



HumanDiffusion: diffusion model using perceptual gradients

Yota Ueda, Shinnosuke Takamichi, Yuki Saito,
Norihiko Takamune, Hiroshi Saruwatari

The University of Tokyo, Japan.

weda@g.ecc.u-tokyo.ac.jp, shinnosuke_takamichi@ipc.i.u-tokyo.ac.jp

Abstract

We propose *HumanDiffusion*, a diffusion model trained from humans’ perceptual gradients to learn an acceptable range of data for humans (i.e., human-acceptable distribution). Conventional HumanGAN aims to model the human-acceptable distribution wider than the real-data distribution by training a neural network-based generator with human-based discriminators. However, HumanGAN training tends to converge in a meaningless distribution due to the gradient vanishing or mode collapse and requires careful heuristics. In contrast, our HumanDiffusion learns the human-acceptable distribution through Langevin dynamics based on gradients of human perceptual evaluations. Our training iterates a process to diffuse real data to cover a wider human-acceptable distribution and can avoid the issues in the HumanGAN training. The evaluation results demonstrate that our HumanDiffusion can successfully represent the human-acceptable distribution without any heuristics for the training.

Index Terms: diffusion models, human computation, black-box optimization, crowdsourcing, speech perception

1. Introduction

Generative models can produce data that are indistinguishable from real data. In particular, deep generative models [1, 2, 3, 4], which are based on deep neural networks (DNNs), have significantly improved the quality of generated data in media research, including speech synthesis [5, 6, 7], natural language processing [8], and image synthesis [9]. The generative models can represent real-data distributions and produce data that follow the learned distributions [1]. In contrast, humans may accept data as natural even when the data are outliers of a real-data distribution [10]. For example, in speech perception, humans can accept synthesized or processed speech. In this paper, we use the term *human-acceptable distribution* defined as the data distribution whose data humans can accept as natural [10].

HumanGAN [10] was proposed to model a human-acceptable distribution, whereas the basic generative adversarial network (GAN) [1] can only represent a real-data distribution. GAN is a type of deep generative model and consists of a DNN-based generator and a DNN-based discriminator. HumanGAN replaces the discriminator of GAN with a human-based discriminator. HumanGAN regards humans as black-box systems that output perceptual evaluation values, given the generated data. By using the estimated perceptual gradient of the human-based discriminator, one can then train a generator to represent a human-acceptable distribution. However, HumanGAN often suffers from gradients vanishing and mode collapse during training similar to the original GAN [11]. In other words, it requires careful heuristics to prevent the learned distribution from converging in a meaningless region, such as outside the human-acceptable distribution or around data whose evaluation

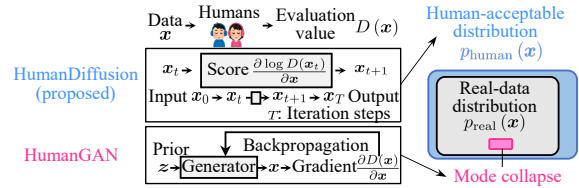


Figure 1: *HumanDiffusion and HumanGAN. HumanDiffusion can represent a human-acceptable distribution whereas HumanGAN cannot without a heuristic solution.*

value is extremely high (i.e., the mode of perceptual evaluation).

To solve these problems, we propose *HumanDiffusion*, a diffusion model using perceptual gradients. For modeling a human-acceptable distribution, we introduce diffusion models [12] to sample data that follows a real-data distribution using the gradient of the distribution. **Fig. 1** shows a comparison of HumanDiffusion and HumanGAN, and the distributions these models represent. On the basis of a diffusion model, our HumanDiffusion iterates a process to diffuse real data to cover a wider human-acceptable distribution thereby avoiding the issues encountered during the HumanGAN training. HumanDiffusion is evaluated in terms of its phoneme perception. The experimental results show that HumanDiffusion can successfully represent a human-acceptable distribution.

2. Related Work

Human perception is incorporated into DNNs such as reinforcement learning [13], and genetic algorithm [14]. HumanGAN [10] deals with perceptual gradients to train a generator. Because humans are better at relative evaluation than absolute evaluation, gradient-based methods can train DNNs well.

2.1. HumanGAN

HumanGAN [10] represents the human-acceptable distribution $p_{\text{human}}(\mathbf{x})$, where \mathbf{x} is data, that is wider than the real-data distribution $p_{\text{real}}(\mathbf{x})$. A DNN-based generator of HumanGAN is trained using a human-based discriminator (humans’ perceptual evaluation) instead of the DNN-based discriminator of GAN [1]. Let N be the number of data. The generator $G(\cdot)$ transforms prior data $\{z_n\}_{1 \leq n \leq N}$ produced from a prior distribution $\pi(z)$ to data $\{\hat{x}_n\}_{1 \leq n \leq N}$. The human-based discriminator $D(\cdot)$ imitates humans’ perceptual evaluations. $D(\cdot)$ takes \hat{x}_n as an input and outputs a posterior probability that the input is perceptually acceptable. The objective function is

$$L_{\text{HumanGAN}} = \sum_{n=1}^N D(G(z_n)). \quad (1)$$

$G(\cdot)$ is trained to maximize L_{HumanGAN} . A model parameter θ of $G(\cdot)$ is iteratively updated as $\theta \leftarrow \theta + \alpha \partial L_{\text{HumanGAN}} / \partial \theta$,

where α is the learning rate, and $\partial L_{\text{HumanGAN}}/\partial\theta = \partial L_{\text{HumanGAN}}/\partial\mathbf{x} \cdot \partial\mathbf{x}/\partial\theta = \partial D(\mathbf{x})/\partial\mathbf{x} \cdot \partial\mathbf{x}/\partial\theta$.

$\partial\mathbf{x}/\partial\theta$ can be estimated analytically, but $\partial D(\mathbf{x})/\partial\mathbf{x}$ cannot because the human-based discriminator $D(\cdot)$ is not differentiable. HumanGAN uses the natural evolution strategy (NES) [15] algorithm to approximate the gradient. A small perturbation $\Delta\mathbf{x}_n^{(i)}$ is randomly generated from the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma_{\text{NES}}^2 \mathbf{I})$, and it is added to the generated data $\hat{\mathbf{x}}_n$. i is the perturbation index ($1 \leq i \leq I$). σ_{NES} and \mathbf{I} are the standard deviation and the identity matrix, respectively. Next, a human observes two perturbed data $\{\hat{\mathbf{x}}_n + \Delta\mathbf{x}_n^{(i)}, \hat{\mathbf{x}}_n - \Delta\mathbf{x}_n^{(i)}\}$ and evaluates their difference in the posterior probability of naturalness:

$$\Delta D(\hat{\mathbf{x}}_n^{(i)}) \equiv D(\hat{\mathbf{x}}_n + \Delta\mathbf{x}_n^{(i)}) - D(\hat{\mathbf{x}}_n - \Delta\mathbf{x}_n^{(i)}). \quad (2)$$

$\Delta D(\hat{\mathbf{x}}_n^{(i)})$ ranges from -1 to 1 . For instance, a human will answer $\Delta D(\hat{\mathbf{x}}_n^{(i)}) = 1$ when the human perceives that $\hat{\mathbf{x}}_n + \Delta\mathbf{x}_n^{(i)}$ is substantially more acceptable than $\hat{\mathbf{x}}_n - \Delta\mathbf{x}_n^{(i)}$. $\partial D(\hat{\mathbf{x}}_n)/\partial\mathbf{x}$ is approximated as [15]

$$\frac{\partial D(\hat{\mathbf{x}}_n)}{\partial\mathbf{x}} = \frac{1}{2\sigma_{\text{NES}}I} \sum_{i=1}^I \Delta D(\hat{\mathbf{x}}_n^{(i)}) \cdot \Delta\mathbf{x}_n^{(i)}. \quad (3)$$

However, HumanGAN requires various heuristics during training, such as initialization and training stops. First, HumanGAN should roughly know the shapes of the human-acceptable distribution in advance to make the training easier. Data that deviate significantly from the human-acceptable distribution cannot be distinguished by humans, resulting in an inaccurate estimation of the gradient in Eq. (3). Moreover, the generator training should be stopped before converging to represent the human-acceptable distribution because the training causes mode collapse due to the training based on the maximization of L_{human} . Concretely, if the generator finds one region in the data space where humans evaluate them as highly acceptable during the training, the learned distribution converged in a limited range of the whole human-acceptable distribution.

2.2. Diffusion models

The diffusion models [12] use the gradient of the data distribution to train a model that represents the real-data distribution $p_{\text{real}}(\mathbf{x})$. The score of the real-data distribution $p_{\text{real}}(\mathbf{x})$ is defined as $\frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$. The score network $S(\cdot)$ is parameterized by θ and trained to approximate the score.

Even if the true $p_{\text{real}}(\mathbf{x})$ is not observable, as long as the score $\frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$ is observable, an iterative update with Langevin dynamics [16] can be applied as follows. Langevin dynamics is a kind of Markov chain Monte Carlo (MCMC) method using the gradients of the data distributions. First, an input data \mathbf{x}_0 is sampled from a prior distribution $\pi(\mathbf{z})$. Then, the data at $(t+1)$ th step ($t = 0, 1, \dots, T-1$) is calculated based on the Langevin dynamics using the score:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\epsilon^2}{2} \frac{\partial \log p_{\text{real}}(\mathbf{x}_t)}{\partial\mathbf{x}} + \epsilon \mathbf{r}, \quad (4)$$

where $\epsilon > 0$ and \mathbf{r} are a fixed step size and random vector from the isotropic Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, respectively. Finally, the distribution of the output data at the final step T , \mathbf{x}_T , will equal $p_{\text{real}}(\mathbf{x})$ when $\epsilon \rightarrow 0$ and $T \rightarrow \infty$. Because sampling from Eq. (4) only requires the score $\frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$, the score network $S(\mathbf{x}) \approx \frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$ can be trained, and it can then approximately produce samples with Langevin dynamics. Given real

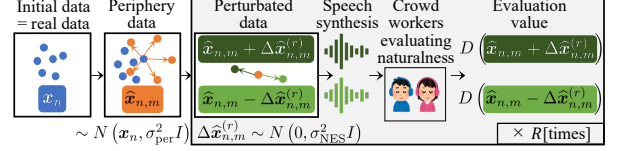


Figure 2: Tasks for crowdworkers to estimate the score. Periphery data are sampled from real data. A human observes two perturbed data and evaluates their perceptual difference in naturalness acceptability. The score is estimated using evaluation and perturbation.

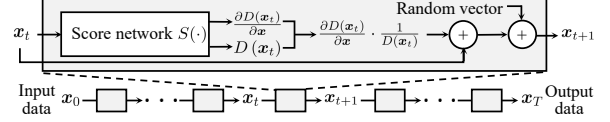


Figure 3: Training score network and inference in HumanDiffusion. The score network outputs gradients and evaluation values to calculate scores. Samples are produced with Langevin dynamics.

data $\{\mathbf{x}_n\}_{1 \leq n \leq N}$, the score $\frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$ is predicted by using $S(\cdot)$.

The score network $S(\mathbf{x}; \theta)$ for estimating $\frac{\partial \log p_{\text{real}}(\mathbf{x})}{\partial\mathbf{x}}$ is trained by score matching without training a model to estimate $p_{\text{real}}(\mathbf{x})$. The objective function for training $S(\cdot)$ is

$$L_{\text{Diffusion}} = \sum_{n=1}^N \left| S(\mathbf{x}_n; \theta) - \frac{\partial \log p_{\text{real}}(\mathbf{x}_n)}{\partial\mathbf{x}} \right|^2. \quad (5)$$

As explained above, the diffusion model can generate data that follow a real-data distribution by using scores of the real-data distribution. If we can observe scores of the human-acceptable distribution instead of the real-data distribution, and if the score network can learn them, we expect to be able to generate data according to the human-acceptable distribution.

3. HumanDiffusion

3.1. Components of HumanDiffusion

Fig. 2 shows the process of HumanDiffusion. HumanDiffusion uses the gradient of perceptual evaluation to train a model representing the human-acceptable distribution $p_{\text{human}}(\mathbf{x}) = \frac{1}{Z} D(\mathbf{x})$, where $Z = \int D(\mathbf{x}) d\mathbf{x}$ is the normalization coefficient. The score is defined as $\frac{\partial \log D(\mathbf{x})}{\partial\mathbf{x}}$. We process periphery data sampled from real data \mathbf{x}_n to estimate the score around the real-data distribution. Let M be the number of periphery data per real data. We sample periphery data $\{\hat{\mathbf{x}}_{n,m}\}_{1 \leq n \leq N, 1 \leq m \leq M}$ from $\mathcal{N}(\mathbf{x}_n, \sigma_{\text{per}}^2 \mathbf{I})$, where σ_{per} is the standard deviation. The perceptual evaluation function $D(\cdot)$ is driven by humans and takes data \mathbf{x} as the input and outputs the perceptual evaluation value of the naturalness of the data in the range of $[0, 1]$. If the data is natural, the evaluation value is 1, and if the data is unnatural, the evaluation value is 0. The score network $S(\cdot)$ is trained to approximate the score of $D(\mathbf{x})$. The objective function is defined as the minimization of

$$L_{\text{HumanDiff}} = \sum_{n=1}^N \sum_{m=1}^M \left| S(\mathbf{x}_{n,m}; \theta) - \frac{\partial \log D(\hat{\mathbf{x}}_{n,m})}{\partial\mathbf{x}} \right|^2. \quad (6)$$

3.2. Estimating score of perceptual evaluation

Similar to the HumanGAN, the score $\frac{\partial \log D(\mathbf{x})}{\partial\mathbf{x}}$ cannot be estimated because the evaluation functions are based on the perceptual evaluation. We use the NES algorithm to estimate the

score. The NES algorithm uses evaluation values of two perturbed data. With the index i , a human observes two perturbed data $\hat{\mathbf{x}}_{n,m} + \Delta \mathbf{x}_{n,m}^{(i)}$, $\hat{\mathbf{x}}_{n,m} - \Delta \mathbf{x}_{n,m}^{(i)}$, where $\Delta \mathbf{x}_{n,m}$ is sampled from $N(0, \sigma_{\text{NES}}^2 \mathbf{I})$ in the same way as Eq. (3). In HumanGAN, humans responded only to the difference between the two samples (Eq. (2)), but in HumanDiffusion, humans responded to the absolute value of naturalness for each of these two samples.

To estimate the score using the NES algorithm in the same manner as using HumanGAN, we adopt the chain rule to the score: $\frac{\partial \log D(\mathbf{x})}{\partial \mathbf{x}} = \frac{1}{D(\mathbf{x})} \frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$. The score is obtained by estimating $D(\mathbf{x})$ and $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$. We remodel the score network $S(\cdot)$ to output $S_D(\cdot) \approx D(\mathbf{x})$ and $S_{\nabla D}(\cdot) \approx \frac{\partial D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}}$ instead of $\frac{\partial \log D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}}$, as shown in Fig. 3. If the dimension of the data is d , this network outputs values in $d + 1$ dimensions. The loss function of the score network (Eq. (6)) is redefined as

$$L_{\text{HumanDiff}}^{\text{unconditional}} = \sum_{n=1}^N \sum_{m=1}^M (|S_D(\hat{\mathbf{x}}_{n,m}) - D(\hat{\mathbf{x}}_{n,m})|^2 + \left| S_{\nabla D}(\hat{\mathbf{x}}_{n,m}) - \frac{\partial \log D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}} \right|^2). \quad (7)$$

$\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$ can be approximated in the same manner as in HumanGAN:

$$\frac{\partial D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}} = \frac{1}{2\sigma_{\text{NES}} I} \sum_{i=1}^I \left\{ D(\hat{\mathbf{x}}_{n,m} + \Delta \mathbf{x}_{n,m}^{(i)}) - D(\hat{\mathbf{x}}_{n,m} - \Delta \mathbf{x}_{n,m}^{(i)}) \right\} \cdot \Delta \mathbf{x}_{n,m}^{(i)}. \quad (8)$$

Next is the estimation of $D(\mathbf{x})$, where $D^{(i)}(\hat{\mathbf{x}}_{n,m})$ at the index i is given by $D^{(i)}(\hat{\mathbf{x}}_{n,m}) = (D(\hat{\mathbf{x}}_{n,m} + \Delta \mathbf{x}_{n,m}^{(i)}) + D(\hat{\mathbf{x}}_{n,m} - \Delta \mathbf{x}_{n,m}^{(i)})) / 2$, and $D(\hat{\mathbf{x}}_{n,m})$ is estimated from it. The most intuitive method is to estimate the mean $D(\hat{\mathbf{x}}_{n,m}) = \frac{1}{I} \sum_{i=1}^I D^{(i)}(\hat{\mathbf{x}}_{n,m})$. However, since $D^{(i)}(\hat{\mathbf{x}}_{n,m})$ is a defined range of values, the distribution of $D^{(i)}(\mathbf{x}_{n,m})$ is not symmetric near its lower bound (0, i.e., the perceived quality of the data is extremely poor) or, its upper bound (1, the perceived quality is extremely good). Therefore, the mean is not a suitable representative value of the distribution. Therefore, in this paper, we use kernel distribution estimation given $P(D^{(i)}(\hat{\mathbf{x}}_{n,m}))$ and consider its mode as $D(\hat{\mathbf{x}}_{n,m})$. Preliminary experiments were conducted using both the mean and mode methods, and the mean did not show the sampling convergence.

From the above formulation, the score network is a network that outputs both $D(\mathbf{x})$ and $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$, as shown in Fig. 3. If the dimension of the data is d , this network outputs values in $d + 1$ dimensions. The score network is trained using Eq. (6) by computing $\frac{\partial \log D(\mathbf{x})}{\partial \mathbf{x}}$ from the output $D(\mathbf{x})$ and $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$.

3.2.1. Compensating for inaccurate gradient

If two perturbed data of extremely poor quality (e.g., sounds that do not sound like a speech) are observed and evaluated by a human, the score is very inaccurate¹. Therefore, we assume that the naturalness of the real data is not poor at all, and we

¹Intuitively, it is difficult to evaluate the naturalness of two extremely poor quality data. Therefore, the score of periphery data $\mathbf{x}_{n,m}$ of extremely poor quality cannot be estimated from the gradient alone.

apply a regularization that brings the gradient $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$ closer to that of the real data used to sample the periphery data as

$$\frac{\partial D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}} \leftarrow \frac{\partial D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}} + b(\hat{\mathbf{x}}_{n,m} - \mathbf{x}_n), \quad (9)$$

where b is a hyperparameter. This method is based on an existing diffusion model that robustly estimates scores in the absence of any data [12]. Preliminary experiments have confirmed that without this regularization, the scores for extremely poor-quality data are very close to zero, and the sampling described below does not converge.

3.3. Sampling with Langevin dynamics

The score is computed using Eqs. (8) and (9) and modeled them using Eq. (6). Langevin dynamics is used to sample the data according to the human-acceptable distribution $p_{\text{human}}(\mathbf{x})$.

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{\epsilon^2}{2} S(\mathbf{x}_t) + \epsilon \mathbf{r}. \quad (10)$$

The initial distribution that \mathbf{x}_0 follows is a real-data distribution (e.g., data generated by a trained GAN). The training iteration is continued until convergence.

4. Experiments and Results

4.1. Experimental setup

Data space and speech analysis-synthesis. Phoneme perception experiments were conducted as described in a previous report on HumanGAN using basically the same experimental setup [10]. We basically followed the experimental setup of the HumanGAN paper [10], specifically, we used the phoneme (Japanese /a/), the JVPD corpus [17], preprocessing, the WORLD vocoder [18, 19], and speech features including log spectral envelopes. We applied principal component analysis (PCA) to the log spectral envelopes and used the first and second principal components. The two-dimensional principal components were normalized to have zero-mean and unit-variance. The speech synthesis step also followed the previous report on HumanGAN paper [10]. First, the first and second principal components were generated by the neural network and de-normalized. Then, a 1-second speech sample was synthesized using the remaining speech features, i.e., F0, and aperiodicities. The evaluation was performed by using the Lancers crowdsourcing platform [20].

Perceptual test. We carried out perceptual evaluations. Two speech waveforms generated from $\hat{\mathbf{x}}_{n,m} + \Delta \mathbf{x}_{n,m}^{(i)}$, $\hat{\mathbf{x}}_{n,m} - \Delta \mathbf{x}_{n,m}^{(i)}$ were presented to a listener, and the listener provided the naturalness values $D(\hat{\mathbf{x}}_{n,m} + \Delta \mathbf{x}_{n,m}^{(i)})$ and $D(\hat{\mathbf{x}}_{n,m} - \Delta \mathbf{x}_{n,m}^{(i)})$ using slide bars. The slide bars range from 0 to 1. Listeners selected 1 if the speech was natural and 0 when the speech was unnatural. The total number of listeners was 150.

Score network and HumanDiffusion sampling. The score network was a feed-forward neural network. It inputs two-dimensional data \mathbf{x} and outputs one-dimensional $D(\mathbf{x})$ and two-dimensional $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$. The network consisted of a two-unit input layer, 3×128 -unit softplus [21] hidden layers, and three-unit sigmoid (for $D(\mathbf{x})$) and linear (for $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$) output layers. We used the Adam [22] optimizer with a learning rate of $\alpha = 0.001$ for the training. The number of real data N , the number of periphery data per real datum M , the number of perturbations I , the number of training iterations, and the standard deviation of NES σ_{NES} , the standard deviation σ_{per} were set to

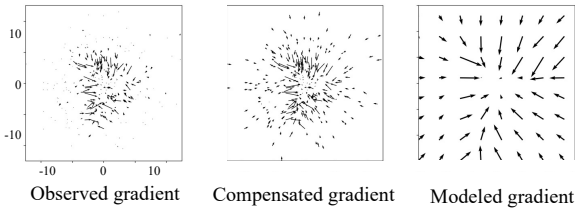


Figure 4: *Observed gradients, gradients with regularization, and modeled gradient by score network.*

100, 3, 20, 10000, 1.0 and 10, respectively. For the Langevin dynamics sampling, the number of data sampled from real-data distribution $N(0, 1)$, step size ϵ , the number of iterations were set to 200, 0.0001, 100000. We confirmed that the distribution converged after 10, 000 iterations of the sampling.

HumanGAN. For comparison, we also trained HumanGAN. The generator was a feed-forward neural network. The model consisted of a two-unit input layer, 3×128 -unit softplus hidden layers, and a two-unit linear output layer. Although the original training of HumanGAN requires listening experiments to determine the perceptual gradient for each iteration, in this study, we used the output of the score network to reduce the cost of experiments. Specifically, the output $\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$ of the score network was used for training. We used the Adam optimizer with a learning rate of $\alpha = 0.01$ for the training. The number of training iterations was set to 10000. The initial parameters were set to cover a neighborhood of the perceptual distribution as described in the report on HumanGAN [10].

4.2. Qualitative evaluation of score network training

First, we qualitatively confirm the observed scores and the score network training. The left panel in Fig. 4 is the estimated gradient for each $\frac{\partial D(\hat{\mathbf{x}}_{n,m})}{\partial \mathbf{x}}$. As described in Section 3, the real data distribution follows a mean of 0 and a variance of 1. This figure shows that a gradient occurs over a wider range than that of the real data distribution. In other words, as shown in the HumanGAN paper, the perceptual distribution exists over a wider range than the real data distribution. On the other hand, when the data are outliers far from the real data distribution (e.g., when the x-axis value exceeds ± 5), the gradient is almost zero. This indicates that the perceptual gradient has vanished owing to extremely poor sound quality. Regularization compensates for this gradient vanishing, and the result is shown in the middle panel of Fig. 4. This regularization produces a gradient toward the real data distribution even for data that are outliers. The result of training this gradient in the score network is shown on the right panel of the figure. Finally, a gradient toward the center is learned.

4.3. Quantitative/qualitative evaluation of sampling

Next, we compare the data generated by HumanGAN and HumanDiffusion to show that the HumanDiffusion sampling is reasonable. Figs. 5 a) and b) show the distributions of data generated by HumanGAN and HumanDiffusion, respectively. The gradients are also shown for reference.

HumanGAN learns only to move data in the gradient direction, which causes mode collapse, as shown on the right panel of Fig. 5 a) unless a heuristic iteration limit is set. In contrast, HumanDiffusion includes a stochastic movement term based on MCMC and thus captures a wider range than the real data distribution, as shown on the right panel of Fig. 5 b). The results demonstrate that HumanDiffusion can learn a wider

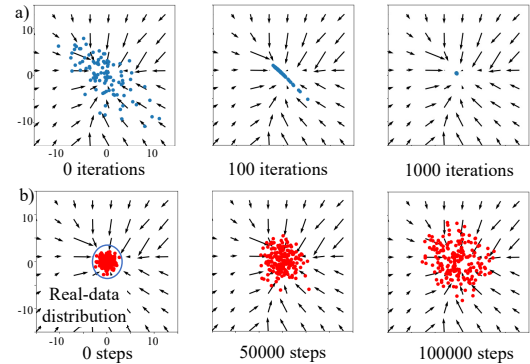


Figure 5: *Data generated by a) HumanGAN and b) HumanDiffusion.*

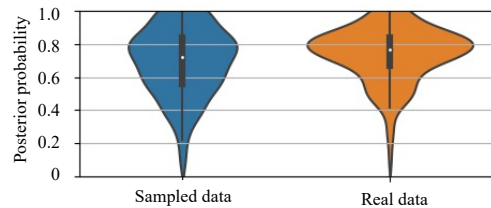


Figure 6: *Violin plots of posterior acceptability. The white point indicates the average.*

human-acceptable distribution without introducing heuristics used in HumanGAN. The variance of the real data distribution is $[1.0, 1.0]$ (unit variance), whereas the variance of the data distribution obtained by HumanDiffusion is $[9.2, 8.9]$, indicating that HumanDiffusion represents a wider distribution than the real data distribution.

4.4. Evaluation using real and generated data

Finally, we quantitatively verified whether HumanDiffusion can generate samples from a human-acceptable distribution wider than a real-data distribution. We prepared two data sets: from the real-data and the HumanDiffusion distributions. The posterior probabilities of these data sets were evaluated using the MUSHRA test. The posterior probability ranged within $[0, 1]$ and corresponds to $D(\mathbf{x})$. The numbers of the data are 200 respectively. The total number of listeners was 40. A listener evaluated 20 samples.

Fig. 6 shows the violin plots of the posterior probability. The average of real-data distribution data is 73 and the average of generated data is 69. Therefore, we consider that the generated data are sufficiently natural, although the variation of the data distribution is much larger than that of the real-data distribution. It is clear that HumanDiffusion can represent a human-acceptable distribution.

5. Conclusions

In this paper, we presented HumanDiffusion, which can represent a human-acceptable (humans' perceptually recognizable) distribution. We evaluated the use of HumanDiffusion in modeling phoneme perception, and we qualitatively and quantitatively demonstrated that HumanDiffusion can represent a human-acceptable distribution. As our future work, we will examine the scalability of the HumanDiffusion in terms of feature dimensionality.

Acknowledgements: This work was supported by JSPS KAKENHI 23H03418, 21H04900, and JST FOREST JP-MJFR226V.

6. References

- [1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.
- [2] D. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv*, vol. abs/1312.6114, 2013.
- [3] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," in *Proc. ICLR*, San Diego, U.S.A., 2015.
- [4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 684–10 695.
- [5] Y. Saito, S. Takamichi, and H. Saruwatari, "Statistical parametric speech synthesis incorporating generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 84–96, 2018.
- [6] Y. A. Li, A. Zare, and N. Mesgarani, "Starganv2-vc: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion," *arXiv preprint arXiv:2107.10394*, 2021.
- [7] E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone," in *International Conference on Machine Learning*. PMLR, 2022, pp. 2709–2720.
- [8] L. Floridi and M. Chiriatti, "Gpt-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, pp. 681–694, 2020.
- [9] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8188–8197.
- [10] K. Fujii, Y. Saito, S. Takamichi, Y. Baba, and H. Saruwatari, "Humangan: generative adversarial network with human-based discriminator and its evaluation in speech perception modeling," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6239–6243.
- [11] Z. Zhang, M. Li, and J. Yu, "On the convergence and mode collapse of gan," in *SIGGRAPH Asia 2018 Technical Briefs*, 2018, pp. 1–4.
- [12] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.
- [13] N. Lambert, L. Castricato, L. von Werra, and A. Havrilla, "Illustrating reinforcement learning from human feedback (rlhf)," *Hugging Face Blog*, 2022, <https://huggingface.co/blog/rlhf>.
- [14] H. Asadi, S. Mohamed, K. Nelson, S. Nahavandi, and D. R. Zadeh, "Human perception-based washout filtering using genetic algorithm," in *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part II 22*. Springer, 2015, pp. 401–411.
- [15] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. ICML*, vol. 2, Stockholm, Sweden, 2018, pp. 2137–2146.
- [16] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *International Conference on Machine Learning*, 2011, pp. 681–688.
- [17] "Vowel database: Five Japanese vowels of males, females, and children along with relevant physical data (JVPD)," <http://research.nii.ac.jp/src/en/JVPD.html>.
- [18] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE transactions on information and systems*, vol. E99-D, no. 7, pp. 1877–1884, 2016.
- [19] M. Morise, "D4C, a band-a-periodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.
- [20] "Lancers," <https://www.lancers.jp/>.
- [21] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," *arXiv preprint arXiv:1612.02295*, 2016.
- [22] D. Kingma and B. Jimmy, "Adam: A method for stochastic optimization," in *arXiv preprint arXiv:1412.6980*, 2014.