Conditional denoising diffusion probabilistic models for the clustering of images

Seydina Ousmane Niang, Charles Bouveyron, Marco Corneli, Pierre Latouche

Conditional denoising diffusion probabilistic models for the clustering of images

Seydina NIANG [1] & Charles BOUVEYRON [1] & Marco CORNELI[1,2] & Pierre LATOUCHE [3]

[1] *Université Côte d'Azur, Inria, CNRS, Laboratoire J.A.Dieudonné, Maasai team, Nice, France*
[2] *Université Côte d'Azur, Laboratoire CEPAM, Nice, France*
[3] *Université Clermont Auvergne, CNRS, Laboratoire LMBP, Aubière, France*

Résumé. Dans le domaine du traitement d'images, les modèles de diffusion du débruitage (DDPM) ont gagné en popularité pour leur capacité à modéliser les distributions de données complexes tout en permettant une génération réaliste d'images. Ces modèles reposent sur deux étapes principales : une phase de diffusion directe où les images sont graduellement corrompues par du bruit gaussien, suivie d'une étape de décodage inverse où les images bruitées sont débruitées étape par étape à l'aide d'un réseau de neurones. Dans ce travail, nous proposons une extension du modèle DDPM en introduisant une version conditionnelle qui prend en compte l'appartenance des images à différents groupes. Cette approche permet de mieux capturer la structure sous-jacente des données en tenant compte de leur regroupement naturel. Concrètement, le modèle repose sur l'hypothèse que les images sont réparties en Q groupes, et chaque image est modélisée comme provenant d'une distribution conditionnelle sur les groupes. Un réseau de neurones est ensuite entraîné pour prédire le bruit à supprimer lors de la génération des images, en fonction de leur appartenance à un groupe.

L'inférence dans ce modèle complexe est effectuée à l'aide d'un algorithme de type EM variationnel, qui permet d'estimer de manière efficace les paramètres de regroupement et les variables latentes. Cependant, l'optimisation des paramètres du réseau de neurones pose des défis supplémentaires, nécessitant l'utilisation de techniques avancées comme la descente de gradient stochastique. En combinant les avantages des modèles de diffusion du débruitage avec une approche de clustering conditionnelle, cette méthode ouvre de nouvelles perspectives dans le domaine de l'apprentissage automatique et du traitement d'images, offrant des outils puissants pour l'analyse et la génération de données visuelles complexes.
Mots-clés. Modèles de diffusion du débruitage, traitement d'images, algorithme EM, descente de gradient stochastique, inférence variationnelle.

Abstract. This document presents an innovative method for clustering image data while learning to generate new images. In the field of image processing, denoising diffusion probabilistic models (DDPMs) have gained popularity for their ability to model complex data distributions while enabling realistic image generation. These models rely on two main steps: a forward diffusion phase where images are gradually corrupted by Gaussian noise, followed by a reverse decoding step where the noisy images are denoised step by step using a neural network. In this work, we propose an extension of the DDPM model by introducing a conditional version that takes into account the membership of images to different clusters. This approach allows for better capturing the underlying structure of the data by considering their natural grouping. Specifically, the model is based on the

assumption that images are distributed into Q clusters and that each image is modeled as coming from a conditional distribution over the clusters. A neural network is then trained to predict the noise to be removed during data generation, based on the image membership to a specific cluster. Inference in this complex model is performed using a variational EM-type algorithm, which efficiently estimates the clustering parameters and latent variables. However, optimizing the parameters of the neural network poses additional challenges, requiring the use of techniques such as stochastic gradient descent.
By combining the advantages of denoising diffusion models with a conditional clustering approach, this method opens up new perspectives in the field of machine learning and image processing, providing powerful tools for the analysis and generation of complex visual data.
Keywords. Diffusion probabilistic models, variational inference, expectation-maximisation algorithm (EM), image processing, stochastic gradient descent.

1 Introduction

In recent years, generative models have made significant strides in generating human-like natural language, high-quality synthetic images and diverse human speech and music. These models find applications in various domains, such as generating images from text prompts or learning useful feature representations. Despite their ability to produce realistic outputs, there remains ample room for improvement in generative models, which could have broad implications across graphic design, gaming, music production and beyond.
Generative adversarial networks (GAN) (Creswell et al., 2018) recently led the field of image generation tasks, as measured by metrics like FID (Fréchet inception distance), inception score and precision used to evaluate the quality and diversity of generated images. However, GANs often struggle with diversity and can be challenging to train effectively, requiring careful tuning of hyper-parameters and regularizers. While likelihood-based models offer advantages in terms of diversity and ease of training, they still fall short in visual fidelity compared to GANs. Diffusion models, a class of likelihood-based models, have shown promising results in producing high-quality images with desirable properties such as distribution coverage and scalability. As shown in Dhariwal and Nichol, 2021, they can outperform GANs in the context of image processing. Here we propose to condition the diffusion process to the assignment to a cluster of the images and we believe that this can improve the efficiency of the generation while learning a way to cluster the images.

2 Conditional diffusion probabilistic model for the clustering of images

2.1 Denoising diffusion probabilistic models

A denoising diffusion probabilistic model (DDPM) (Ho, Jain, and Abbeel, 2020) makes use of two Markov chains: a forward chain that perturbs data to noise and a reverse chain that converts noise back to data. The former is typically hand-designed with the goal

to transform any data distribution into a simple prior distribution (e.g., standard Gaussian), while the latter Markov chain reverses the former by learning transition kernels parameterized by deep neural networks. New data points are subsequently generated by first sampling a random vector from the prior distribution, followed by ancestral sampling through the reverse Markov chain. Formally, given a target data distribution $x_0 \sim q^\star(x_0)$, the forward Markov process generates a sequence of random variables $x_1, x_2 \ldots, x_T$ with transition kernel $q(x_t|x_{t-1})$. Using the chain rule of probability and the Markov property, we can factorize the joint distribution of $x_1, x_2, \ldots x_T$ conditioned on x_0, denoted as $q(x_1, \ldots, x_T|x_0)$, into

$$q(x_1, ..., x_T|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{1}$$

In DDPMs, we handcraft the transition kernel $q(x_t|x_{t-1})$ to incrementally transform the data distribution $q^\star(x_0)$ into a tractable prior distribution. One typical design for the transition kernel is Gaussian perturbation and the most common choice for the transition kernel is

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \cdot \mathbf{I}_D), \tag{2}$$

where D is the dimensionality of the data x_0 and $\beta_t \in (0, 1)$ is a hyper-parameter chosen ahead of model training. This kernel is the most used, although other types of kernels are also applicable in the same vein. Specifically, with $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t) \cdot \mathbf{I}_D). \tag{3}$$

Given x_0, we can easily obtain a sample of x_t by sampling a Gaussian vector $\epsilon \sim \mathcal{N}(0, \mathbf{I}_D)$ and applying the reprametrisation trick:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon. \tag{4}$$

When $\bar{\alpha}_T \sim 0$ (true for high values for T), x_T follows a Gaussian white noise $q(x_T) \simeq \mathcal{N}(X_T; 0, \mathbf{I}_D)$. Intuitively, this forward process slowly injects noise to data until all structures are lost.

For generating new data samples, DDPMs start by first generating an unstructured noise vector from the prior distribution (which is typically trivial to obtain), then gradually remove noise by running a learnable Markov chain in the reverse time direction. Specifically, the reverse generative Markov chain is parameterized by a prior distribution $p(x_T) = \mathcal{N}(X_T; 0, \mathbf{I}_D)$ and a learnable transition kernel $p_\theta(x_{t-1}|x_t)$. This learnable transition kernel takes the form of

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{5}$$

where θ denotes model parameters and the mean $\mu_\theta(x_t, t)$ and variance $\Sigma_\theta(x_t, t)$ are parameterized by a deep neural networks. With this reverse Markov chain in hand, we can generate a data sample x_0 by first sampling a noise vector $x_T \sim p(x_T)$, then iteratively sampling from the learnable transition kernel $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ until $t = 1$.

Key to the success of this sampling process is training the reverse Markov chain to match the actual time reversal of the forward Markov chain. That is, we have to adjust the parameter θ so that the joint distribution of the reverse Markov chain $p_\theta(x_0, \ldots x_T)$ closely approximates that of the forward process $q(x_0, \ldots x_T)$. This is achieved by minimizing the Kullback-Leibler (KL) divergence between these two:

$$\mathcal{L} = \mathrm{KL}(p_\theta(x_0, \ldots x_T)||q(x_0, \ldots x_T)). \tag{6}$$

To sum up, diffusion models are powerful generative models that work by gradually adding noise to a latent representation of the data and then inverting the process to reconstruct the data. This process can be used to create realistic images, even when the latent representation is simple. A major question is whether diffusion models are flexible enough to be adapted to the clustering field or not.

2.2 Our contribution: mixture of denoising diffusion probabilistic models

Mixture models are a powerful statistical technique used in machine learning and data analysis, particularly for the purpose of clustering. Clustering is the process of grouping similar data points together, where the similarity between data points is defined based on certain characteristics or features. Through this section, we build a mixture of denoising diffusion probabilistic models for the aim of images clustering.
Let us suppose that the train dataset is gathered into a matrix $\{X_i^0\}_{i\leq N}$ corresponding to a collection of images regrouped into Q types (clusters). Each line X_i^0 of the matrix X_0 is a flattened image in R^D. N denotes the number of images considered and D is the number of pixels.

Generative model:
We posit the following distribution for the data:

$$p(X^0|\mu,\theta,\pi) = \prod_{i=1}^{N}\sum_{q=1}^{Q}\pi_q p_\theta(X_i^0|\mu_q), \tag{7}$$

where $\pi = (\pi_q)_q$ denotes the mixture proportions (the probability that an image is drawn following the q-th component of the mixture), μ_q corresponds to the mean of the prior Gaussian distribution of the q-th diffusion model and p_θ refers to a conditional diffusion model (Lu et al., 2022). Indeed we suppose that the diffusion process (the way noise is added to images) is specific to each cluster. Clustering arises from this probabilistic formulation with the introduction of an unobserved random variable $Z_i \in \{0,1\}^Q$ such that $Z_{iq} = 1$ if and only if the image i belongs to the q-th component. Z_i is supposed to be drawn following a multinomial distribution of parameter π

$$Z_i \overset{\text{iid}}{\sim} \text{Multinomial}(1, \pi = (\pi_{1:Q})), \quad \forall i \in \{1,\dots,N\}. \tag{8}$$

The corresponding partition $Z = \{Z_i\}$ is then considered as the set of discrete latent variables which are to be estimated, along with the model parameters. Traditionally the posterior distribution of Z is estimated by an expectation-maximization algorithm (EM) through variational inference.
Hence the conditional distribution of the i-th image given Z_i is:

$$X_{i\,|Z_{iq}=1}^0 \sim p_\theta(\cdot|\mu_q).$$

Instead of generating image directly from latent noise as done in variational auto-encoders (Kipf and Welling, 2016), we construct the image sequentially. We begin by introducing for each image i, $T \in \mathbf{N}^\star$ latent variables $X_i^{1:T}$ such that $X_i^{0:T}$ conditioned to Z is a reverse

time Markov chain with transition kernel $p_\theta(X_i^{t-1}|X_i^t, \mu_q)$. Hence:

$$p_\theta(X_i^{0:T}|Z, \mu) = \prod_{q=1}^{Q} \left(p(X_i^T) \prod_{t=1}^{T} p_\theta(X_i^{t-1}|X_i^t, \mu_q) \right)^{Z_{iq}}, \tag{9}$$

where

$$\begin{aligned} p(X_i^T) &= \mathcal{N}(X_i^T; 0, I_D) \quad \text{and} \\ p_\theta(X_i^{t-1}|X_i^t, Z_{iq} = 1, \mu_q) &= \mathcal{N}(X_i^{t-1}; \mu_\theta(X_i^t, t, \mu_q), \bar{\delta}_t \cdot I_D), \end{aligned} \tag{10}$$

with μ_θ a neural network and $\bar{\delta} > 0$ an hyper-parameter.
We point out that the reverse transition kernel now also depends on the cluster means μ_q. This assumption seems natural, since we start with a white noise and want to generate images of a certain type.

3 Inference

The complete-data log-likelihood is computed by integrating on all the latent variables, giving:

$$\log p_\theta(X^0) = \log \left[\sum_Z \int_{X^1,\dots,X^T,} p_\theta(X^0, \dots, X^T, Z) dX^1 \dots dX^T \right], \tag{11}$$

where , $X^t := (X_i^t)_i$ for all t and all i.
Unfortunately, the above log-likelihood is untractable. To tackle this problem, we rely on variational inference. We introduce a variational posterior distribution $q(\cdot)$ on the latent variables that factorizes as:

$$\begin{aligned} q(X^1, \dots, X^T, Z|X^0) &= q(X^0, \dots, X^T|Z, X^0) \cdot q(Z) \\ &= \prod_{i=1}^{N} q_i(Z_i) \prod_{q=1}^{Q} \left(\prod_{t=1}^{T} q(X_i^t|X_i^{t-1}, \mu_q) \right)^{Z_{iq}}, \end{aligned} \tag{12}$$

where

$$q_i(Z_i) := \mathcal{M}(Z_i; 1, \tau_i), \tag{13}$$

and

$$q(X_i^t|X_i^{t-1}, \mu_q) = \mathcal{N}\left(X_i^t; \frac{1-m_t}{1-m_{t-1}} \sqrt{\alpha_t} X_i^{t-1} + \left(m_t - \frac{1-m_t}{1-m_{t-1}} m_{t-1} \right) \sqrt{\bar{\alpha}_t} \mu_q, \quad \delta_{t|t-1} I_D \right), \tag{14}$$

with $\delta_{t|t-1} = \delta_t - \left(\frac{1-m_t}{1-m_{t-1}} \right)^2$, $\delta_t = (1-\bar{\alpha}_t)^2 - m_t^2 \bar{\alpha}_t$ and $m_0 \simeq 0 \leq \dots \leq m_T \simeq 1$.
As it can be seen, the model is flexible enough to introduce noise in distinct ways based on cluster membership. What sets our approach apart is the unconventional aspect of both the variational and generative distributions being governed by the same parameter μ. However we need both the forward and reverse process (i.e. from image to noise) to be cluster-dependant.

$q(X_i^t|X_i^0, \mu_q)$ is computed by marginalisation w.r.t the intermediate latent variables, resulting into

$$q(X_i^t|X_i^0, Z_{iq} = 1, \mu_q) = \mathcal{N}(X_i^t; (1-m_t)\sqrt{\bar{\alpha}_t} X_i^0 + m_t \sqrt{\bar{\alpha}_t} \mu_q, \delta_t I_D). \tag{15}$$

Here the mean of the Gaussian distribution is an interpolation of the original image and the cluster mean. The interest of this formula can be outlined by the reparametrization trick:

$$X_i^T = (1 - m_T)\sqrt{\bar{\alpha}_T}X_i^0 + m_T\sqrt{\bar{\alpha}_T}\mu_q + \sqrt{\delta_t}\eta_T,$$

where $\eta_T \sim \mathcal{N}(0, I_D)$.
Given that $m_T \simeq 1$ and $\alpha_T \to 0$, it follows that X_i^T behaves nearly like white noise, denoted as $X_i^T \simeq \eta_T$. This result is coherent with $p(X_i^T) = \mathcal{N}(X_i^T; 0, I_D)$.

If we inject the variational distribution into the untractatable log-likelihood, we obtain:

$$\begin{aligned}\log p_\theta(X^0) &= \log\left[\sum_Z \int_{X^1,\dots,X^T} \frac{p_\theta(X^0,\dots,X^T,Z)}{q(X^1,\dots,X^T,Z|X^0)} \cdot q(X^1,\dots,X^T,Z|X^0)dX^1\dots dX^T\right] \\ &= \log E_{X^1,\dots,X^T,Z\sim q(\cdot|X^0)}\left[\frac{p_\theta(X^0,\dots,X^T,Z)}{q(X^1,\dots,X^T,Z|X^0)}\right].\end{aligned} \tag{16}$$

Since log is concave, the Jensen inequality gives:

$$\log p_\theta(X^0) \geq E_{X^1,\dots,X^T,Z\sim q(\cdot)}\left[\log \frac{p_\theta(X^0,\dots,X^T,Z)}{q(X^1,\dots X^T,Z|X^0)}\right] =: \mathcal{L}(\theta,\mu,\tau,\pi). \tag{17}$$

We have derived one variational lower bound $\mathcal{L}(\cdot)$ from the marginal log-likelihood, we now proceed by optimising the lower bound w.r.t. the parameters.

Optimisation

In this subsection, we delve into the optimization process for our model. We start by presenting a theorem that decomposes the variational lower bound.

Proposition 1. *The variational lower bound can be decomposed as following:*

$$\begin{aligned}\mathcal{L}(\cdot) = &\sum_{i,q}\tau_{iq}\left[-\frac{1}{2}\left|\left|\sqrt{\bar{\alpha}_T}\mu_q\right|\right|_2^2 - \sum_{t>1}E_{X^t}\left[\frac{1}{2\bar{\delta}_t}(\left|\left|\tilde{\mu}(X_i^t,X_i^0,\mu_q) - \mu_\theta(X_i^t,t,\mu_q)\right|\right|_2^2)\right] + \log\pi_q - \log\tau_{i,q}\right] \\ &+ \sum_{i,q}\tau_{iq}E_{X^1}\log p_\theta(X_i^0|X_i^1,\mu_q)\end{aligned} \tag{18}$$

where

$$\begin{aligned}&E_{X^t} \triangleq E_{X^t\sim q} \quad \textit{and} \\ \tilde{\mu}(X_i^t,X_i^0,\mu_q) = &\frac{1-m_t}{1-m_{t-1}}\frac{\delta_{t-1}}{\delta_t}\sqrt{\alpha_t}X_i^t + (1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t}\sqrt{\bar{\alpha}_{t-1}}X_i^0 \\ &+ \left(m_{t-1}\delta_t - \frac{m_t(1-m_t)}{1-m_{t-1}}\alpha_t\delta_{t-1}\right)\frac{\sqrt{\bar{\alpha}_{t-1}}}{\delta_t}\mu_q.\end{aligned} \tag{19}$$

Note that by setting β_1 to a very small value,one has $X_i^1 \sim X_i^0$ so that the last term of $\mathcal{L}$ can be neglected.
Moreover, since $X_i^t|X_i^0 \sim \mathcal{N}((1-m_t)\sqrt{\bar{\alpha}_t}X_i^0 + m_t\sqrt{\bar{\alpha}_t}\mu_q; \delta_t I_D)$, then we can re-parameterise X_i^t as:

$$X_i^t = (1-m_t)\sqrt{\bar{\alpha}_t}X_i^0 + m_t\sqrt{\bar{\alpha}_t}\mu_q + \sqrt{\delta_t}\epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}(0, I_D).$$

Then any expectation with respect to X_i^t can be transformed to an expectation with respect to the couple (X_i^0, ϵ_t) (the proof of this can easily be obtained by a change of variable on the integral).
Now we have to fix an architecture for the neural network μ_θ.

Architecture of μ_θ:
Notice that the optimal value for $\mu_\theta(X_i^t, t, \mu_q)$ is $\tilde{\mu}(X_i^t, X_i^t, \mu_q)$ (in Eq 19)

As μ_θ allows us to remove the correct amount of noise to pass from latent image X_i^t to X_i^{t-1} and given that this later was added linearly in the forward process (see the functional form of $q(X_i^t|X_i^{t-1})$), we suppose that μ_θ has the following functional form:

$$\mu_\theta(X_i^t, t, \mu_q) = c_{X^t} X_i^t + c_{\mu_q}\mu_q + c_{\eta_\theta}\eta_\theta(X_i^t, t, \mu_q)$$

where η_θ is a neural network and

$$\begin{aligned} c_{X_t} &= \frac{1-m_t}{1-m_{t-1}}\frac{\delta_{t-1}}{\delta_t}\sqrt{\alpha_t} + (1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t}\frac{1}{\sqrt{\alpha_t}}. \\ c_{\mu_q} &= \left(m_{t-1}\delta_t - \frac{m_t(1-m_t)}{1-m_{t-1}}\alpha_t\delta_{t-1}\right)\frac{\sqrt{\bar{\alpha}_{t-1}}}{\delta_t}. \\ c_{\eta_t\theta} &= -(1-m_{t-1})\frac{\delta_{t|t-1}}{\delta_t\sqrt{\alpha_t}}\sqrt{1-\bar{\alpha}_t}. \end{aligned}$$

After some calculations, one gets

$$\tilde{\mu} - \mu_\theta = c_{\eta_t}\left(\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(m_t\sqrt{\bar{\alpha}_t}(\mu_q - X_i^0) + \sqrt{\delta_t}\epsilon_t\right) - \epsilon_\theta\right). \tag{20}$$

Hence
$\mathcal{L} \simeq \sum_{i,q}\tau_{iq}\left[-\frac{1}{2}\left|\left|\sqrt{\bar{\alpha}_T}\mu_q\right|\right|_2^2 - \sum_{t>1}E_{X_i^0,\epsilon}\left[\frac{c_{\eta_t}^2}{2\delta_t}\left|\left|\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(m_t\sqrt{\bar{\alpha}_t}(\mu_q - X_i^0) + \sqrt{\delta_t}\epsilon_t\right) - \eta_\theta\right|\right|_2^2\right] + \log\pi_q - \log\tau_{i,q}\right]$

3.1 Variational EM algorithm

Proposition 2. *The optimal updates of τ_{iq} (E-step) and π_q (M-step) are given by:*

$$\begin{aligned} \tau_{iq} &= \frac{\pi_q \exp\left(-\frac{1}{2}||\sqrt{\bar{\alpha}_T}\mu_q||_2^2 - \sum_{t>1}\frac{1}{2\delta_t}f_{\theta,i,t,q} - 1\right)}{\sum_{q=1}^Q \pi_q \exp\left(-\frac{1}{2}||\sqrt{\bar{\alpha}_T}\mu_q||_2^2 - \sum_{t>1}\frac{1}{2\delta_t}f_{\theta,i,t,q} - 1\right)}, \\ \pi_q &= \frac{N_q}{N}, \end{aligned} \tag{22}$$

where $N_q = \sum_i \tau_{iq}$ and

$$\begin{aligned} f_{\theta,i,t,q} &= E_{X_i^t}\left|\left|\tilde{\mu}(X_i^t, t, \mu_q) - \mu_\theta(X_i^t, t, \mu_q)\right|\right|_2^2 \\ &= E_{X_i^0,\epsilon_t}\left[\left[\frac{c_{\eta_t}^2}{2\bar{\delta}_t}\left|\left|\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(m_t\sqrt{\bar{\alpha}_t}(\mu_q - X^0) + \sqrt{\delta_t}\epsilon_t\right) - \epsilon_\theta(M(X_i^0, t, \epsilon_t), t, \mu_q)\right|\right|_2^2\right]\right], \end{aligned}$$

with $M(X_i^0, t, \epsilon_t) = (1-m_t)\sqrt{\bar{\alpha}_t}X_i^0 + m_t\sqrt{\bar{\alpha}_t}\mu_q + \sqrt{\delta_t}\epsilon_t$.

To optimize the model's parameters (M-step), we employ the VB-EM algorithm, as outlined in (Kipf and Welling, 2016), which is described in Algorithm 1.

We can see that the exact value of τ_{iq} cannot be explicitly determined due to the expectation inside $f_{\theta,i,t,q}$.

4 Conclusion

We have demonstrated the potential extension of diffusion models, a class of likelihood-based models with a stationary training objective, to the clustering field, akin to numerous other generative models. Unfortunately, due to time constraints and the unresolved issue regarding τ_{iq}, experimental validation was not feasible within this study. However, we intend to present these experiments directly during the presentation should our paper be accepted, providing a more comprehensive evaluation of our approach.

Algorithm 1: Training algorithm

while *not convergence, for each step* k **do**
  For $\boldsymbol{i = 1, \dots N}$
    For $\boldsymbol{q = 1, \dots Q}$
      Compute τ_{iq}^k and π_q^k according to Equation 22
  For $\boldsymbol{l = 1, 2, \dots N_{\text{iter}}}$ **do**
    For $\boldsymbol{q = 1, \dots Q}$ **do**
      Sample $t \sim \text{Uniform}\{1, \dots T\}$, $\epsilon_t \sim \mathcal{N}(0, I_D)$ and $i \sim \text{Uniform}\{1, \dots N\}$
      Compute $X_i^t = (1 - m_t)\sqrt{\bar{\alpha}_t} X_i^0 + m_t \sqrt{\bar{\alpha}_t}\mu_q + \sqrt{\delta_t}\epsilon_t$
      Take gradient step on

$$\nabla_{\theta,\mu_q}\tau_{iq}\left[\left|\left|\frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(m_t\sqrt{\bar{\alpha}_t}(\mu_q - X_i^0) + \sqrt{\delta_t}\epsilon_t\right) - \eta_\theta(X_{iq}^t, t, \mu_q)\right|\right|_2^2 + \frac{\bar{\alpha}_T}{2}\,||\mu_q||_2^2\right]$$

Bibliography

Creswell, Antonia et al. (2018). "Generative adversarial networks: An overview". In: *IEEE signal processing magazine* 35.1, pp. 53–65.

Dhariwal, Prafulla and Alexander Nichol (2021). "Diffusion models beat gans on image synthesis". In: *Advances in neural information processing systems* 34, pp. 8780–8794.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33, pp. 6840–6851.

Kipf, Thomas N and Max Welling (2016). "Variational graph auto-encoders". In: *arXiv preprint arXiv:1611.07308*.

Lu, Yen-Ju et al. (2022). "Conditional diffusion probabilistic model for speech enhancement". In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 7402–7406.