Noise Conditional Maximum Likelihood Estimation with Score-based Sampling

Henry Li
Yale University
henry.li@yale.edu

Yuval Kluger
Yale University
yuval.kluger@yale.edu

Abstract

We introduce a simple modification to the standard maximum likelihood estimation (MLE) framework. Rather than maximizing a single unconditional likelihood of the data under the model, we maximize a family of *noise conditional* likelihoods consisting of the data perturbed by a continuum of noise levels. We find that models trained this way are more robust to noise, obtain higher test likelihoods, and generate higher quality images. They can also be sampled from via a novel score-based sampling scheme which combats the classical *covariate shift* problem that occurs during sample generation in autoregressive models. Applying this augmentation to autoregressive image models, we obtain 3.32 bits per dimension on the ImageNet 64x64 dataset, and substantially improve the quality of generated samples in terms of the Frechet Inception distance (FID) — from 37.50 to 12.09 on the CIFAR-10 dataset.

1 Introduction

Likelihood maximization models, *i.e.*, models trained by maximizing log-likelihood, are a leading class of modern generative models. Of these, autoregressive models boast state-of-the-art performance in many domains, including images Salimans et al. [2017], Child et al. [2019], text Vaswani et al. [2017], and audio Oord et al. [2016]. These architectures also show great promise for modeling long range dependencies Tay et al. [2020], Gu et al. [2021].

However, while log-likelihood is broadly agreed upon as one of the most rigorous metrics for goodness-of-fit in statistical and generative modeling, models with high likelihoods do not necessarily produce samples of high visual quality. This phenomenon has been discussed at length by Theis et al. [2015], Huszár [2015], and corroborated in empirical studies Grover et al. [2018], Kim et al. [2022].

Autoregressive models have an additional affliction: they have notoriously unstable dynamics during sample generation Bengio et al. [2015] due to their sequential sampling algorithm, which can cause errors to compound across time steps. Such errors cannot usually be corrected *ex post facto* due to the autoregressive structure of the model, and can substantially affect downstream steps as we find that their likelihoods are sensitive to even the most minor of perturbations.

Score-based diffusion models Song et al. [2020], Ho et al. [2020] offer a different perspective on the matter. Even though sampling is also sequential, diffusion models are more robust to perturbations because, in essence, they are trained as denoising functions Ho et al. [2020]. Moreover, the update direction in each step is unconstrained (unlike token-wise autoregressive models, which can only update one token at a time, and only once), meaning the model can correct errors from previous steps. However, diffusion models are poor likelihood models, as they cannot be trained via maximum likelihood, and density evaluations are inexact and require solving ODEs involving hundreds to thousands of function evaluations. Thus we wonder: is there a conceptual middle ground?

NeurIPS 2022 Workshop on Score-Based Methods.

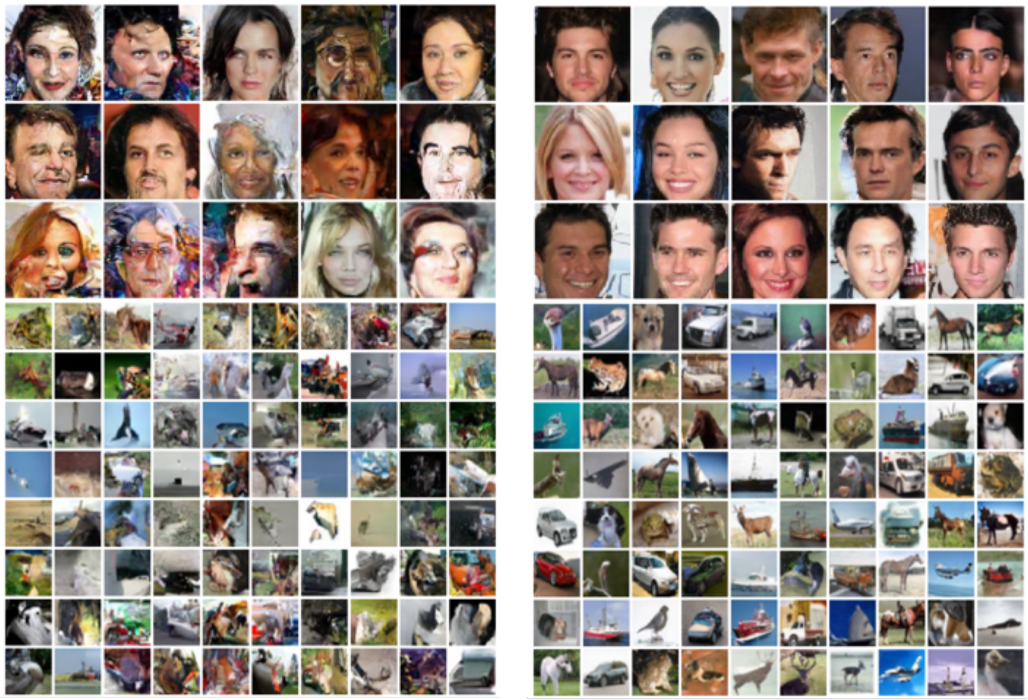

Figure 1: Generated samples on CelebA 64x64 (above) and CIFAR-10 (below). Autoregressive models trained via vanilla maximum likelihood (left) are brittle to sampling errors and can quickly diverge, producing nonsensical results. Those trained via our proposed algorithm (right) are more robust, which can significantly increase the coherence of the generated images.

In this paper, we offer such a framework. We further analyze the likelihood-sample quality mismatch in autoregressive models, and propose techniques inspired by diffusion models to alleviate it. In particular, we leverage the fact that the score function is naturally learned as a byproduct of maximum likelihood estimation. This allows a novel two-part sampling strategy with noisy sampling and score-based refinement.

Our contributions are threefold. 1) We investigate the pitfalls of training and inference under the log-likelihood maximization scheme, particularly regarding sensitivity to noise corruptions. 2) We propose a simple sanity test for checking the noise-robustness of likelihood models. 3) We introduce a novel framework for the training and sampling of likelihood maximization models that improves noise-robustness and substantially boosts the sample quality of the resulting model. Ultimately, we obtain a model that can generate samples at a quality approaching that of diffusion models, without losing the maximum likelihood framework and $\mathcal{O}(1)$ likelihood evaluation speed of likelihood maximization models.

2 The Pitfalls of Maximum Likelihood

We first show that density models trained to maximize the standard log-likelihood are surprisingly sensitive to minor perturbations. We then discuss why this is bad for generative modeling performance.

2.1 A Simple Sanity Test

Consider the class of minimally corrupted probability densities we call p_π, where

$$p_\pi = p_{data} * p_{mult_{\{-1,0,1\}}(\pi/2, 1-\pi, \pi/2)}, \qquad \pi \in [0,1]. \tag{1}$$

Here, $*$ denotes the convolution operator, and $p_{mult_{\{a,b,c\}}(\alpha,\beta,\gamma)}$ is the density a d-dimensional multinomial distribution taking on a, b, and c with probabilities α, β, and γ, respectively. p_π is *minimally corrupted* in the sense that, if p_{data} is an integer-discretized distribution (say, 8-bit images),

p_π describes the distribution of points p_{data} that have had their least significant bit incremented or decremented with probability π.

To the human eye, the difference between samples drawn from p_π and p_{data} is almost imperceptible, even for $\pi = 1$ (see Fig **??**). However, for likelihood models, this perturbation drastically increases the negative log-likelihood of the data under the model (see Table 1), to the point that it significantly undermines (if not outright nullifies) any recent advances in density estimation. This basic inconsistency suggests that the learned density of many standard likelihood models is brittle and overly emphasizes bit-level statistics that have little influence on the inherent content of the image.

2.2 Why We Should Care

We provide two reasons for why failing this test is problematic, especially for autoregressive models.

First, noise is natural — and being less robust to noise also means being a poorer fit to natural data. Outside of the log-likelihood, measures of generative success in generative models fall under two categories: qualitative assessments (*i.e.*, the no-reference perceptual quality assessment Wang et al. [2002] or 'eyeballing' it) and quantitative heuristics (*i.e.*, computing statistics of hidden activations of pretrained CNNs Salimans et al. [2016], Heusel et al. [2017], Sajjadi et al. [2018]). Both strategies either rely directly on the human visual system, or are known to be closely related to it Güçlü and van Gerven [2015], Yamins et al. [2014], Khaligh-Razavi and Kriegeskorte [2014], Eickenberg et al. [2017], Cichy et al. [2016]. Therefore, implicit in the use of these criteria is the existence of a human (or human-like) model of images q_{human}, where $q_{human} \approx p_{data}$ Huszár [2015]. The fact that we find samples from p_π nearly indistinguishable from p_{data}, whereas p_θ finds them very different suggests that $p_{data} \approx q_{human} \neq p_\theta$.

Second, sample quality suffers. This holds for general likelihood models, given what we argue in the first point — namely $p_\theta \neq q_{human}$. However, noise-sensitivity is doubly problematic in autoregressive models. Due to the sequential nature of autoregressive sampling and the fact that models are trained entirely on data from the *true* distribution, any sampling error can drastically affect the sampling trajectory. This is related to the well-known *covariate shift* phenomenon Bengio et al. [2015], Shimodaira [2000] Moreover, such errors compound quickly. Table 1 shows that mis-sampling pixels by even a single bit can cause drastic changes to the overall likelihood. This can explain why standard autoregressive models commonly produce nonsensical results (Fig 1).

3 Noise Conditional Maximum Likelihood

To alleviate the problems discussed in Section 2, we propose a simple modification to the standard objective in maximum likelihood estimation. Rather than evaluating a single likelihood as in the vanilla formulation, we consider a family of noise conditional likelihoods

$$\mathbb{E}_{\sigma\sim\mu}\mathbb{E}_{\mathbf{x}\sim p_\sigma} \log p_{\boldsymbol{\theta},\sigma}(\mathbf{x}), \tag{2}$$

where p_σ is a stochastic process indexed by noise scales σ describing a noise-corrupted version of p_{data}, and μ is a distribution over such scales. We call this the noise conditional maximum likelihood (NCML) framework. In general, (2) is an all-purpose plug-in objective that can be used with any likelihood model adapted to accept a noise conditioning vector, though a continuous likelihood (e.g. Salimans et al. [2017], Li and Kluger [2022]) is necessary for computation of the score function.

Letting σ be the time index of a diffusion process, our approach becomes closely related to score-based diffusion models Song et al. [2020], albeit with two crucial differences.

First, instead of merely estimating the noise conditional score function of the perturbed data density p_σ for $\sigma \in [0, T]$, we directly estimate p_σ itself. However, we still learn the noise conditional score function as a by-product of NCML. Moreover, we may access the score function by simply differentiating the log likelihood. Therefore, we can refine sampled points via Langevin dynamics. This provides an alternative strategy for sampling from $p_{\theta,\sigma}$, which we explore in 3.1.

Second, we need not design our diffusion so that p_T approximates the limiting stationary distribution of the process. This is necessary in diffusion models as the limiting prior is the only tractable distribution to initialize the sampling algorithm with. Since we have learned the density itself for all $\sigma \in [0, T]$, we may initialize from any point of the diffusion, which increases the flexibility of the sampling strategy, and can drastically reduce the steps required to solve the reverse diffusion.

Model	CIFAR-10				ImageNet 64x64		
	FID	NLL $\pi = 0$*	NLL $\pi = 0.5$	NLL $\pi = 1$	NLL $\pi = 0$*	NLL $\pi = 0.5$	NLL $\pi = 1$
ELBO							
VDM	7.41	**2.49**	**3.75**	**3.97**	**3.40**	3.76	3.88
ScoreFlow	**5.40**	2.90	3.82	3.99	-	-	-
VDVAE	-	2.84	3.90	4.10	3.52	**3.66**	**3.82**
Likelihood							
Flow++	-	3.09	3.86	4.08	3.69	3.82	3.99
DenseFlow	48.15	2.98	3.80	4.02	3.35	3.68	3.85
PixelCNN++	55.72	2.92	3.84	4.01	3.52	3.84	4.00
PixelSNAIL	36.62	2.85	3.83	3.99	-	-	-
Sparse Transformer	37.50	**2.80**	3.82	3.98	3.44	3.73	3.89
NCPN (ML)	46.72	2.91	3.83	3.99	3.49	3.68	3.88
NCPN (NCML-VE)	32.71	2.87	3.75	3.95	**3.32**	3.67	3.85
NCPN (NCML-subVP)	23.42	2.95	3.69	3.94	3.36	3.66	3.82
NCPN (NCML-VP)	**12.09**	3.20	**3.62**	**3.91**	3.43	**3.63**	**3.79**

Table 1: Results on CIFAR-10 and ImageNet 64x64. Negative log-likelihood (NLL) is in bits per dimension. Lower is better. *NLL with $\pi = 0$ is equivalent to NLL of the original data.

3.1 Sampling with Autoregressive NCML Models

The NCML framework allows for two sampling strategies. The first is to draw directly from the noise-free distribution $p_{\boldsymbol{\theta},0}$, in which case the conditional likelihood simplifies to a standard (unconditional) likelihood, and sampling is identical to that for a standard autoregressive model.

However, as discussed in Section 2, this strategy is unstable and tends to quickly accumulate errors. This motivates an alternative sampling strategy, which involves drawing from $p_{\theta,\sigma}$ for $\sigma > 0$, then solving a reverse diffusion process back to $\sigma = 0$. The latter is possible due to the fact that the reverse diffusion is itself a diffusion process that depends on the score function Anderson [1982], which we have access to. This is identical to the sampling procedure in score-based diffusion models Song et al. [2020], except for the key difference that we need not initialize with the prior distribution.

4 Experiments and Discussion

In all experiments in Table 1, we choose p_σ to be the variance exploding (VE), variance preserving (VP), and sub-variance preserving (sub-VP) SDEs, respectively. Due to space constraints, we refer to Song et al. [2020] for more details. For our architecture, we introduce the noise conditional pixel-wise network (NCPN), which consists of a PixelCNN backbone with added attention layers. We evaluate all models on minimally perturbed transformations (see 2.1) of CIFAR-10 and ImageNet 64x64 for $\pi \in \{0, \frac{1}{2}, 1\}$, where we note that $p_{\pi=0} = p_{data}$. All noise conditional models, *i.e.*, ours, VDM Kingma et al. [2021], and ScoreFlow Song et al. [2021], are evaluated at $t = 0$.

While it is clear that all models have reduced likelihoods when evaluated on the perturbed distribution $p_\pi, \pi \in \{\frac{1}{2}, 1\}$, we note that our models are more robust to such transformations, even though they are evaluated under the noiseless condition, and trained on a different class of noise, *i.e.*, the marginal likelihoods of diffusion processes. Furthermore, sample quality across all models correlates better with likelihoods on the perturbed distributions than likelihoods on the base distribution.

5 Conclusion

We proposed a simple sanity test for checking the robustness of likelihoods to minor perturbations. We found that most likelihood models are not robust under this test, and developed a new framework that improves performance in this setting, with substantial improvements in training and sampling.

References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 864–872. PMLR, 2018.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6(1):1–13, 2016.

Michael Eickenberg, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, 152: 184–194, 2017.

Matej Grcić, Ivan Grubišić, and Siniša Šegvić. Densely connected normalizing flows. *Advances in Neural Information Processing Systems*, 34:23968–23982, 2021.

Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Umut Güçlü and Marcel AJ van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019a.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019b.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015.

Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915, 2014.

Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. In *International Conference on Machine Learning*, pages 11201–11228. PMLR, 2022.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Henry Li and Yuval Kluger. Neural inverse transform sampler. In *International Conference on Machine Learning*, pages 12813–12825. PMLR, 2022.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in neural information processing systems*, 31, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf`.

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.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. *arXiv preprint arXiv:2011.04006*, 2020.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Zhou Wang, Hamid R Sheikh, and Alan C Bovik. No-reference perceptual quality assessment of jpeg compressed images. In *Proceedings. International conference on image processing*, volume 1, pages I–I. IEEE, 2002.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Model	NLL $\pi = 0$*	NLL $\pi = 0.5$	NLL $\pi = 1$
NCPN (ML)	2.25	3.72	4.35
NCPN (NCML VE)	2.22	3.63	4.21
NPCN (NCML sub-VP)	2.31	3.44	3.98
NCPN (NCML VP)	2.48	3.14	3.67

Table 2: Results on CelebA 64x64. Negative log-likelihood (NLL) is in bits per dimension. Lower is better. *NLL with $\pi = 0$ is equivalent to NLL of the original data.

A Appendix

A.1 Additional Experimental Details

For experiments on CIFAR-10 and ImageNet 64x64, we compare against Kingma et al. [2021], Song et al. [2021], Child [2020], Ho et al. [2019a], Grcić et al. [2021], Salimans et al. [2017], Chen et al. [2018], Child et al. [2019]. Some results could not be included due to the irreproducibility of the techniques. There is limited existing work on likelihood-based modeling on CelebA 64x64, so we do not provide comparisons here.

Our proposed NCPN architecture consists of the PixelCNN++ backbone Salimans et al. [2017] with axial attention layers Ho et al. [2019b] after each residual block. We retain the hyperparameters of PixelCNN++, changing only the dropout on the CIFAR-10 dataset (from 0.5 to 0.25), which we reduced due to the regularization properties of NCML. For the axial attention layers, we use 8 heads and skip connection rescaling as in Song et al. [2020]. Finally, we add noise conditioning to each residual block via a Gaussian Fourier Projection layer, much like Ho et al. [2020], Song et al. [2020].

For our NCML-trained models, the diffusion times of the VE, VP, and sub-VP SDEs were chosen to be $T = 0.5$, $T = 0.1$, and $T = 0.025$, respectively. The values are somewhat arbitrary, but selected such that the standard deviation of the per-pixel differences between samples in p_{data} and their noised counterparts in p_T was ≈ 10. We suspect that further improvements can be made to the empirical results if these numbers were chosen more judiciously.

All NCPN models were trained on RTX 2080 Ti GPUs for 500,000 iterations. This is approximately 1.5 weeks of training. We use the same NCPN architecture and hyperparameters across all datasets (except for dropout, which is set to 0.25 on CIFAR-10 and 0.00 on ImageNet 64x64 and CelebA 64x64). All NCPN models have 73M parameters.

A.2 Additional Figures and Tables

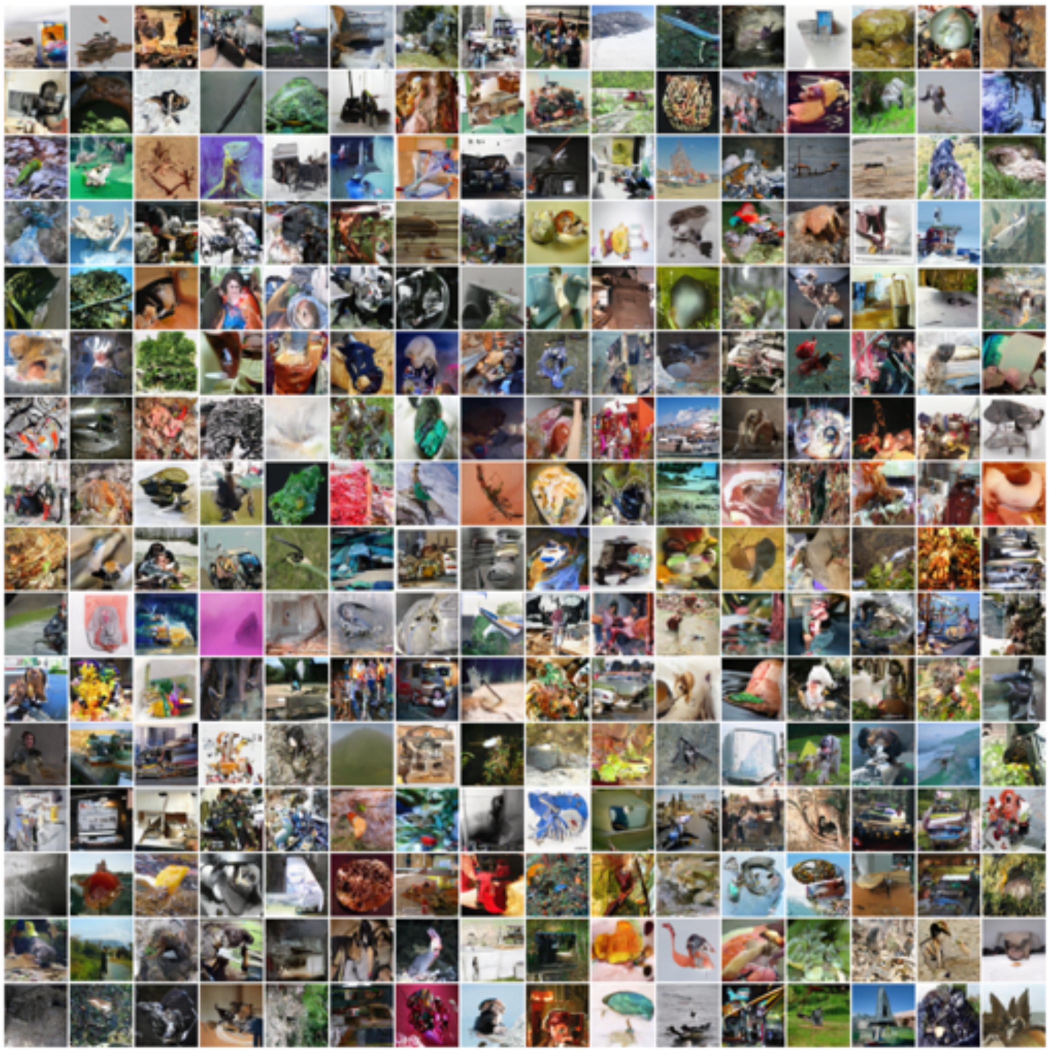

Figure 2: Samples from NCPN trained on ImageNet 64x64, with p_t as a variance preserving (VP) diffusion process.

Figure 3: Samples from NCPN trained on CelebA 64x64, with p_t as a variance preserving (VP) diffusion process.

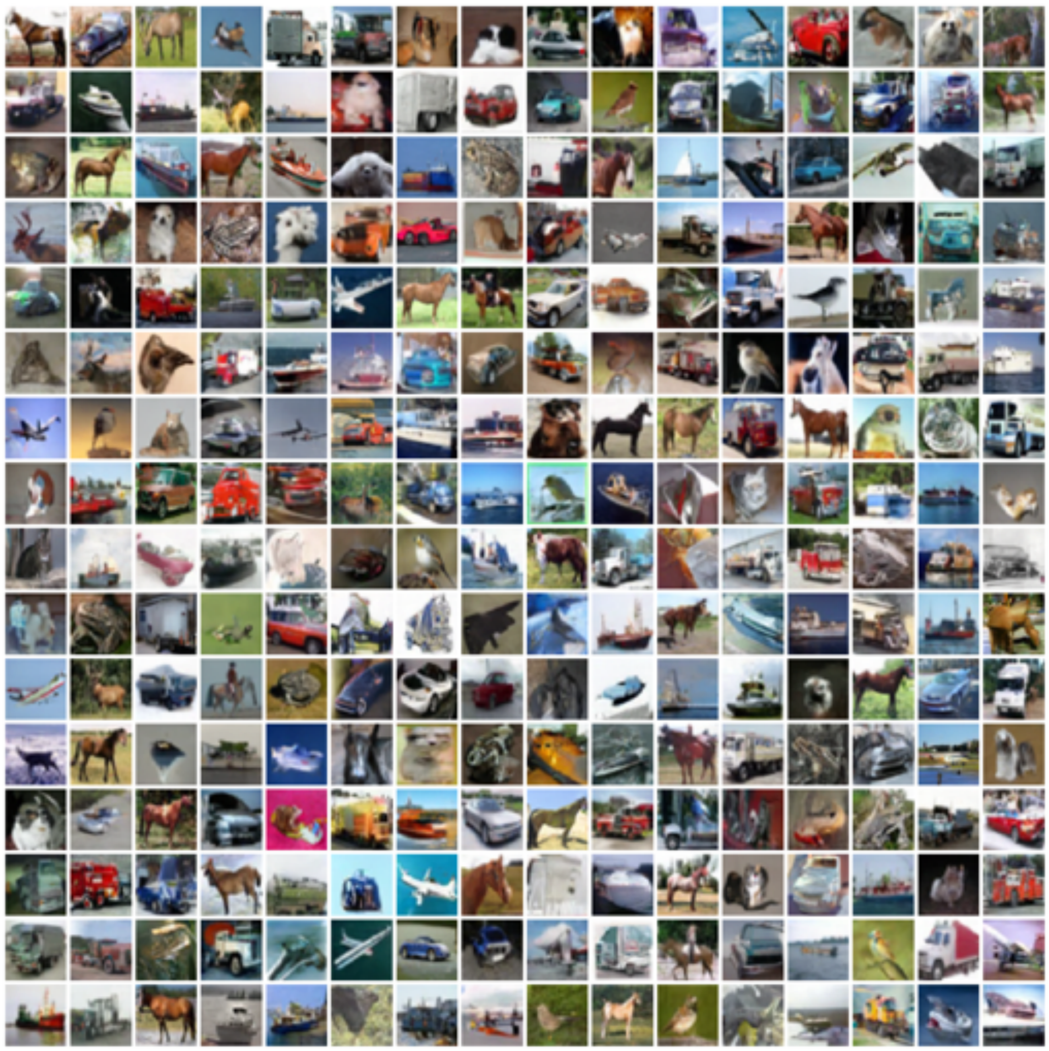

Figure 4: Samples from NCPN trained on CIFAR-10, with p_t as a variance preserving (VP) diffusion process.

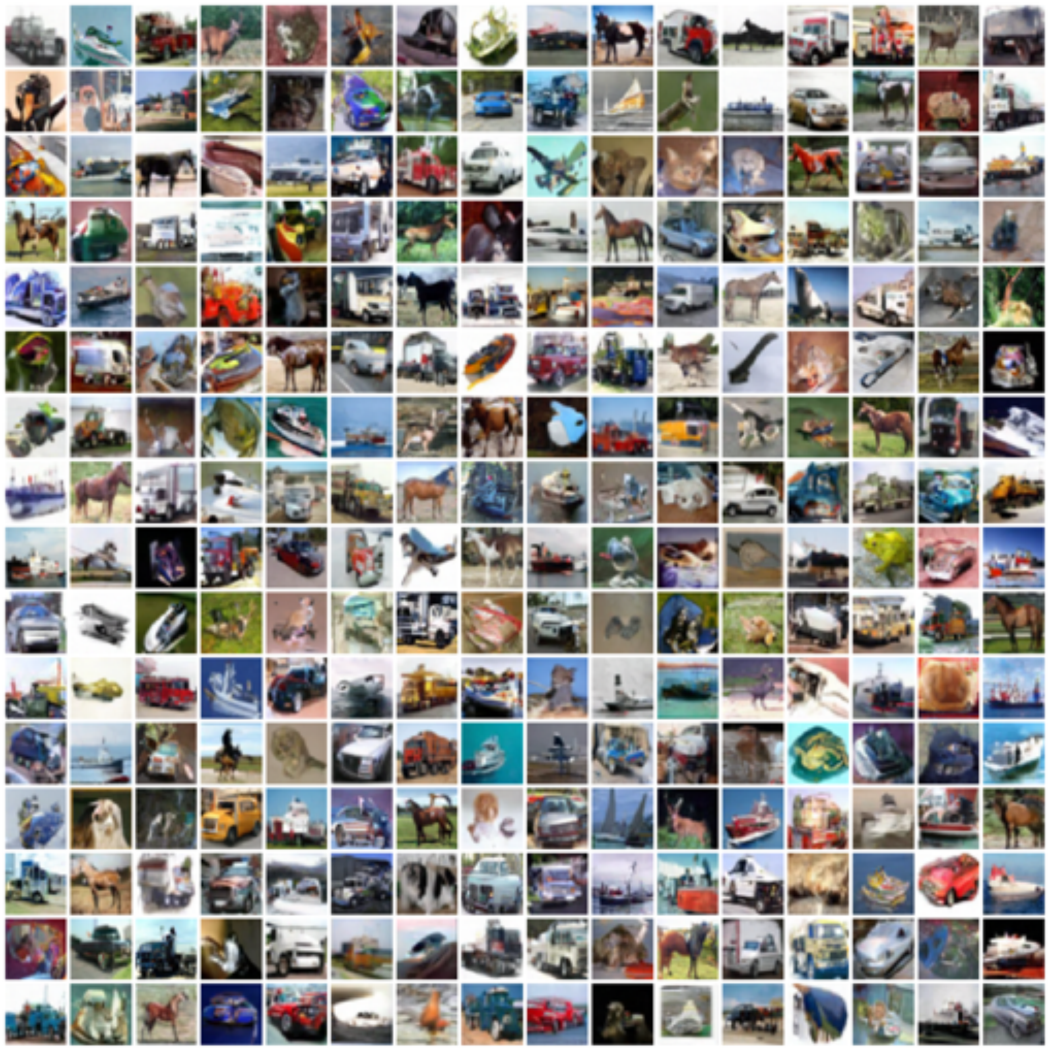

Figure 5: Samples from NCPN trained on CIFAR-10, with p_t as a sub-variance preserving (sub-VP) diffusion process.

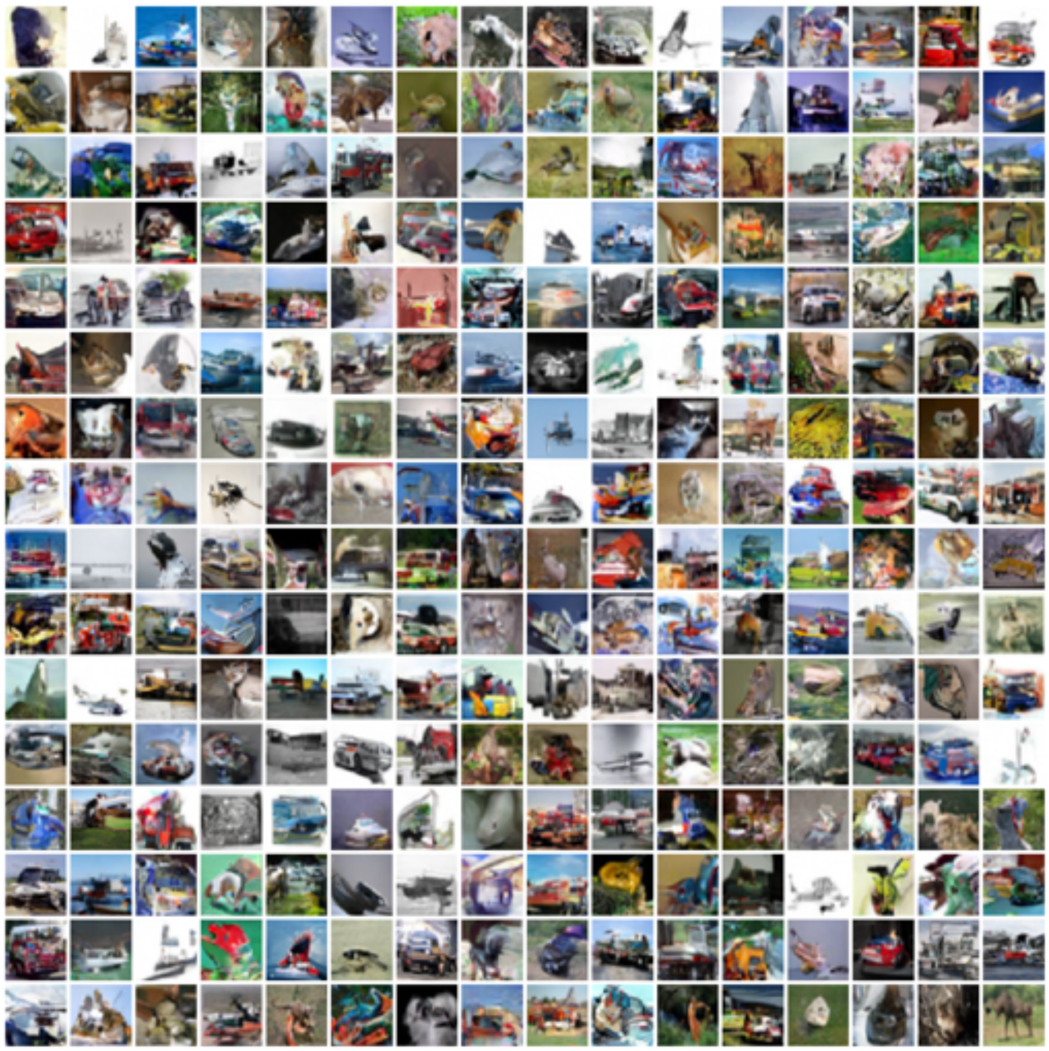

Figure 6: Samples from NCPN trained on CIFAR-10, with p_t as a variance exploding (VE) diffusion process.