenges on high-dimensional or complex data according to benchmark testing on synthetic distributions (Czyż et al., 2023). However, none of these methods directly applies to RF models.

More important, the literature on T2I alignment primarily focuses on Diffusion Models (DMs) such as SD2, while RF-related literature only tangentially considers the problem. For instance, (Li et al., 2024) extends SD3 with more modalities, (Dalva et al., 2024) enhances FLUX with a linear and fine-grained editing scheme of models' attention output, (Liu et al., 2024) proposes a novel text-conditioned pipeline to turn Stable Diffusion (SD) into an ultra-fast one-step model – while all these works present CLIP score evaluations, they neither focus nor they are designed to address T2I alignment. At the same time, the intrinsic different nature of the RF models architecture (e.g., SD3 replaces the U-Net of SD2 with a DiT architecture and also add an extra text transformer) calls for redesigning some of the mechanics of DM-based T2I aligment methods.

In summary, in this work:

- We introduce RFMI, an RF-based point-wise MI estimator (Section 3) leveraging the relation between the score of the density $\nabla \log p_t$ and the velocity field u_t.
- We design a self-supervised fine-tuning approach, called RFMI FT (Section 4), that uses a small number of fine-tuning samples to improve the pre-trained T2I model alignment with no inference-time overhead, nor auxiliary models other than the generative model itself.
- We demonstrate the validity of our MI estimator considering both a synthetic benchmark involving various challenging data distributions (Section 5.1) where the true MI is known, as well as a T2I benchmark (Huang et al., 2023) (Section 5.2).

2 Preliminaries

This work relies on recent advances in generative modeling (Esser et al., 2024; FLUX, 2023) as a building block to design an estimator for the mutual information between two random variables. Here, we give the necessary background to develop our methodology, that revolves around flow models (Chen et al., 2019) and flow matching framework (Lipman et al., 2023).

Let $x \in \mathbb{R}^d$ denote a data point in the d-dimensional Euclidean space associated with the standard Euclidean inner product, and $X \in \mathbb{R}^d$ a Random Variable (RV) with continuous Probability Density Function (PDF) $p_X : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, where $\int_{\mathbb{R}^d} p_X(x)\mathrm{d}x = 1$. We use the notation $X \sim p_X$ to indicate that X is distributed according to p_X.

The key concepts we consider within the framework of flow matching are:

1. flow : a time-dependent $C^r\left([0,1] \times \mathbb{R}^d, \mathbb{R}^d\right)$ mapping $\psi \colon (t,x) \mapsto \psi_t(x)$
2. velocity field : a time-dependent $C^r\left([0,1] \times \mathbb{R}^d, \mathbb{R}^d\right)$ mapping $u \colon (t,x) \mapsto u_t(x)$
3. probability path : a time-dependent PDF $(p_t)_{0 \leq t \leq 1}$

Given a source distribution p – e.g., standard Gaussian distribution $\mathcal{N}(0, I)$, and the data target distribution q, the goal of generative flow modeling is to build a flow that transforms $X_0 \sim p$ into $X_1 := \psi_1(X_0)$ such that $X_1 \sim q$.

Considering an arbitrary probability path p_t, u_t is said to *generate* p_t if its flow ψ_t satisfies $X_t := \psi_t(X_0) \sim p_t$ for $t \in [0,1)$, $X_0 \sim p_0$, since there is an equivalence between flows and velocity fields derived from the ordinary differential equation (ODE)

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\psi_t(x) = u_t(\psi_t(x)) & \text{(flow ODE)} \\ \psi_0(x) = x & \text{(flow initial conditions).} \end{cases} \tag{1}$$

One practical way to verify if a vector field u_t generates a probability path p_t is to verify if the pair (u_t, p_t) satisfies the Continuity Equation for $t \in [0,1)$

$$\frac{\mathrm{d}}{\mathrm{d}t}p_t(x) + \operatorname{div}(p_t u_t)(x) = 0. \tag{2}$$

Given a prescribed probability path p_t satisfying the boundary conditions $p_0 = p$ and $p_1 = q$, the goal of Flow Matching (FM) is to learn a parametric velocity field u_t^θ that matches the ground truth velocity field u_t known to generate the desired probability path p_t. This goal is realized by minimizing the regression loss: $\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{X_t \sim p_t} \| u_t^\theta(X_t) - u_t(X_t) \|^2$.

In practice, the ground truth marginal velocity field u_t is not tractable, as it requires marginalizing over the entire training set – i.e., $u_t(x) = \int u_t(x \mid x_1)\, p_{1|t}(x_1 \mid x)\, \mathrm{d}x_1$. Instead, we consider the Conditional Flow Matching (CFM) loss: $\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t, X_1, X_t \sim p_{t|1}(\cdot|X_1)} \left\| u_t^\theta(X_t) - u_t(X_t \mid X_1) \right\|^2$, where the ground truth conditional velocity field $u_t(\cdot \mid x_1)$ is tractable, as it only depends on a single data sample $X_1 = x_1$. The two losses are equivalent for learning purposes: since their gradients coincide, the minimizer of CFM loss is the marginal velocity $u_t(x)$.

Training using CFM loss requires (i) designing $p_{t|1}(\cdot \mid x_1)$ yielding a marginal probability path p_t satisfying the boundary conditions and then (ii) finding $u_t(\cdot \mid x_1)$ generating $p_{t|1}(\cdot \mid x_1)$. These two tasks could be reduced to defining a $C^2\left([0,1) \times \mathbb{R}^d, \mathbb{R}^d\right)$ conditional flow $\psi : [0,1) \times \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\psi_t(x \mid x_1) = \begin{cases} x & t = 0 \\ x_1 & t = 1 \end{cases}$.

A popular objective is to minimize a bound of the Kinetic Energy, which results in the flow

$$\psi_t(x \mid x_1) = t x_1 + (1-t)x. \tag{3}$$

When x is a data sample x_0 of $X_0 \sim p$, eq. (3) becomes $\psi_t(x_0 \mid x_1) = t x_1 + (1-t)x_0$, the conditional velocity field reduces to $u_t(x_t \mid x_1) = x_1 - x_0$ for all $t \in [0,1)$, and the CFM loss simplifies to

$$\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t, (X_0, X_1) \sim \pi_{0,1}} \left\| u_t^\theta(X_t) - (X_1 - X_0) \right\|^2, \tag{4}$$

where $\pi_{0,1}$ denotes the joint distribution known as the source-target coupling. The conditional probability path $p_{t|1}$ of this linear conditional flow, defined using the push-forward formula, satisfies the boundary conditions; furthermore, it is a particular case of Gaussian paths $p_{t|1}(\cdot \mid x_1) = \mathcal{N}(\cdot \mid b_t x_1, a_t^2 I)$ with $b_t = t$ and $a_t = 1 - t$, if the source distribution p is the standard Gaussian.

Since a property of this linear conditional flow is that the Kinetic Energy of the marginal velocity $u_t(x)$ is not bigger than the Kinetic Energy of the original coupling $\pi_{0,1}$ used to train the model, Liu et al. (2022) denote u_t^θ optimized with CFM loss as Rectified Flow (RF).

As we are interested in conditional generation, we define the guidance RV as $Y \sim p_Y$, with data samples $y \in \mathcal{Y} \subset \mathbb{R}^k$. Given access to labeled target samples (x_1, y), the goal of conditional FM is to train the parameters θ of a single velocity field $u_t^\theta : \mathbb{R}^d \times \mathbb{R}^k \to \mathbb{R}^d$ to match the ground truth guided velocity field $u_t(\cdot \mid y)$ known to generate the desired guided probability path $p_{t|Y}(\cdot \mid y)$ satisfying the boundary conditions $p_{t=0|Y}(\cdot \mid y) = p(\cdot)$ and $p_{t=1|Y}(\cdot \mid y) = q(\cdot \mid y)$, for all values of y. The guided version of CFM loss is $\mathcal{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t, (X_0, X_1, Y) \sim \pi_{0,1,Y}} \left\| u_t^\theta(X_t \mid Y) - (X_1 - X_0) \right\|^2$. Furthermore, if the model is trained with Gaussian paths, the Classifier-Free Guidance (CFG) technique can be applied at inference to enhance the sample quality, for which during training y will be masked as null-condition $\emptyset$ with probability p_{uncond}, in order to train $u_t^\theta(\cdot \mid \emptyset)$ to approximate the unconditional velocity field u_t generating the unconditional probability path p_t.

Finally, an essential result known as the Instantaneous Change of Variables (Chen et al., 2019) is required for our work. Given the Continuity Equation, the differential equation governing the evolution of the log-probability density is:

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t\left(\psi_t(x)\right) = -\operatorname{div}\left(u_t\right)\left(\psi_t(x)\right). \tag{5}$$

3 RFMI: Estimating Mutual Information with Rectified Flows

Recall that the MI between two random variables $X \sim p_X$ and $Y \sim p_Y$ can be defined as the Kullback–Leibler (KL) divergence between the joint distribution and the product of marginals: $I(X;Y) = D_{\mathrm{KL}}\left(p_{(X,Y)} \| p_X p_Y\right)$. Furthermore, let $p_{X|Y}(\cdot|y)$ be the conditional distribution of X given $Y = y$. Using the identity $p_{(X,Y)}(x,y) = p_{X|Y}(x \mid y)p_Y(y)$, it clearly holds that $\mathrm{I}(X;Y) = \mathbb{E}_Y\left[D_{\mathrm{KL}}\left(p_{X|Y} \| p_X\right)\right]$, which indicates that the more the distributions $p_{X|Y}$ and p_X differ on average, the greater the information gain.

In this section we introduce RFMI, a new method to estimate the MI between X and the guidance signal Y leveraging conditional RF models (Esser et al., 2024; FLUX, 2023). To keep the notation concise and aligned with the notions in Section 2, we will refer to p_X as q, and $p_{X|Y}(\cdot|y)$ as $q(\cdot|y)$.

Consider the case of linear conditional flow $\psi_t\left(x \mid x_1\right) = tx_1 + (1-t)x$ with x being samples of Gaussian prior $X_0 \sim p = \mathcal{N}(0, I)$. This flow's conditional velocity field $u_t(\cdot \mid x_1)$ generates a Gaussian path $p_{t|1}\left(\cdot \mid x_1\right)$ satisfying $p_{0|1}\left(\cdot \mid x_1\right) = p$ and $p_{1|1}\left(\cdot \mid x_1\right) = \delta_{x_1}(\cdot)$. The conditional pair $\left(u_t(\cdot \mid x_1), p_{t|1}\left(\cdot \mid x_1\right)\right)$ does not depend from the guidance variable Y, while its marginal counterpart does, as $\left(u_t(\cdot \mid y) = \int u_t\left(\cdot \mid x_1\right) p_{1|t,Y}\left(x_1 \mid \cdot, y\right) \mathrm{d}x_1,\ p_{t|Y}(\cdot \mid y) = \int p_{t|1}\left(\cdot \mid x_1\right) q\left(x_1 \mid y\right) \mathrm{d}x_1\right)$. As a consequence, by applying the Marginalization trick, the guided velocity field $u_t(\cdot \mid y)$ generates the guided probability path $p_{t|Y}(\cdot \mid y)$, $p_{t|Y}(\cdot \mid y)$ satisfies $p_{0|Y}(\cdot \mid y) = p(\cdot)$ and $p_{1|Y}(\cdot \mid y) = q\left(\cdot \mid y\right)$. Note that when $y = \emptyset \in \{\emptyset\}$ the marginal case is reduced to unconditional generation: $u_t(\cdot \mid \emptyset) = u_t$, $p_{t|\{\emptyset\}}(\cdot \mid \emptyset) = p_t$, and $q\left(\cdot \mid \emptyset\right) = q$. Overall, we express MI using the guided and the unconditional marginal probability paths at the endpoint $t = 1$ as

$$\begin{aligned}
\mathrm{I}(X;Y) &= \mathbb{E}_Y\left[D_{\mathrm{KL}}\left(p_{X|Y} \| p_X\right)\right] \\
&= \mathbb{E}_Y\left[\int q(x \mid Y) \log\left(\frac{q(x \mid Y)}{q(x)}\right) \mathrm{d}x\right] \\
&= \mathbb{E}_Y\left[\int p_{1|Y}(x_1 \mid Y) \log\left(\frac{p_{1|Y}(x_1 \mid Y)}{p_1(x_1)}\right) \mathrm{d}x_1\right].
\end{aligned} \tag{6}$$

In practice, we train a single conditional RF neural network $u_t^\theta(x \mid y)$, using the CFM loss, for all values $y \in \{\mathcal{Y}, \emptyset\}$. Since the minimizer of CFM loss is $u_t(\cdot \mid y)$, $u_t^\theta(x \mid y)$ is a valid approximation of $u_t(\cdot \mid y)$.

Next, we develop an expression of MI using $u_t(\cdot \mid y)$ and u_t, and use the conditional RF model to estimate the MI between X and Y. To do so, we first express the score functions associated to the marginal probability paths using the marginal velocity fields. These two terms are related according to the following

Proposition 3.1 (Relation between velocity field and score function)**.** *For Gaussian paths $p_{t|1}\left(\cdot \mid x_1\right) = \mathcal{N}\left(\cdot \mid b_t x_1, a_t^2 I\right)$, the relation between the conditional velocity field $u_t(\cdot \mid x_1)$ and the* ***score function*** *$\nabla \log p_{t|1}\left(\cdot \mid x_1\right)$ of the conditional probability path $p_{t|1}(\cdot \mid x_1)$ is derived as :*

$$u_t(x \mid x_1) = \frac{\dot{b}_t}{b_t} x + \left(\dot{b}_t a_t - b_t \dot{a}_t\right) \frac{a_t}{b_t} \nabla \log p_{t|1}\left(x \mid x_1\right). \tag{7}$$

This relation also holds for their marginal counterpart, both in the guided case and in the unconditional case:

$$\begin{cases} u_t(x \mid y) = \dfrac{\dot{b}_t}{b_t} x + \left(\dot{b}_t a_t - b_t \dot{a}_t\right) \dfrac{a_t}{b_t} \nabla \log p_{t|Y}(x \mid y) \\ u_t(x) = \dfrac{\dot{b}_t}{b_t} x + \left(\dot{b}_t a_t - b_t \dot{a}_t\right) \dfrac{a_t}{b_t} \nabla \log p_t(x). \end{cases} \tag{8}$$

See proof in Eq. (F.4) (Albergo & Vanden-Eijnden, 2023), Eq. (7) (Zheng et al., 2023).

In particular, for linear conditional flow with Gaussian prior, i.e. $b_t = t$, $a_t = 1 - t$, we have $\dot{b}_t = 1$, $\dot{a}_t = -1$, and Equation (8) becomes

$$\begin{cases} \nabla \log p_{t|Y}(x \mid y) = \dfrac{t u_t(x \mid y) - x}{1 - t} \\ \nabla \log p_t(x) = \dfrac{t u_t(x) - x}{1 - t}. \end{cases} \tag{9}$$

We note that Equation (9) is only defined for $t \in [0, 1)$. As $t \to 1$, by taking the limit of Equation (9) using l'Hopital's rule, the limit of score function is (proof in Appendix A.1.1):

$$\begin{cases} \lim_{t \to 1} \nabla \log p_{t|Y}(x \mid y) = \lim_{t \to 1} -\partial_t u_t(x \mid y) \\ \lim_{t \to 1} \nabla \log p_t(x) = \lim_{t \to 1} -\partial_t u_t(x). \end{cases} \tag{10}$$

Given the guided and unconditional ground truth marginal velocity fields $u_t(\cdot|y)$ and u_t, it is possible to show that MI can be computed exactly, as done in the following

Proposition 3.2 (MI computation)**.** *Given a linear conditional flow with Gaussian prior, the MI between the target data X and the guidance signal Y is given by*

$$\begin{aligned} \mathrm{I}(X; Y) &= \mathbb{E}_Y \left[\int p_{1|Y}(x_1 \mid Y) \log \left(\frac{p_{1|Y}(x_1 \mid Y)}{p_1(x_1)} \right) \mathrm{d}x_1 \right] \\ &= \mathbb{E}_Y \left[\int_0^1 \mathbb{E}_{X_t|Y} \left[\frac{t}{1 - t} u_t(X_t|Y) \cdot (u_t(X_t|Y) - u_t(X_t)) \right] dt \right]. \end{aligned} \tag{11}$$

This can be proven leveraging Equation (2), Equation (5), and the result of Proposition 3.1. The full proof of Proposition 3.2 is given in Appendix A.1.2.

Similarly, it is easy to show that, given an individual guidance sample $Y = y$, it is possible to use Equation (11) to compute the **point-wise MI** as

$$\mathrm{I}(X; y) = \int_0^1 \mathbb{E}_{X_t|Y=y} \left[\frac{t}{1 - t} u_t(X_t|Y = y) \cdot (u_t(X_t|Y = y) - u_t(X_t)) \right] dt. \tag{12}$$

The integral in Equation (11) can be estimated by uniform sampling $t \sim \mathcal{U}(0, 1)$. However, in practice, since the denominator $(1 - t) \to 0$ as $t \to 1$, this estimator has unbounded variance. To reduce variance, since the argument of the integral has constant magnitude on average, it would be tempting to use importance sampling where $t \sim f(t) \propto \frac{t}{1-t}$. This ratio, however, is hard to normalize (as it integrates to ∞). As an alternative, we consider the following un-normalized density $\tilde{f}_\epsilon$ proportional to such ratio for most of its support, and then truncated to a constant for large t:

$$\tilde{f}_\epsilon(t) = \begin{cases} \frac{t}{1-t} & t \in [0, t_\epsilon) \\ \frac{t_\epsilon}{1-t_\epsilon} & t \in [t_\epsilon, 1] \end{cases} \tag{13}$$

To implement such non-uniform sampling in practice, we use the inverse transform sampling method, with the inverse Cumulative Distribution Function (CDF) described in

Proposition 3.3 (Non-uniform sampling for importance sampling)**.** *The inverse CDF of a PDF proportional to truncated* $\frac{t}{1-t}$ *is*

$$F_\epsilon^{-1}(u) = \begin{cases} 1 + W(-e^{-Zu-1}) & u \in \left[0, \frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}\right), \\ 1 + \frac{1-t_\epsilon}{t_\epsilon}\left[\ln(1-t_\epsilon) + Zu\right] & u \in \left[\frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}, 1\right], \end{cases} \tag{14}$$

in which W *is the Lambert's* W*-function*[1]*, and the normalizing constant is* $Z = -\ln(1-t_\epsilon)$.

We show the proof of Proposition 3.3 in Appendix A.1.3.

Finally, given the parametric approximations of marginal velocity fields through minimization of CFM loss, and the result in Proposition 3.2, now we are able to propose an MI estimator defined as

$$\mathrm{I}(X;Y) \approx \mathbb{E}_Y\left[\int_0^1 \mathbb{E}_{X_t|Y}\left[\frac{t}{1-t} u_t^\theta(X_t|Y) \cdot \left(u_t^\theta(X_t|Y) - u_t^\theta(X_t)\right)\right] dt\right] \tag{15}$$

For the estimation of point-wise MI between generated image and guidance prompt, in practice we found that using the velocity field calibrated by CFG as $u_t^\theta(X_t|Y)$ in Equation (15) leads to better performance than using the vanilla guided output given by the model. Furthermore, for generating images at high resolution, to reduce training's computational cost and to speed up inference, it is common to apply RF on a lower dimensional manifold (e.g. the compressed latent space of a pretrained Variational Autoencoder). It is easy to show that the MI between images X and prompts Y equals to that between images' latents Z and prompts Y, i.e. $I(X;Y) = I(Z;Y)$.

An important property of our estimator is that it is neither an upper nor a lower bound of the true MI, since the difference between the ground truth velocity fields and their parametric approximation can be positive or negative. This property frees our estimation method from the pessimistic results of McAllester & Stratos (2020).

Algorithm 1: RFMI FT

Input : Pre-trained model: $\boldsymbol{u}_\theta$, Prompt set: $\mathcal{Y}$
Hyperparam: Image pool size: M; Top MI-aligned samples: k
Output : Fine-tuned diffusion model $\boldsymbol{u}_{\theta^*}$

// Fine-tuning set
$\mathcal{S} \leftarrow \{\}$
for $\boldsymbol{y}^{(i)}$ *in* $\mathcal{Y}$ **do**
  for j *in* $1, \cdots, M$ **do**
    // Generate and compute MI
    $\boldsymbol{z}^{(j)}$, $\mathrm{I}(\boldsymbol{z}^{(j)}, \boldsymbol{y}^{(i)})$ = `PointWiseMI(`$\boldsymbol{u}_\theta$, $\boldsymbol{y}^{(i)}$`)`
    // Append samples and MI
    $\mathcal{S}[\boldsymbol{y}^{(i)}]$`.append(`$\boldsymbol{z}^{(j)}, I(\boldsymbol{z}^{(j)}, \boldsymbol{y}^{(i)})$`)`
  end
  // Retain only Top-k elements
  $\mathcal{S}[\boldsymbol{y}^{(i)}]$ = `Top-`$k(\mathcal{S}[\boldsymbol{y}^{(i)}])$
end
return $\boldsymbol{u}_{\theta^*}$ = `FineTune(`$\boldsymbol{u}_\theta$, $\mathcal{S}$`)`

Algorithm 2: RFMI

Input : Pre-trained model: $\boldsymbol{u}_\theta$; Prompt: $\boldsymbol{y}$
Hyperparam: Step size: Δt
Output : Generated clean latent: $\boldsymbol{z}$; Point-wise MI: $\mathrm{I}(\boldsymbol{z}, \boldsymbol{y})$

Function `PointWiseMI(`$\boldsymbol{u}_\theta$, $\boldsymbol{y}$`)`:
  // Initial latent sample
  $\boldsymbol{z}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\mathrm{I}(\boldsymbol{z}, \boldsymbol{y}) = 0$
  for t *in* $0, ..., 1$ **do**
    // MI estimation (eq. (12))
    $\mathrm{I}(\boldsymbol{z}, \boldsymbol{y})$ += $\frac{t}{1-t} u_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{y}, t) \cdot (u_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{y}, t) - u_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \emptyset, t))$
    // Denoising step
    $\boldsymbol{z}_t$ += $u_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \boldsymbol{y}, t)\Delta t$
  end
  return $\boldsymbol{z}$, $\mathrm{I}(\boldsymbol{z}, \boldsymbol{y})$

4 Improving Text-Image Alignment with MI-guided Self-Supervised Fine-tuning

Given a pre-trained RF model – e.g., Stable Diffusion 3 (Esser et al., 2024) or Flux.1 (FLUX, 2023), we leverage RFMI to improve the model's alignment via fine-tuning. Specifically, our *self-supervised* approach **relies on the pre-trained model** to create a synthetic fine-tuning set of prompt-image pairs with a high degree of alignment selected using the **point-wise MI estimates obtained from the pre-trained model itself**.

As described in Algorithm 1, we begin with a set of fine-tuning prompts $\mathcal{Y}$ (manually crafted or already available (Huang et al., 2023)) and for each prompt $y^{(i)} \in \mathcal{Y}$, we generate M synthetic images and record their point-wise MI (Equation (12) and Algorithm 2). Then, we rank the image-prompt pairs $(z^{(j)}, y^{(i)})$, $j \in [1, M]$ based on the estimated point-wise

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.lambertw.html

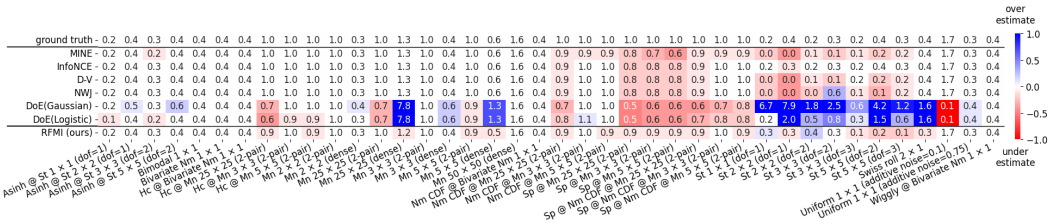

Figure 1: MI estimation results. Color indicates relative negative bias (red) and positive bias (blue).

Table 1: T2I-CompBench alignment results (%) on images generated with CFG=4.5

	Category (BLIP-VQA)	Shape (UniDet)	2D-spatial (UniDet)	3D-spatial (UniDet)[2]	Numeracy (UniDet)
SD3.5-M	57.96	31.31	38.81	61.15	
RFMI FT	61.78	33.92	42.28	64.00	
abs. difference	3.82	2.61	3.47	2.85	
relative gain	6.59	8.34	8.94	4.66	

MI and the top k pairs to the fine-tuning dataset $\mathcal{S}$. Finally, we fine-tune u_θ with efficient LoRA adaptation (Hu et al., 2021). Detailed fine-tuning hyperparameters in Appendix B.

We highlight the efficiency of Algorithm 2 which combines image latent generation and point-wise MI computation. Since MI estimation involves computing an expectation over diffusion times t, it is easy to integrate the MI estimation into the same generation loop. Moreover, the function is easy to parallelize to speed up the fine-tuning set $\mathcal{S}$ composition.

5 Experimental Evaluation

5.1 Synthetic benchmark

As a preliminary step, we assess the quality of RFMI using a known benchmark (Czyż et al., 2023) composed of 40 tasks with synthetic data generated from a variety of known distributions where the true MI is known, and venturing beyond the typical Gaussian distributions to include harder cases (e.g., distributions with high MI or long tails).

We consider four alternative neural estimators as baselines, namely MINE (Belghazi et al., 2021), InfoNCE (van den Oord et al., 2019), NWJ (Nguyen et al., 2010) and DOE (McAllester & Stratos, 2020). All methods are trained/tested using 100k/10k samples, where each sample is composed of two data points x and y concatenated as input for the neural network.

Figure 1 shows the ground truth MI and each method estimates, with colors reflecting the difference between the MI estimate and the true value – the lighter the shade, the smaller the estimation error. Overall, RFMI is on par or better than alternative methods.

5.2 Text-Image Alignment Evaluation

We evaluate RFMI on T2I-CompBench++ (Huang et al., 2023), a T2I benchmark composed of 700/300 (train/test) prompts across 8 categories including attribute binding (color, shape, and texture categories), object relationships (2D-spatial, 3D-spatial, and non-spatial associations), numeracy and complex composition tasks. Prompts are generated with predefined rules or ChatGPT (OpenAI, 2024) and the evaluation uses BLIP-VQA (Huang et al., 2023), UniDet (Zhou et al., 2022), or GPT-4V (Yang et al., 2023) depending on the category. As Huang et al. (2023) recently found, Stable Diffusion 3 already "saturates" performance

[2]For prompts depicting 3D-spatial relation, T2I-CompBench++ leverages UniDet (Zhou et al., 2022) for object detection and Dense Vision Transformer (Ranftl et al., 2021) for depth estimation.

on certain categories (see Table XIII in Huang et al. (2023)): then, we focus our evaluation only on categories having an alignment score lower than 0.7, namely the 4 categories shape, 2D-spatial, 3D-spatial, and numeracy.

We applied RFMI FT (Algorithm 1) on Stable Diffusion 3.5-Medium Esser et al. (2024) (SD3.5-M)[3] and extended Huang et al. benchmark to include this latest RF-based T2I model. Table 1 collects the results, with absolute difference and percentage gain between SD3.5-M and RFMI FT summarized at the bottom. As expected, RFMI FT improves the T2I alignment of SD3.5-M by a sizable margin across all the 4 challenging categories. Qualitative visualization examples are shown in Figure 2.

We highlight that our approach RFMI FT is not sensitive to the neural network architecture or the type of data, so it could be integrated beyond T2I task and into other disciplines where rectified flow is adopted for conditional generation.

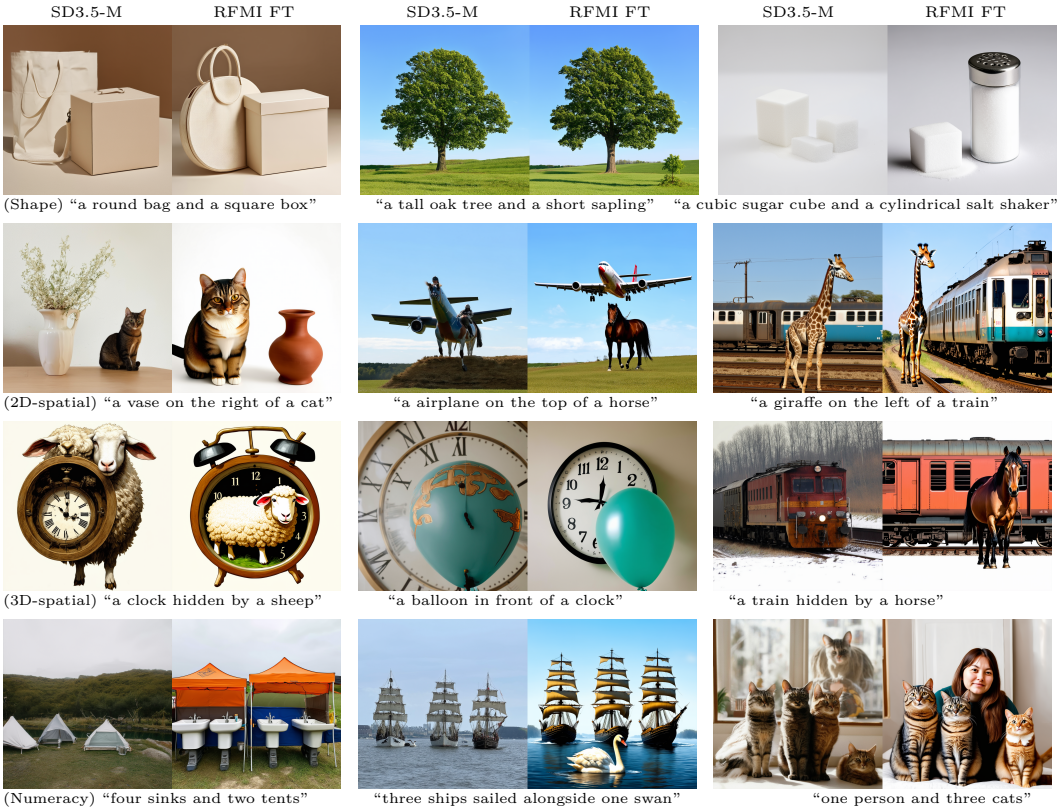

Figure 2: Qualitative examples from Table 1 (same seed used for a given prompt).

6 Conclusion

In this study, we introduced RFMI, a novel RF-based MI estimator which provides a unique perspective on MI estimation by leveraging the theory of FM-based generative models. To show its effectiveness, we first considered a synthetic benchmark where the true MI is known and we showed that RFMI is on par or better than alternative neural estimators. Then, we considered the T2I alignment problem and used RFMI in a self-supervised fine-tuning approach. Specifically, we used the point-wise MI value between text and image estimated by the pre-trained RF model to create a synthetic fine-tuning set for improving the model alignment. Our empirical evaluation on MI estimation benchmark and T2I alignment benchmark illustrated the effectiveness of RFMI. Our lightweight, self-supervised fine-tuning method does not depend on specific model architectures, so it can be used to improve alignment of a variety of RF models in the future.

[3]From a preliminary investigation we observed a $\times 4$ computational costs for FLUX (FLUX, 2023) so we could not collect results in time for the submission.

References

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. URL https://arxiv.org/abs/2209.15571.

Anonymous. Flow-based variational mutual information: Fast and flexible approximations, 2025. URL https://openreview.net/forum?id=spDUv05cEq.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: Mutual information neural estimation, 2021. URL https://arxiv.org/abs/1801.04062.

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.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2019. URL https://arxiv.org/abs/1806.07366.

Paweł Czyż, Frederic Grabowski, Julia E. Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators, 2023. URL https://arxiv.org/abs/2306.11078.

Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow transformers, 2024. URL https://arxiv.org/abs/2412.09611.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017. URL https://arxiv.org/abs/1605.08803.

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, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2403.03206.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=8OTPepXzeh.

Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun Reddy 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*, 2023. URL https://openreview.net/forum?id=PUIqjT4rzq7.

FLUX. Black forest labs. https://github.com/black-forest-labs/flux, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *arXiv preprint arXiv:2307.06350*, 2023.

Benno Krojer, Elinor Poole-Dayan, Vikram Voleti, Christopher Pal, and Siva Reddy. Are diffusion models vision-and-language reasoners? *arXiv preprint arXiv:2305.16397*, 2023.

LAION project. Laion datasets. https://laion.ai/projects/, 2024.

Shufan Li, Konstantinos Kallidromitis, Akash Gokul, Zichun Liao, Yusuke Kato, Kazuki Kozuka, and Aditya Grover. Omniflow: Any-to-any generation with multi-modal rectified flows, 2024. URL https://arxiv.org/abs/2412.01169.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, and qiang liu. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1k4yZbbDqX.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information, 2020. URL https://arxiv.org/abs/1811.04251.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, November 2010. ISSN 1557-9654. doi: 10.1109/tit.2010.2068870. URL http://dx.doi.org/10.1109/TIT.2010.2068870.

OpenAI. Chatgpt (gpt-4) [large language model], 2024. URL https://chat.openai.com/chat.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021. URL https://arxiv.org/abs/2103.13413.

Dazhong Shen, Guanglu Song, Zeyue Xue, Fu-Yun Wang, and Yu Liu. Rethinking the spatial inconsistency in classifier-free diffusion guidance, 2024.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Xindi Wu, Dingli Yu, Yangsibo Huang, Olga Russakovsky, and Sanjeev Arora. Conceptmix: A compositional image generation benchmark with controllable difficulty, 2024. URL https://arxiv.org/abs/2408.14339.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision), 2023. URL https://arxiv.org/abs/2309.17421.

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.

Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Simple multi-dataset detection, 2022. URL https://arxiv.org/abs/2102.13086.

A Appendix

A.1 Proofs

A.1.1 Proof of the limiting case of proposition 3.1

Proposition A.1. *(Limiting case of Prop. 3.1) For linear conditional flow with Gaussian prior, as $t \to 1$, the relation between the marginal velocity field and the score function of the marginal probability path is:*

$$\begin{cases} \lim_{t\to 1} \nabla \log p_{t|Y}(x \mid y) = \lim_{t\to 1} -\partial_t u_t(x \mid y) \\ \lim_{t\to 1} \nabla \log p_t(x) = \lim_{t\to 1} -\partial_t u_t(x) \end{cases} \tag{16}$$

Proof. To simplify notation, we present proof in the *unconditional* setting. Consider the case of conditional flow at the form $\psi_t(x \mid x_1) = a_t x + b_t x_1$, where a_t and b_t are chosen to satisfy $\psi_\iota(x \mid x_1) = \begin{cases} x & t=0 \\ x_1 & t=1 \end{cases}$, x is sample x_0 of RV $X_0 \sim p$, and x_1 is sampled from the target distribution q. The marginalization trick shows that ψ_t generates a p_t satisfying $p_0 = p$ and $p_1 = q$. Using the expression of marginal probability flux (Eq. (14) in Albergo & Vanden-Eijnden (2023)), at $t = 1$, we have:

$$\begin{aligned} j_{t=1}(x) &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \left[\partial_t \psi_t(x_0|x_1)\right]|_{t=1} \delta\left(x - \psi_{t=1}(x_0|x_1)\right) p_0(x_0)\, p_1(x_1)\, dx_0 dx_1 \\ &= \int_{\mathbb{R}^d \times \mathbb{R}^d} \left(\dot{a}_t|_{t=1} x_0 + \dot{b}_t|_{t=1} x_1\right) \delta(x - x_1)\, p_0(x_0)\, p_1(x_1)\, dx_0 dx_1 \\ &= \int_{x_0 \in \mathbb{R}^d} x_0 p_0(x_0)\, dx_0 \int_{x_1 \in \mathbb{R}^d} \dot{a}_t|_{t=1} p_1(x_1)\, \delta(x - x_1)\, dx_1 \quad + \\ &\quad \int_{x_0 \in \mathbb{R}^d} p_0(x_0)\, dx_0 \int_{x_0 \in \mathbb{R}^d} \dot{b}_t|_{t=1} x_1 p_1(x_1)\, \delta(x - x_1)\, dx_1 \\ &= \mathbb{E}[X_0] \dot{a}_1 p_1(x_1) + \dot{b}_1 x_1 p_1(x_1) \end{aligned} \tag{17}$$

It follows that the marginal velocity field at $t = 1$ is

$$\begin{aligned} u_1(x) &= j_1(x)/p_1(x) \\ &= \mathrm{E}[X_0]\dot{a}_1 + \dot{b}_1 x_1 \end{aligned} \tag{18}$$

If the conditional flow is linear, we have $a_t = (1 - t)$ and $b_t = t$, and therefore $\dot{a}_t = -1$, $\dot{b}_t = 1$. Furthermore, if X_0 is the standard Gaussian, we have $\mathrm{E}[X_0] = 0$. This means that Equation (18) becomes $u_1(x) = 0 \times (-1) + 1 \times x_1 = x_1$. Inserting this equality into the numerator in Equation (9), both the denominator and the numerator converge to 0 when $t \to 1$:

$$\begin{cases} t u_t(x) - x \xrightarrow{t\to 1} 1 \times x_1 - x_1 = 0 \\ 1 - t \xrightarrow{t\to 1} 1 - 1 = 0 \end{cases} \tag{19}$$

By applying l'Hôpital's rule,

$$\begin{aligned} &\nabla log p_1(x) \\ &= \lim_{t\to 1} \frac{\partial_t \left(t u_t(x) - x\right) = [\partial_t t] u_t(x) + t[\partial_t u_t(x)] - [\partial_t x] = u_t(x) + t[\partial_t u_t(x)] - u_t(x)}{\partial_t (1 - t) = -1} \\ &= \lim_{t\to 1} [-\partial_t u_t(x)] \end{aligned}$$

It is easy to show that the considerations above also hold in the *guided* case, whereby the marginal items (i.e., the probability flux j_t, flow ψ_t, velocity field u_t, probability path p_t and target distribution q) are expressed in their guided form. □

A.1.2 Proof of Proposition 3.2

Proposition A.2 (MI computation)**.** *Given a linear conditional flow with Gaussian prior, the MI between the target data X and the guidance signal Y is given by*

$$\begin{aligned} \mathrm{I}(X;Y) &= \mathbb{E}_Y\left[\int_{\mathbb{R}^d} p_{1|Y}(x_1 \mid Y)\log\left(\frac{p_{1|Y}(x_1 \mid Y)}{p_1(x_1)}\right)\mathrm{d}x_1\right] \\ &= \mathbb{E}_Y\left[\int_0^1 \mathbb{E}_{X_t|Y}\left[\frac{t}{1-t}u_t(X_t|Y)\cdot(u_t(X_t|Y)-u_t(X_t))\right]dt\right]. \end{aligned} \tag{20}$$

Proof. What needs to be proved in Equation (20) is the equivalence of the terms inside the expectation. To keep notation concise, in the following we will rename the guided pair $\big(p_{t|Y}(\cdot \mid y), u_t(\cdot \mid y)\big)$ as simply $\big(p_t^A(\cdot), u_t^A(\cdot)\big)$, and the marginal pair $(p_t(\cdot), u_t(\cdot))$ as $\big(p_t^B(\cdot), u_t^B(\cdot)\big)$, so what needs to be proved becomes:

$$\int_{\mathbb{R}^d} p_1^A(x_1)\log\left(\frac{p_1^A(x_1)}{p_1^B(x_1)}\right)\mathrm{d}x_1 = \int_0^1 \mathbb{E}_{p_t^A}\left[\frac{t}{1-t}u_t^A(x)\cdot\big(u_t^A(x)-u_t^B(x)\big)\right]dt \tag{21}$$

To prove Equation (21), we start with expanding its LHS:

$$\begin{aligned} &\int_{\mathbb{R}^d} p_1^A(x)\log\left(\frac{p_1^A(x)}{p_1^B(x)}\right)\mathrm{d}x \\ &\overset{(i)}{=} \int_{\mathbb{R}^d} p_0^A(x)\log\left(\frac{p_0^A(x)}{p_0^B(x)}\right)\mathrm{d}x + \int_0^1 \partial_t \int_{\mathbb{R}^d} p_t^A(x)\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\mathrm{d}x dt \\ &\overset{(ii)}{=} 0 + \int_0^1 \partial_t \int_{\mathbb{R}^d} p_t^A(x)\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\mathrm{d}x dt \\ &\overset{(iii)}{=} \int_0^1 \int_{\mathbb{R}^d} \partial_t\left[p_t^A(x)\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x dt \\ &\overset{(iv)}{=} \int_0^1 \left[\underbrace{\int_{\mathbb{R}^d}\left[\partial_t p_t^A(x)\right]\left[\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x}_{①} + \underbrace{\int_{\mathbb{R}^d}\left[p_t^A(x)\right]\left[\partial_t\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x}_{②}\right]dt \end{aligned} \tag{22}$$

where (i) follows from the fundamental theorem of calculus; (ii) follows from the fact that both $p_0^A(x)$ and $p_0^B(x)$ coincide with source distribution p at $t=0$; (iii) follows from switching differentiation (∂_t) and integration ($\int_{\mathbb{R}^d}$) as justified by Leibniz's rule; (iv) follows from using the Product Rule (i.e. $(u\cdot v)' = u'\cdot v + u\cdot v'$) on ∂_t.

About the term ① in Equation (22), using in sequential order (i) the Continuity Equation on $[\partial_t p_t^A(x)]$, (ii) the Product Rule, (iii) the Divergence Theorem, and (iv) the assumption that p_t^A vanishes at infinity gives

$$\begin{aligned} &\int_{\mathbb{R}^d}\left[\partial_t p_t^A(x)\right]\left[\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x \\ &\overset{(i)}{=} \int_{\mathbb{R}^d}\left[-\nabla_x\cdot(p_t^A(x)u_t^A(x))\right]\left[\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x \\ &\overset{(ii)}{=} \int_{\mathbb{R}^d}\big(p_t^A(x)u_t^A(x)\big)\cdot\left[\nabla_x\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x - \int_{\mathbb{R}^d}\nabla_x\cdot\left(p_t^A(x)u_t^A(x)\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right)dx \\ &\overset{(iii)}{=} \int_{\mathbb{R}^d}\big(p_t^A(x)u_t^A(x)\big)\cdot\left[\nabla_x\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right]\mathrm{d}x - \oint_{\partial\mathbb{R}^d}\left(p_t^A(x)u_t^A(x)\log\left(\frac{p_t^A(x)}{p_t^B(x)}\right)\right)\cdot\mathbf{n}dS \\ &\overset{(iv)}{=} \int_{\mathbb{R}^d}\big(p_t^A(x)u_t^A(x)\big)\cdot\big(\nabla_x\log p_t^A(x) - \nabla_x\log p_t^B(x)\big)\mathrm{d}x - 0 \\ &= \mathbb{E}_{p_t^A}\left[u_t^A(x)\cdot\big(\nabla_x\log p_t^A(x) - \nabla_x\log p_t^B(x)\big)\right] \end{aligned} \tag{23}$$

About the term ② in Equation (22), by using in sequence (i) Leibniz's rule and Continuity Equation on $[\partial_t p_t^B(x)]$, and (ii) the equality $\nabla \cdot (\rho \mathbf{v}) = \rho(\nabla \cdot \mathbf{v}) + (\nabla \rho) \cdot v$ if $\mathbf{v}$ is a vector field and ρ is a scalar function, we obtain

$$
\begin{aligned}
&\int_{\mathbb{R}^d} [p_t^A(x)][\partial_t \log \left(\frac{p_t^A(x)}{p_t^B(x)} \right)] \mathrm{d}x \\
=&\int_{\mathbb{R}^d} [p_t^A(x)][\partial_t \log(p_t^A(x)) - \partial_t \log p_t^B(x)\ ] \mathrm{d}x \\
=&\int_{\mathbb{R}^d} [p_t^A(x)][\frac{\partial_t p_t^A(x)}{p_t^A(x)} - \frac{\partial_t p_t^B(x)}{p_t^B(x)}] \mathrm{d}x \\
=&\int_{\mathbb{R}^d} [\partial_t p_t^A(x) - p_t^A(x) \frac{\partial_t p_t^B(x)}{p_t^B(x)}] \mathrm{d}x \\
=&\int_{\mathbb{R}^d} \partial_t p_t^A(x) \mathrm{d}x - \int_{\mathbb{R}^d} \frac{p_t^A(x)}{p_t^B(x)} \partial_t p_t^B(x) \mathrm{d}x \\
\overset{(i)}{=}&\partial_t \int_{\mathbb{R}^d} p_t^A(x) \mathrm{d}x - \int_{\mathbb{R}^d} \frac{p_t^A(x)}{p_t^B(x)} (-\nabla_x \cdot (p_t^B(x) u_t^B(x))] \mathrm{d}x \\
\overset{(ii)}{=}&\partial_t 1 + \int_{\mathbb{R}^d} \frac{p_t^A(x)}{p_t^B(x)} \left(u_t^B(x) \cdot \nabla_x p_t^B(x) + p_t^B(x) \nabla_x \cdot u_t^B(x) \right) \mathrm{d}x \\
=&0 + \int_{\mathbb{R}^d} \frac{p_t^A(x)}{p_t^B(x)} u_t^B(x) \cdot \nabla_x p_t^B(x) + \frac{p_t^A(x)}{p_t^B(x)} p_t^B(x) \nabla_x \cdot u_t^B(x) \mathrm{d}x \\
=&\int_{\mathbb{R}^d} p_t^A(x) \left(\frac{\nabla_x p_t^B(x)}{p_t^B(x)} \cdot u_t^B(x) + \nabla_x \cdot u_t^B(x) \right) \mathrm{d}x \\
=&\int_{\mathbb{R}^d} p_t^A(x) \left(\left(\nabla_x \log p_t^B(x) \right) \cdot u_t^B(x) + \nabla_x \cdot u_t^B(x) \right) \mathrm{d}x \\
\overset{(iii)}{=}&0
\end{aligned}
\tag{24}
$$

in which the last equality (iii) follows from Instantaneous Change of Variables: recall that in Instantaneous Change of Variables (Equation (5)) $\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(x) = -\operatorname{div}(u_t)(x)$, its RHS can be rewritten as $-\operatorname{div}(u_t)(x) = -\nabla_x \cdot u_t(x)$; using the chain rule, its LHS is equivalent to $\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(x) = \frac{d \log p_t(x)}{dx} \cdot \frac{dx}{dt} = (\nabla_x \log p_t(x)) \cdot u_t(x)$. It follows that the Instantaneous Change of Variables yields $\left(\nabla_x \log p_t^B(x) \right) \cdot u_t^B(x) + \nabla_x \cdot u_t^B(x) = 0$.

Finally, (i) injecting Equation (23) and Equation (24) back into Equation (22), and (ii) expressing the score functions with velocity fields (Equation (9)) give:

$$
\begin{aligned}
&\int_{\mathbb{R}^d} p_1^A(x) \log \left(\frac{p_1^A(x)}{p_1^B(x)} \right) \mathrm{d}x \\
\overset{(i)}{=}&\int_0^1 \left[\mathbb{E}_{p_t^A} \left[u_t^A(x) \cdot \left(\nabla_x \log p_t^A(x) - \nabla_x \log p_t^B(x) \right) \right] \right] dt \\
\overset{(ii)}{=}&\int_0^1 \mathbb{E}_{p_t^A} \left[u_t^A(x) \cdot \left[\frac{t u_t^A(x) - x}{1-t} - \frac{t u_t^B(x) - x}{1-t} \right] \right] dt \\
=&\int_0^1 \mathbb{E}_{p_t^A} \left[\frac{t}{1-t} u_t^A(x) \cdot \left(u_t^A(x) - u_t^B(x) \right) \right] dt
\end{aligned}
\tag{25}
$$

i.e. Equation (21) is proven. □

A.1.3 Proof of Proposition 3.3

Proposition A.3 (Non-uniform sampling for importance sampling)**.** *The inverse CDF of a PDF proportional to truncated* $\frac{t}{1-t}$ *is*

$$F_\epsilon^{-1}(u) = \begin{cases} 1 + W(-e^{-Zu-1}) & u \in \left[0, \frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}\right), \\ 1 + \frac{1-t_\epsilon}{t_\epsilon}\left[\ln(1-t_\epsilon) + Zu\right] & u \in \left[\frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}, 1\right], \end{cases} \tag{26}$$

in which W *is the Lambert's* W*-function, and the normalizing constant is* $Z = -\ln(1-t_\epsilon)$*.*

Proof. Our goal is to sample from a PDF proportional to $\frac{t}{1-t}$ for most of its support. For some large $t_\epsilon \in [0,1]$, we define the following un-normalized density (Equation (13))

$$\tilde{f}_\epsilon(t) = \begin{cases} \frac{t}{1-t} & t \in [0, t_\epsilon) \\ \frac{t_\epsilon}{1-t_\epsilon} & t \in [t_\epsilon, 1] \end{cases} \tag{27}$$

By integrating Equation (27) w.r.t. t, we get the cumulative function of the unnormalzied density

$$\tilde{F}_\epsilon(t) = \begin{cases} -\ln(1-t) - t & t \in [0, t_\epsilon) \\ -\ln(1-t_\epsilon) + \frac{t_\epsilon}{1-t_\epsilon}(t-1) & t \in [t_\epsilon, 1] \end{cases} \tag{28}$$

Evaluating it at $t=1$ gives us the normalizing constant $Z = \tilde{F}_\epsilon(t=1) = -\ln(1-t_\epsilon)$, from which we obtain the CDF $F_\epsilon(t) = \frac{\tilde{F}_\epsilon(t)}{Z}$, and the inverse CDF that we need for sampling (using the inverse CDF transform)

$$F_\epsilon^{-1}(u) = \begin{cases} 1 + W(-e^{-Zu-1}) & u \in \left[0, \frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}\right) \\ 1 + \frac{1-t_\epsilon}{t_\epsilon}\left[\ln(1-t_\epsilon) + Zu\right] & u \in \left[\frac{-\ln(1-t_\epsilon)-t_\epsilon}{Z}, 1\right] \end{cases} \tag{29}$$

in which W is the Lambert's W-function.

Specifically, to solve the equation

$$\frac{-\ln(1-t) - t}{Z} = u \tag{30}$$

we denote $w := 1-t$ and $b := -Zu - 1$ for simplicity, then Equation (30) becomes $\ln w = b + w$, which could be rewritten as $-we^{-w} = -e^b$, therefore $-w$ is the Lambert W function $W(-e^b)$. We note that the Lambert W function can be solved because $Z > 0$ and $u \geq 0$ give $-e^b \geq -e^{-1}$. Furthermore, since $-e^b < 0$, both the W_0 and W_{-1} branches of the Lambert W function are defined; but since we are interested in the solution that remains in the range $-1 \leq W(x)$ to make Equation (29) well defined, W_0 is the branch of interest. □

B Experimental protocol details

We report in Table 2 all the hyper-parameters we used for our experiments.

Table 2: Training hyperparameters.

Name	**Value**
Trainable model	MMDiT
PEFT	LoRA
Rank	32
α	32
Learning rate (LR)	$5e-6$
Gradient norm clipping	0.005
LR scheduler	Constant
LR warmup steps	400
Optimizer	AdamW
AdamW - β_1	0.9
AdamW - β_2	0.999
AdamW - weight decay	$1e-4$
AdamW - ϵ	$1e-8$
Resolution	1024×1024
CFG scale	4.5
Denoising steps	100
M	50
k	1
Global batch size	240
Training iterations	2000

To construct a fine-tuning set $\mathcal{S}$ based on point-wise MI, we use the pre-trained SD3.5-M and, given a prompt, conditionally generate 50 images with CFG = 4.5 at resolution 1024×1024, while at the same time computing point-wise MI between the prompt and each image latent. Only the image with the highest MI is kept. This process is done twice for all the 700 fine-tuning prompts $\mathcal{Y}$ defined by T2I-Combench. Given the constructed fine-tuning set, we finetune SD3.5-M for 2000 iterations with LoRA adaptation.

Note that (i) there is no overhead at image generation time: once the pre-trained SD3.5-M has been fine-tuned with RFMI FT, conditional sampling takes the same amount of time of "vanilla" SD3.5-M and (ii) the time to process the workloads scales down (almost linearly) with the number of GPUs used according to our observations.