MEASURING SEMANTIC INFORMATION PRODUCTION IN GENERATIVE DIFFUSION MODELS

Florian Handke
Department of Artificial Intelligence
Radboud University
florian.handke@ru.nl

Félix Koulischer
Department of Information Technology
Ghent University
felix.koulischer@ugent.be

Gabriel Raya
Jheronimus Academy of Data Science
Tilburg University
g.raya@jads.nl

Luca Ambrogioni
Donders Institute for Brain, Cognition and Behaviour
Radboud University
l.ambrogioni@donders.ru.nl

ABSTRACT

It is well known that semantic and structural features of the generated images emerge at different times during the reverse dynamics of diffusion, a phenomenon that has been connected to physical phase transitions in magnets and other materials. In this paper, we introduce a general information-theoretic approach to measure when these class-semantic "decisions" are made during the generative process. By using an online formula for the optimal Bayesian classifier, we estimate the conditional entropy of the class label given the noisy state. We then determine the time intervals corresponding to the highest information transfer between noisy states and class labels using the time derivative of the conditional entropy. We demonstrate our method on one-dimensional Gaussian mixture models and on DDPM models trained on the CIFAR10 dataset. As expected, we find that the semantic information transfer is highest in the intermediate stages of diffusion while vanishing during the final stages. However, we found sizable differences between the entropy rate profiles of different classes, suggesting that different "semantic decisions" are located at different intermediate times.

1 INTRODUCTION

Generative diffusion models (Sohl-Dickstein et al., 2015) operate through two complementary processes: a predefined forward stochastic process, which gradually adds noise to data, and its correspondingly learned reverse process, which can be interpreted as a form of dynamic denoising. In the continuous limit, both processes can be described in terms of differential equations (Song et al., 2021), i.e., diffusion processes. The reverse dynamics of this generative process has been studied extensively in recent years, revealing profound theoretical connections to key concepts in the statistical physics of phase transitions. Specifically, Raya & Ambrogioni (2023) and Ambrogioni (2024) have provided theoretical and empirical evidence for the existence of spontaneous symmetry-breaking phase transitions during generation, which correspond to temporally localized "decision windows" that align with bifurcations in the generative process. In line with their findings, Biroli et al. (2024) have characterized a *speciation* phase during which a sample's general features are determined, which is consistent with the observation that class-specific characteristics emerge at different times during generation (Sclocchi et al., 2024). This phenomenon was further studied mathematically by Li & Chen (2024), who have provided a theoretical framework by which the bounds of this *critical window* can be predicted, yet for a restricted set of initial distributions. From a more applied side, Kynkäänniemi et al. (2024) have demonstrated that restricting classifier-free guidance to an intermediate range of noise levels is essential in preserving both sample diversity and quality, emphasizing the practical importance of a critical interval in conditional generation.

In spite of this growing body of theoretical literature, a practical and scalable method to localize these decision windows in models trained on complex datasets with unknown distributions is still

missing. In this paper, using ideas from information theory, we introduce a distribution-agnostic method to localize the temporal windows corresponding to class-semantic structure generation in trained diffusion models.

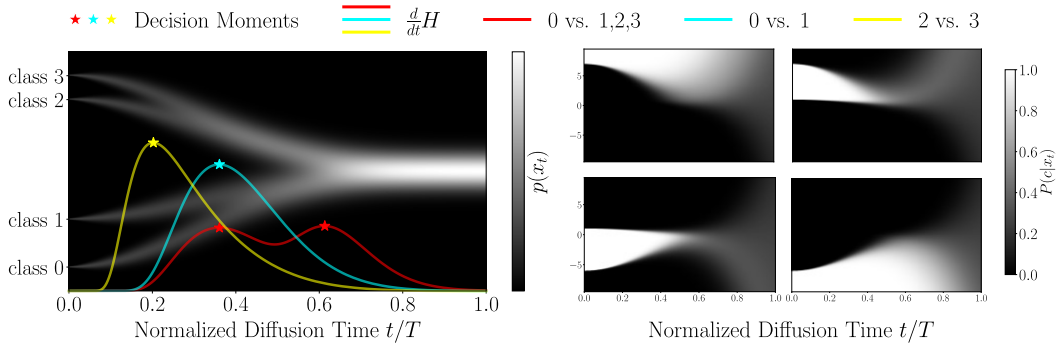

Figure 1: **Generative information transfer in 1D Gaussian-mixture diffusion**.
Left) Diffusion of a one-dimensional equally weighted Gaussian-mixture with four data points (classes) at (-8,-4, 6, 8) along with $\dot{H}(\mathrm{z}|\mathrm{x}_t)$ for different decision problems. **Right**) Posterior evolutions of classes 3 (top left), 2 (top right), 1 (bottom left), 0 (bottom right). The time axes represent the normalized number of noise additions according to a linear noising schedule.

The main idea is to measure the generative information production relative to a defined decision problem by computing the conditional entropy of the accordingly partitioned classes given the noisy states. The importance of a given time point for the generation of a given class can then be assessed through the corresponding entropy production (i.e., the temporal derivative of the conditional entropy) which quantifies how many bits of information have been transferred by the denoising process relative to the defined partition. w

2 Generative Diffusion from an Information Theoretical Perspective

The amount of information retained in a data source can be quantified by the entropy and is typically measured in bits (Shannon, 1948):

$$H(\mathrm{y}) = -\sum_{j=1}^{n} P(y_j) \log_2 P(y_j)$$

where y_j are the values of a random variable y defined on vocabulary $\mathcal{X}$ with n elements and $P(\mathbf{y})$ is its probability mass function. In generative diffusion, the marginal entropy $H(\mathrm{x}_t)$ can be used for addressing temporal questions, such as the generative information potential of a model at a specific point in time. However, it does not quantify the semantic content of the information that is restored. By shifting the focus to the class label random variable c, conditional to the state random variable x_t, we can derive a formulation that quantifies semantic information.

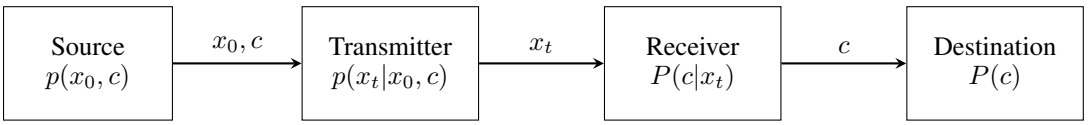

Figure 2: **Transmitting class-semantic information in generative diffusion**.
Information transmission in generative diffusion from the perspective of the class random variable c. In contrast to the conventional description, the receiver $p(x_0|x_t)$ is exchanged with a classifier $P(c|x_t)$, and the destination $p(x_0)$ is replaced by $P(c)$.

As illustrated in Figure 2, the class generation process can be conceptualized as a form of information transfer through a noisy channel given by the forward process. An image x_0 is sampled from the dataset and then transmitted through a noisy channel $x_0 \rightarrow x_T$ determined by the forward process, where a decoder recovers information concerning its class label. The optimal probabilistic

decoder can be obtained using Bayes rule $P(c_k|x_t) = \frac{p(x_t|c_k)P(c_k)}{\sum_{j=1}^{n} p(x_t|c_j)P(c_j)}$. The uncertainty in the class label random variable c given a noisy state random variable x_t may then be quantified through the conditional entropy

$$H(\mathrm{c}|\mathrm{x}_t) = -\int p(x_t) \sum_{k=1}^{K} P(c_k|x_t) \log_2 P(c_k|x_t)\, dx_t \tag{1}$$

We can thus express the total amount of information transmitted up to time t as $T_t = H(\mathrm{c}) - H(\mathrm{c}|\mathrm{x}_t)$. This *information transfer* captures the amount of information that has been gained by measuring the noisy state up to time t. Note that this information is fully contained in the current state x_t since in a Markov forward process we have $P(c|x_{t:T}) = P(c|x_t)$. The specific contribution of the (infinitesimal) noise added at time t can be quantified by the entropy rate $\dot{H}(\mathrm{c}|\mathrm{x}_t)$, which is the temporal derivative of the conditional entropy.

3 THE CONDITIONAL ENTROPY

The conditional entropy, as defined in Eq. 1, encompasses the entire set of possible classes. We introduce a binary variable $\mathrm{z} \in \{z_0, z_1\}$ to partition this set leading to the following definition.

$$H(\mathrm{z}|\mathrm{x}_t) = -\int p_{\mathrm{z}}(x_t) \sum_{z} P(z|x_t) \log_2 P(z|x_t) dx_t \tag{2}$$

Here, $p_{\mathrm{z}}(x_t)$ denotes the mixture distribution under the partition induced by z. The prior $P(z)$ can be computed from the corresponding class-specific priors. When equal to 0.5, Eq. 2 becomes proportional to the negative Jensen-Shannon Divergence between the mixture components p_{z_0} and p_{z_1} (Appendix A). Eq. 2 measures the uncertainty left in the decision defined by z given x_t. When defining z_0 to represent one specific class and z_1 to represent everything else, we can measure class-specific generative information transfer. Figure 1 shows the temporal derivative of Eq. 2 for different decision problems. We can see that the maxima of the temporal derivatives align with the bifurcations in the diffusion process of the branches that are involved in the respective decisions. For Gaussian-mixture models, the conditional entropy can be approximated readily as the joint distributions $p(x_t, c)$ are Gaussian, with diffused class-specific mean and variance μ_{kt}, σ_{kt}^2, weighted by a class-specific prior π_k (Appendix B).

3.1 ESTIMATING THE ENTROPY IN TRAINED MODELS

To estimate the entropy in trained models, we use the algorithm proposed by Koulischer et al. (2025) which estimates the class-specific posterior $P(c|x_t)$ by tracking the Markov chain of the forward process backward in time, i.e., $p(x_{T:t}) = p(x_T) \prod_{\tau=T}^{t+1} p(x_{\tau-1}|x_\tau)$. Using the data processing inequality which directly follows from the Markovian structure of the forward process, they show that the likelihood of a specific class depends only on the least noisy state, i.e. $P(c|x_{T:t}) = P(c|x_t)$. Thus, the posterior can be estimated iteratively for the current denoising trajectory:

$$\begin{aligned} \log P(c|x_t) &= \log P(c|x_{t+1}) + \big(\log p(x_t|x_{t+1}, c) - \log p(x_t|x_{t+1})\big) \\ &= \log P(c|x_{t+1}) - \frac{1}{2\sigma_t^2}\big(||x_t - \mu_\theta(x_{t+1}; c)||^2 - ||x_t - \mu_\theta(x_{t+1})||^2\big) \end{aligned} \tag{3}$$

The means, $\mu_\theta(x_{t+1}; c)$ and $\mu_\theta(x_{t+1})$ can be obtained from a model's conditional and unconditional noise predictions respectively. In its present form, Eq. 3 approximates the posterior ratio between a specific class and "everything" which is encoded by the unconditional model, also referred to as *null*-model $p_\theta(x_t|\emptyset)$ in classifier-free guidance literature (Ho & Salimans, 2022). We redefine Eq. 3 in terms of the binary variable z. When partitioning the set of classes into a specific class versus everything else, Eq. 3 yields both, $P(z_0|x_t)$ and $P(z_1|x_t) = 1 - P(z_0|x_t)$. We sample from the mixture distribution $p_{\mathrm{z}}(x_t)$ by splitting the expectation in Eq. 2 into its mixture components and then using the learned scores, $\nabla_{x_t} \log p_\theta(x_t|z_0)$ and $\nabla_{x_t} p_\theta(x_t|z_1)\big)$. When comparing one class with its complement, we choose to approximate $\nabla \log p(x_t|\neg c)$ by the *null*-model, i.e., $\nabla_{x_t} \log p_\theta(x_t|\emptyset)$ as it allows us to estimate Eq. 2 using two forward passes through a trained model for every sample x_t. The larger the set of classes, the better this approximation becomes. By exchanging the *null*-model with a second class-specific model, we can directly compare two classes. A more detailed description can be found in Appendix C.

4 GENERATIVE INFORMATION TRANSFER IN CIFAR10

Figure 3 shows the estimated temporal derivatives of the conditional entropy for the decision problems just described for a model trained on CIFAR10 (Appendix C.3). Here, we use "deer" as the target class (z_0) and "airplane", "bird", "cat", "car", and "not deer" as the counterparts (z_1) respectively.

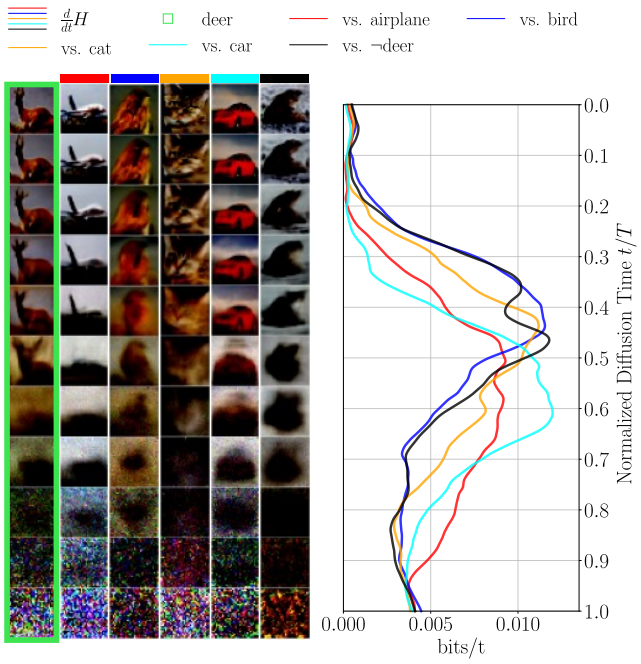

Figure 3: **Generative information transfer in CIFAR10**.
Left) Expected noiseless images throughout the generative process for the five example classes "deer", "airplane", "bird", "cat", "car", and the "null" class which we use to approximate "not deer". **Right**) Approximated temporal derivatives for the binary decisions between "deer" and "airplane", "bird", "cat", "car", and "not deer", respectively.

The temporal derivatives of the conditional entropy of decision problems involving contrasting classes, e.g., "deer" versus "car," which supposedly share minimal structural information, peak at an earlier stage in the generative process, suggesting a preliminary divergence of the respective diffusion branches, similar to the bifurcations illustrated in Figure 1. However, we also observe that across experiments, including those in Appendix C, most of the information transfer occurs within a central interval that is preceded by a data-mean convergence, emphasizing the existence of different diffusion regimes and a *critical window* (Raya & Ambrogioni, 2023; Ambrogioni, 2024; Li & Chen, 2024; Biroli et al., 2024).

5 CONCLUSION & BROADER IMPACT

Although we have outlined a comprehensive method for measuring class-semantic information transfer in generative diffusion models, it still remains to be extensively validated on models trained on more complex class-conditional datasets, such as ImageNet, and those trained on datasets with compositional class hierarchies, such as LAION. In addition, we will attempt to derive an analytical connection between the conditional entropy and the critical decision points corresponding to the first-order phase transitions described by Raya & Ambrogioni (2023). Nevertheless, even in its current form, our approach provides an alternative view of generative diffusion that facilitates our theoretical understanding of the topic. Additionally, it offers a theoretically sound method for performing guided conditional generation in a semi-dynamic fashion that could be the starting point to extend the ideas from Kynkäänniemi et al. (2024) and improve their practical applicability.

REFERENCES

Luca Ambrogioni. The statistical thermodynamics of generative diffusion models: Phase transitions, symmetry breaking and critical instability, 2024. URL https://arxiv.org/abs/2310.17467.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1), November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3. URL http://dx.doi.org/10.1038/s41467-024-54281-3.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Felix Koulischer, Johannes Deleu, Gabriel Raya, Thomas Demeester, and Luca Ambrogioni. Dynamic negative guidance of diffusion models, 2025. URL https://arxiv.org/abs/2410.14398.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models, 2024. URL https://arxiv.org/abs/2404.07724.

Marvin Li and Sitan Chen. Critical windows: non-asymptotic theory for feature emergence in diffusion models, 2024. URL https://arxiv.org/abs/2403.01633.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models, 2023. URL https://arxiv.org/abs/2305.19693.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data, 2024. URL https://arxiv.org/abs/2402.16991.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. URL https://arxiv.org/abs/1503.03585.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL https://arxiv.org/abs/2011.13456.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

A JENSEN-SHANNON DIVERGENCE AND CONDITIONAL ENTROPY

The Jensen-Shannon Divergence (JSD) can be defined using Kullback-Leibler Divergences between two distributions p, q and their mixture $m = \frac{p+q}{2}$. With $p = p(x_t|z_0)$ and $q = p(x_t|z_1)$ JSD can accordingly be written as:

$$\begin{aligned}
JSD(p(x_t|z_0)||p(x_t|z_1)) &= \frac{1}{2}\big(D_{KL}\big(p(x_t|z_0)||m\big) + D_{KL}\big(p(x_t|z_1)||m\big)\big) \\
&= \frac{1}{2}\int p(x_t|z_0)\log\left(\frac{2\cdot p(x_t|z_0)}{p(x_t|z_0)+p(x_t|z_1)}\right) \\
&\quad + p(x_t|z_1)\log\left(\frac{2\cdot p(x_t|z_1)}{p(x_t|z_0)+p(x_t|z_1)}\right) dx_t \\
&= \frac{1}{2}\int p(x_t|z_0)\log\left(\frac{p(x_t|z_0)}{p(x_t|z_0)+p(x_t|z_1)}\right) \\
&\quad + p(x_t|z_1)\log\left(\frac{p(x_t|z_1)}{p(x_t|z_0)+p(x_t|z_1)}\right) dx_t + \log 2
\end{aligned}$$

Now, $\log(\frac{2p}{m})$ can be written in terms of the posterior and prior:

$$\log\left(\frac{p(x_t|z_0)}{p(x_t|z_0)+p(x_t|z_1)}\right) = \log\left(\frac{P(z_0|x_t)/P(z_0)}{P(z_0|x_t)/P(z_0)+P(z_1|x_t)/P(z_1)}\right)$$

which we will denote as $\log p$ for convenience from now on as well as $\log 2 := c$:

$$\begin{aligned}
JSD(p(x_t|z_0)||p(x_t|z_1)) &= \frac{1}{2}\int p(x_t|z_0)\cdot\log p + p(x_t|z_1)\cdot\log(1-p)\, dx_t + c \\
&= \frac{1}{2}\int \frac{P(z_0|x_t)p(x_t)}{P(z_0)}\cdot\log p + p(x_t|z_1)\cdot\log(1-p)\, dx_t + c \\
&= \frac{1}{2}\int \frac{P(z_0|x_t)/P(z_0)+P(z_1|x_t)/P(z_1)}{P(z_0|x_t)/P(z_0)} \\
&\quad + P(z_1|x_t)/P(z_1)\cdot\frac{P(z_0|x_t)}{P(z_0)}p(x_t)\cdot\log p \\
&\quad + p(x_t|z_1)\cdot\log(1-p)\, dx_t + c \\
&= \frac{1}{2}\int \left[\frac{P(z_0|x_t)p(x_t)}{P(z_0)} + \frac{P(z_1|x_t)p(x_t)}{P(z_1)}\right]\cdot p\log p \\
&\quad + p(x_t|z_1)\cdot\log(1-p)\, dx_t + c \\
&= \frac{1}{2}\int \Big[p(x_t|z_0)+p(x_t|z_1)\Big]\cdot p\log p + p(x_t|z_1)\cdot\log(1-p)\, dx_t + c
\end{aligned}$$

Likewise, one can transform the right summand:

$$p(x_t|z_1)\cdot\log(1-p) = \Big[p(x_t|z_0)+p(x_t|z_1)\Big](1-p)\log(1-p)$$

The final result then yields:

$$\begin{aligned}
JSD(p(x_t|z_0)||p(x_t|z_1)) &= \frac{1}{2}\int \Big[p(x_t|z_0)+p(x_t|z_1)\Big]\cdot p\log p \\
&\quad + \Big[p(x_t|z_0)+p(x_t|z_1)\Big]\cdot(1-p)\log(1-p)\, dx_t + c \\
&= \frac{1}{2}\int p(x_t|z_0)\cdot\Big[p\log p + (1-p)\log(1-p)\Big]\, dx_t \\
&\quad + \frac{1}{2}\int p(x_t|z_1)\cdot\Big[p\log p + (1-p)\log(1-p)\Big]\, dx_t + c \\
&= \int \frac{p(x_t|z_0)+p(x_t|z_1)}{2}\cdot\Big[p\log p + (1-p)\log(1-p)\Big] dx_t + c
\end{aligned}$$

B GAUSSIAN MIXTURE DIFFUSION

B.1 DERIVING THE MARGINALS, LIKELIHOODS, AND POSTERIORS

When diffusing a Gaussian-mixture model (GMM) the intermediate distributions are available in closed-form. We will derive them in the following. Starting from the definition of a GMM

$$p(x_0) = \sum_k^K \pi_k \mathcal{N}(x_0|\mu_k, \sigma_k^2)$$

and applying a variance-preserving noise addition with known β_t, the diffusion kernel is defined as:

$$p(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\alpha_t}x_0, 1-\alpha_t) \text{ with } \alpha_t = \prod_{\tau=1}^{t}(1-\beta_t)$$

Accordingly, the marginals are given by:

$$p(x_t) = \sum_k^K \pi_k \int \mathcal{N}(x_t|\sqrt{\alpha_t}x_0, 1-\alpha_t) \cdot \mathcal{N}(x_0|\mu_k, \sigma_k^2)\, dx_0$$

The integral resolves to another Gaussian:

$$I = \int \mathcal{N}(x_t|\sqrt{\alpha_t}x_0, 1-\alpha_t) \cdot \mathcal{N}(x_0|\mu_k, \sigma_k^2)\, dx_0 = \mathcal{N}(x_t|\mu_{kt}, \sigma_{kt}^2)$$

With μ_{kt} and σ_{kt}^2 derived in the following:

$$\begin{aligned}
I &= \frac{1}{\sqrt{2\pi\sigma_k^2}} \frac{1}{\sqrt{2\pi(1-\alpha_t)}} \int \exp\Big(E\Big) dx_0 \\
E &= -\frac{1}{2\sigma_k^2}(x_0-\mu_k)^2 - \frac{1}{2(1-\alpha_t)}(x_t - \sqrt{\alpha_t}x_0)^2 \\
&= -\frac{x_0^2}{2\sigma_k^2} + \frac{x_0\mu_k}{\sigma_k^2} - \frac{\mu_k^2}{2\sigma_k^2} - \frac{x_t^2}{2(1-\alpha_t)} + \frac{\sqrt{\alpha_t}x_0}{1-\alpha_t} - \frac{\alpha_t x_0^2}{2(1-\alpha_t)} \\
&= -\frac{1}{2}\Big(\frac{1}{\sigma_k^2} + \frac{\alpha_t}{1-\alpha_t}\Big)x_0^2 + \Big(\frac{\mu_k}{\sigma_k^2} + \frac{x_t\sqrt{\alpha_t}}{1-\alpha_t}\Big)x_0 - \frac{\mu_k^2}{2\sigma_k^2} - \frac{x_t^2}{2(1-\alpha_t)} \\
&= -\frac{A}{2}x_0^2 + Bx_0 + C = -\frac{A}{2}(x_0 - \frac{B}{A}) + \frac{B^2}{2A} + C \\
&\Rightarrow \\
I &= \frac{1}{\sqrt{2\pi\sigma_k^2}} \frac{1}{\sqrt{2\pi(1-\alpha_t)}} \exp\Big(\frac{B^2}{2A} + C\Big) \int \exp\Big(-\frac{A}{2}(x_0 - \frac{B}{A})^2\Big) dx_0 \\
&= \frac{1}{\sqrt{2\pi\sigma_k^2}} \frac{1}{\sqrt{2\pi(1-\alpha_t)}} \exp\Big(\frac{B^2}{2A} + C\Big) \frac{\sqrt{2\pi}}{\sqrt{A}} \\
&= \frac{1}{\sqrt{2\pi\sigma_k^2(1-\alpha_t)A}} \exp\Big(\frac{B^2}{2A} + C\Big)
\end{aligned}$$

We can directly read off σ_{kt}^2:

$$\sigma_{kt}^2 = \sigma_k^2(1-\alpha_t)A = \alpha_t\sigma_k^2 + (1-\alpha_t)$$

Likewise, we recognize that

$$\begin{aligned}
\frac{B^2}{A} + C &= -\frac{1}{2\sigma_{kt}^2}x_t^2 - \frac{\mu_{kt}}{\sigma_{kt}^2}x_t + \frac{1}{2\sigma_{kt}^2}c \\
\frac{\mu_{kt}}{\sigma_{kt}^2}x_t &= \frac{2(\mu_k\sqrt{\alpha_t})/(\sigma_k^2(1-\alpha_t))}{2((1-\alpha_t)+\alpha_t\sigma_k^2)/(\sigma_k^2(1-\alpha_t))}x_t = \frac{\mu_k\sqrt{\alpha_t}}{\sigma_{kt}^2}x_t
\end{aligned}$$

Again, we can read off μ_{kt}:

$$\mu_{kt} = \sqrt{\alpha_t}\mu_k$$

The likelihood, $p(x_t|c)$ is then simply given by $\mathcal{N}(x_t|\mu_{kt}, \sigma_{kt}^2)$. The only term left is the posterior $P(c|x_t)$ that can be derived from the previous definitions and Bayes' formula:

$$P(c_k|x_t) = \frac{p(x_t|c_k)P(c_k)}{p(x_t)} = \frac{\pi_k \mathcal{N}(x_t|\mu_{kt}, \sigma_{kt}^2)}{\sum_j^K \pi_j \mathcal{N}(x_t|\mu_{jt}, \sigma_{jt}^2)}$$

B.2 Estimating the Conditional Entropy

Recall the definition of the conditional entropy defined under the binary decision z:

$$H(\mathrm{z}|\mathrm{x}_t) = -\int p_{\mathrm{z}}(x_t) \sum_z P(z|x_t) \log P(z|x_t) dx_t$$

Consider the decision between two specific classes, $z_0 \to c_1$, and $z_1 \to c_2$ that are part of a larger set $\mathrm{c} = \{c_1, c_2, c_3, c_4\}$. We can then write $p_{\mathrm{z}}(x_t)$

$$p_{\mathrm{z}}(x_t) = p(x_t|z_0)P(z_0) + p(x_t|z_1)P(z_1)$$

where the priors are defined relative to the decision problem, i.e., $P(z_0) = P(c_1)/(P(c_1) + P(c_2))$ and $P(z_1) = P(c_2)/(P(c_1) + P(c_2))$. Similarly, the posteriors are defined as:

$$P(z_0|x_t) = \frac{P(c_1|x_t)}{P(c_1|x_t) + P(c_2|x_t)}$$
$$P(z_1|x_t) = \frac{P(c_2|x_t)}{P(c_1|x_t) + P(c_2|x_t)}$$

The likelihoods are given by $p(x_t|z_0) = p(x_t|c_1)$ and $p(x_t|z_1) = p(x_t|c_2)$. Thus the conditional entropy for the given decision problem becomes:

$$\begin{aligned} H(\mathrm{z}|\mathrm{x}_t) = &-\int \left[p(x_t|c_1) \frac{P(c_1)}{P(c_1) + P(c_2)} + p(x_t|c_2) \frac{P(c_2)}{P(c_1) + P(c_2)} \right] \cdot \\ &\cdot \sum_j^2 \frac{P(c_j|x_t)}{P(c_1|x_t) + P(c_2|x_t)} \log \left(\frac{P(c_j|x_t)}{P(c_1|x_t) + P(c_2|x_t)} \right) dx_t \\ = &-\sum_i^2 \frac{P(c_i)}{P(c_1) + P(c_2)} \int p(x_t|c_i) \sum_j^2 \frac{P(c_j|x_t)}{P(c_1|x_t) + P(c_2|x_t)} \log \left(\frac{P(c_j|x_t)}{P(c_1|x_t) + P(c_2|x_t)} \right) dx_t \end{aligned}$$

The likelihoods $p(x_t|c_i)$ and the posteriors $P(c_j|x_t)$ were derived in the previous section and are given by:

$$p(x_t|c_i) = \mathcal{N}(x_t|\mu_{it}, \sigma_{it}^2)$$
$$P(c_j|x_t) = \frac{\pi_j \mathcal{N}(x_t|\mu_{jt}, \sigma_{jt}^2)}{\pi_1 \mathcal{N}(x_t|\mu_{1t}, \sigma_{1t}^2) + \pi_2 \mathcal{N}(x_t|\mu_{2t}, \sigma_{2t}^2)}$$

The priors are simply $P(c_i) = \frac{\pi_i}{\pi_1 + \pi_2}$. We estimate the integral using a discretization for x_t and then taking the Riemann sum.

B.3 Additional Examples: Diffusion and Fixed-point Dynamics

Additional setups for symmetric and asymmetric GMMs, equally weighted as well as differently weighted. Additionally, we show the corresponding fixed point dynamics that were derived by optimizing $-\nabla_{x_t} \log p(x_t) + 0.5x_t$ using Powell's hybrid method (Newton-Raphson + steepest descent).

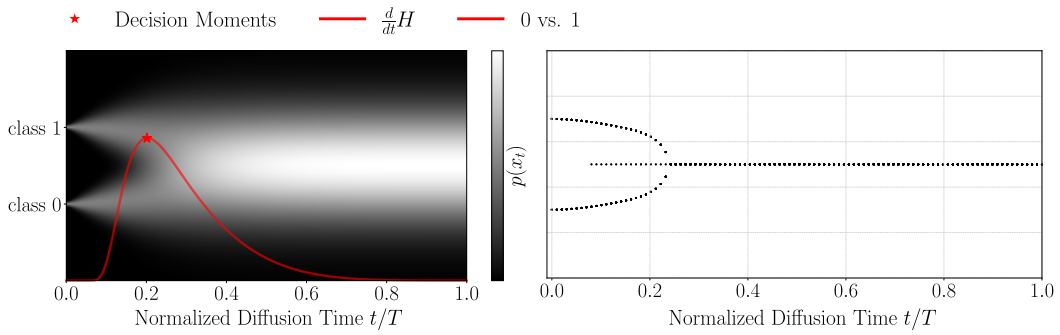

Figure 4: GMM with mixture weights $(\pi_0, \pi_1) = (0.5, 0.5)$ and deltas at $(\mu_0, \mu_1) = (-1, 1)$.

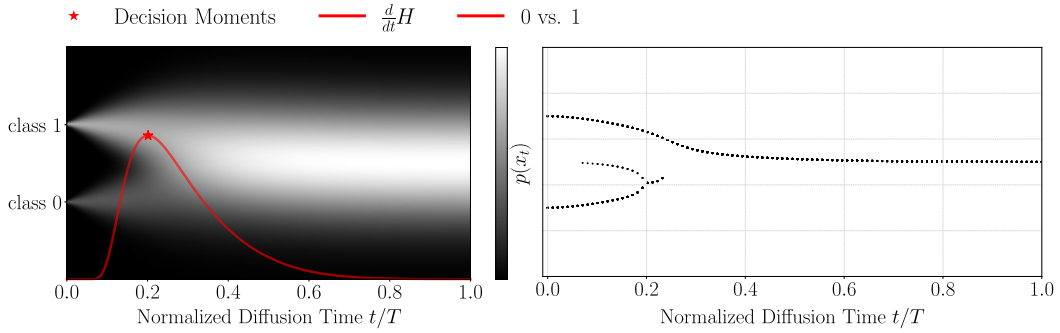

Figure 5: GMM with mixture weights $(\pi_0, \pi_1) = (1/3, 2/3)$ and deltas at $(\mu_0, \mu_1) = (-1, 1)$.

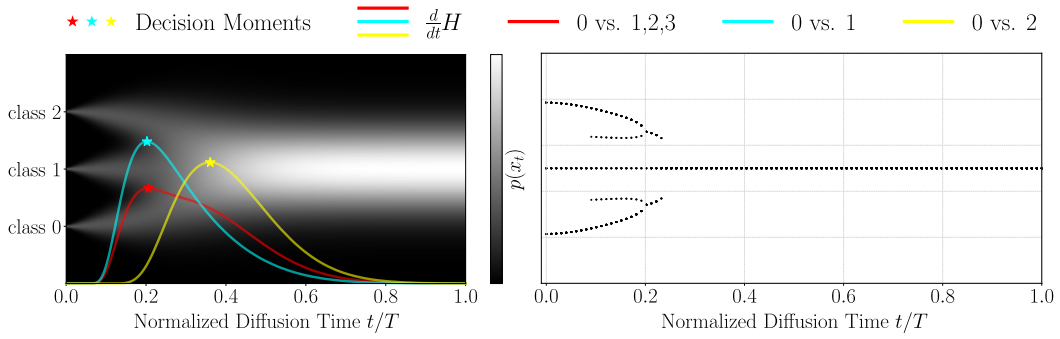

Figure 6: GMM with mixture weights $(\pi_0, \pi_1, \pi_2) = (1/3, 1/3, 1/3)$ and deltas at $(\mu_0, \mu_1, \mu_2) = (-2, 0, 2)$.

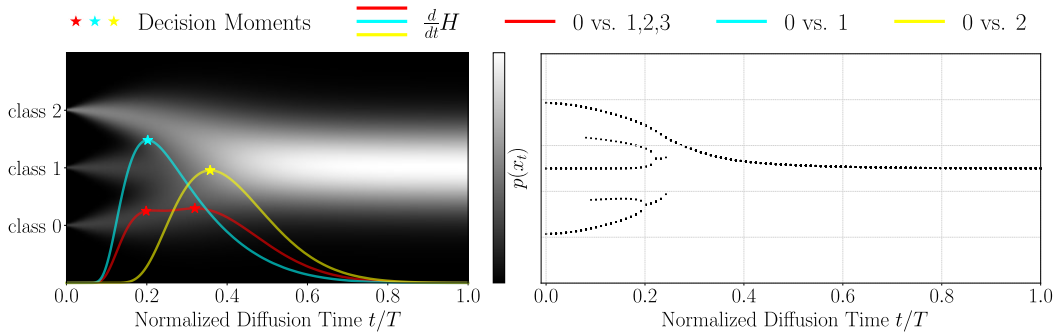

Figure 7: GMM with mixture weights $(\pi_0, \pi_1, \pi_2) = (0.25, 0.25, 0.5)$ and deltas at $(\mu_0, \mu_1, \mu_2) = (-2, 0, 2)$.

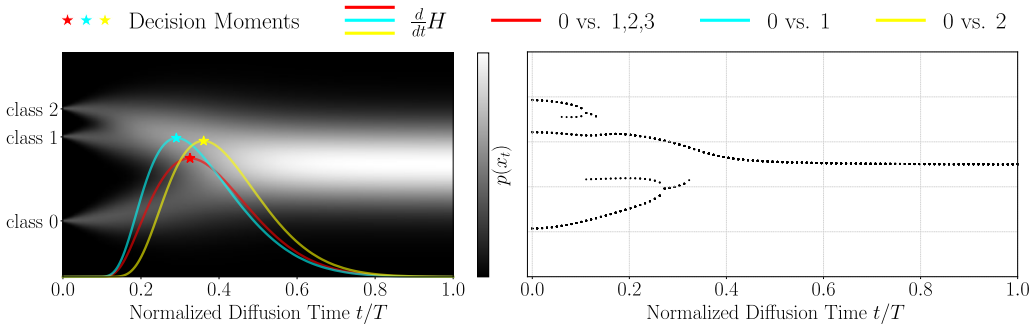

Figure 8: GMM with mixture weights $(\pi_0, \pi_1, \pi_2) = (1/3, 1/3, 1/3)$ and deltas at $(\mu_0, \mu_1, \mu_2) = (-2, 1, 2)$.

Figure 9: GMM with mixture weights $(\pi_0, \pi_1, \pi_2, \pi_3) = (0.25, 0.25, 0.25, 0.25)$ and deltas at $(\mu_0, \mu_1, \mu_2, \mu_3) = (-8, -4, 4, 8)$.

C Diffusion on CIFAR10

C.1 Estimating the Conditional Entropy

To estimate Eq. 2 we need to approximate the sampler, $p_z(x_t)$ and the posterior, $P(z_0|x_t)$. We split the integral in Eq. 2 into two parts:

$$-\sum_z P(z) \int p(x_t|z) \Big[P(z_0|x_t) \log P(z_0|x_t) + P(z_1|x_t) \log P(z_1|x_t) \Big] dx_t$$

We obtain $N_{z_0} = N_{z_1} = 1000$ samples for the individual expectations using ancestral sampling (T=1000 inference steps)

$$x_{t-1}^z = \frac{1}{\sqrt{1-\beta_t}} (x_t^z - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(x_t; z)) + \sqrt{\beta_t} \epsilon$$

where $\epsilon_\theta(x_t; z) = -\sqrt{1-\alpha_t} \nabla_{x_t} \log p(x_t|z)$. For c vs.$\neg c$, we use $\nabla_{x_t} \log p(x_t|\neg c) \approx \nabla_{x_t} \log p(x_t|\emptyset)$. We estimate the posteriors $P(z_0|x_t) = 1 - P(z_1|x_t)$ using the iterative procedure from Koulischer et al. (2025):

$$P(z_0|x_{t-1}) = P(z_0|x_t) \exp\big(\frac{1}{1-\beta_t} (||x_{t-1} - \mu_\theta(x_t; z_0)||^2 - ||x_{t-1} - \mu_\theta(x_t; z_1)||^2)\big)$$

The individual expectations are weighed by their respective priors, $P(z)$: $(0.5, 0.5)$ for c_1 vs. c_2 and $(0.1, 0.9)$ for c vs. $\neg c$. Here is the procedure outlined in more detail:

Algorithm 1 Estimate Conditional Entropy

Input: $N_{z_0}, N_{z_1}, T, P(z_0)$
$x_T^{z_0}, x_T^{z_1} \sim \mathcal{N}(0, I)$
$P(z_0|x_T) = P(z_0)$
$H_T = -\big(P(z_0)\log P(z_0) + (1 - P(z_0))\log(1 - P(z_0))\big)$
for t in $T : 1$ **do**
  for $z \in \{z_0, z_1\}$ **do**
    $x_t = x_t^z$
    $\mu_\theta(x_t; z_0) = \frac{1}{\sqrt{1-\beta_t}}\big(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, z_0)\big)$
    $\mu_\theta(x_t; z_1) = \frac{1}{\sqrt{1-\beta_t}}\big(x_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, z_1)\big)$
    if $t > 1$ **then**
      $\epsilon \sim \mathcal{N}(0, I) \cdot \sqrt{\beta_t}$
    else
      $\epsilon = 0$
    end if
    $x_{t-1} = \mu_\theta(x_t; z_0) + \epsilon$
    $P(z_0|x_{t-1}) = P(z_0|x_t)\exp\big(-\frac{1}{1-\beta_t}\big(||x_{t-1} - \mu_\theta(x_t; z_0)||^2 - ||x_{t-1} - \mu_\theta(x_t; z_1)||^2\big)\big)$
    $P(z_1|x_{t-1}) = 1 - P(z_0|x_{t-1})$
    $x_{t-1}^z = x_{t-1}$
    $H_{t-1}^z = \frac{1}{N_z}\sum_{x_{t-1}} P(z_0|x_{t-1})\log P(z_0|x_{t-1}) + P(z_1|x_{t-1})\log P(z_1|x_{t-1})$
  end for
  $H_{t-1} = -\big(P(z_0)H_t^{z_0} + (1 - P(z_0))H_t^{z_1}\big)$
end for
output: $H_{T:0}$

C.2 Additional Examples

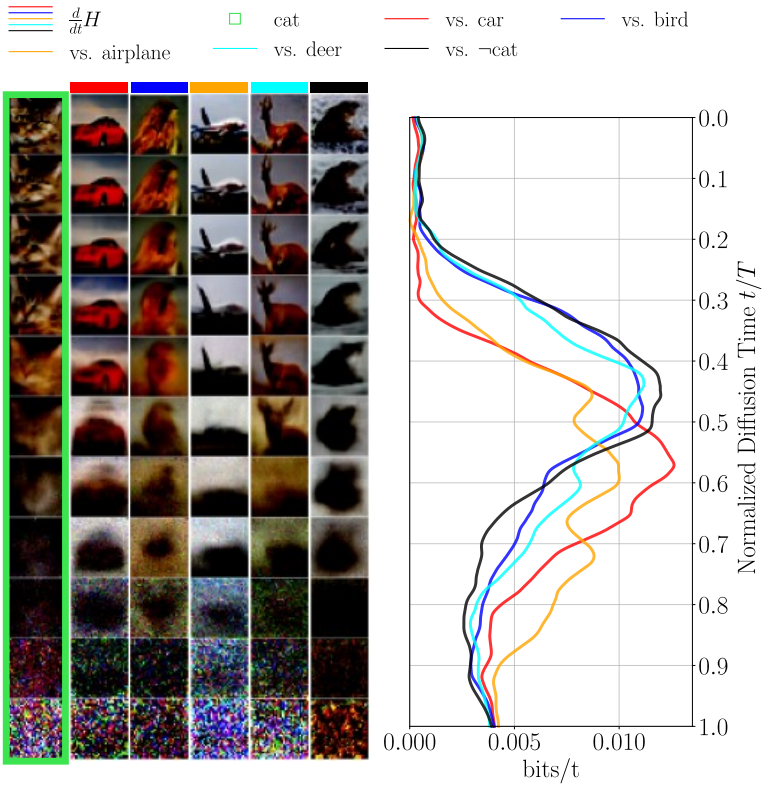

Figure 10: Experiments using the target class "cat".

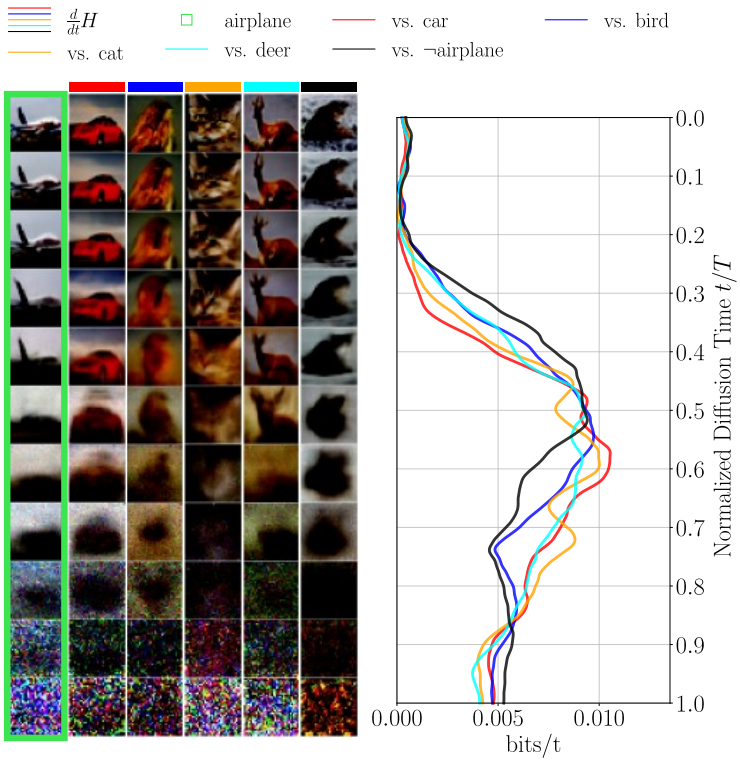

Figure 11: Experiments using the target class ”airplane”.

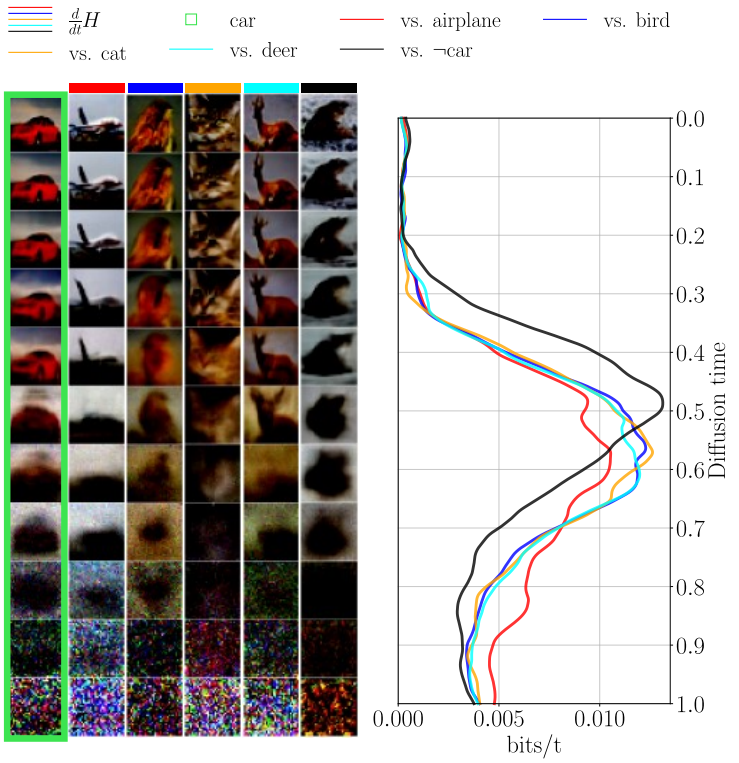

Figure 12: Experiments using the target class ”car”.

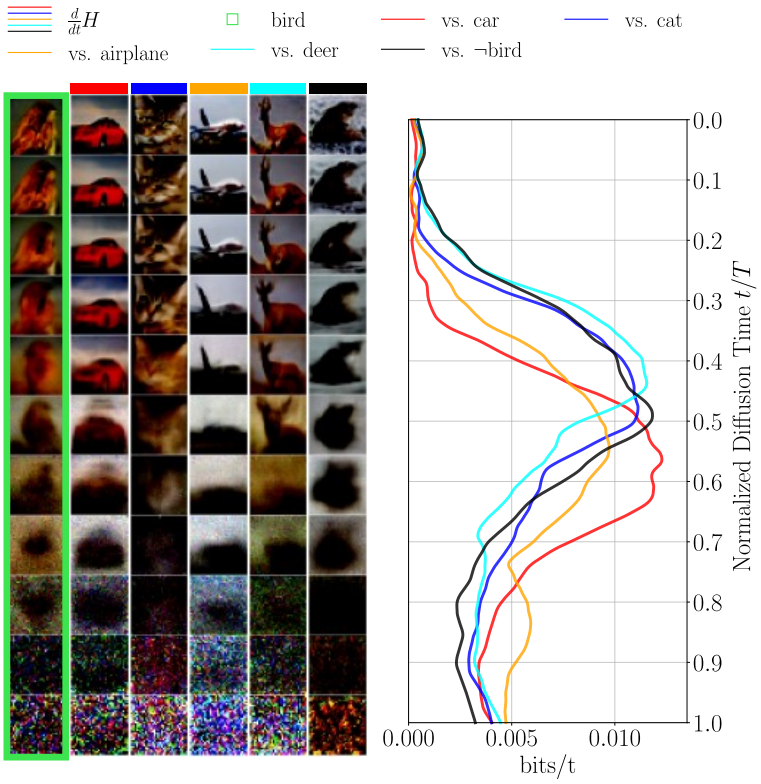

Figure 13: Experiments using the target class "bird".

C.3 Implementation Details

Datasets: All experiments were conducted using the CIFAR-10 Krizhevsky et al. (2009) dataset, which consists of 50,000 training and 10,000 test images, each image is a 32×32 RGB image. The dataset contains 10 classes: *airplane, car, bird, cat, deer, dog, frog, horse, ship* and *truck*, with 6,000 images per class. We trained two types of models: (1) an unconditional model utilizing the full dataset, and (2) class-specific models for the classes *airplane, car, bird, cat,* and *deer*, where each model used 6,000 training images for training.

Network architecture: All models were trained using the U-Net architecture from Denoising Diffusion Probabilistic Models (DDPM) Ho et al. (2020), where the noise prediction model is a modified PixelCNN++ based Unet Van den Oord et al. (2016); Salimans et al. (2017).

Training: Training was performed using the Adam optimizer with a batch size of 128, distributed across three Tesla V100 GPUs on an NVIDIA DGX-1 machine. The models were trained for 800 epochs using PyTorch 1.10.2+cu102, CUDA 10.2, and CuDNN 7605, with a learning rate of 0.0002, employing random horizontal flips as data augmentation.