We thank the reviewers for their valuable feedback. We first respond to three common questions: **A0.1:** We clarify that while TDPM adds a discriminator for $t$=$T$, it introduces no additional parameters into the generator of DDPM. Specifically, TDPM uses the same U-net in DDPM to not only generate $\mathbf{x}_T$ at $t$=$T$ from a random noise, but also transform $\mathbf{x}_{t+1}$ into $\mathbf{x}_t$ for $t < T$. In other words, a separate GAN generator is not needed. **A0.2:** We omitted the time comparison as the generation time is proportional to $T$ (the number of times of going through the U-net). We will clarify the time comparison in the revision. On a V100-32G GPU, the training time (s/epoch) of DDPM and TDPM is 116 vs. 194 (moderately increased) and generation time is linear in $T$:

| Diffusion steps $T$ | 1 | 2 | 4 | 50 | 100 | 1000 |
|---|---|---|---|---|---|---|
| DDPM/TDPM generation (s/img) | 0.033 | 0.057 | 0.127 | 1.521 | 3.132 | 31.029 |

**A0.3:** Comparing with DDGAN of Xiao et al. (ICLR 2022), the discovery of TDPM is resulted from our curiosity on what if we simply relax the endpoint of the diffusion chain to be a flexible implicit distribution while keeping the other parts unchanged. While both leveraging adversarial training and using the same StyleGAN2 discriminator, TDPM is distinct from DDGAN: *i)* Mathematically, DDGAN modifies $\sum_{t=1}^{T}\mathcal{L}_{t-1}$, while TDPM modifies $\mathcal{L}_T$ of the DDPM loss $\sum_{t=1}^{T}\mathcal{L}_{t-1}+\mathcal{L}_T$. Intuitively, keeping the diffusion ending point as isotropic Gaussian, DDGAN increases the diffusion rate to shorten the chain, at the expense of requiring an implicit model to fit every reverse step; By contrast, keeping the diffusion rates of the intermediate steps unchanged, TDPM modifies the diffusion target to an implicit distribution that is modeled by transforming random noise with the same U-net. Therefore, a clear generalization is the hybrid of DDGAN on $\sum_{t=1}^{T}\mathcal{L}_{t-1}$ and TDPM on $\mathcal{L}_T$, which could work well by combining the recently released DDGAN code and our code. *ii)* TDPM reuses the same U-net of DDPM as its generator, while DDGAN introduces several structural modifications to that U-net. *iii)* As $T$ increases, TDPM in general improves its performance, while DDGAN, which recommends the use of $T$=4, may clearly deteriorate its performance. *iv)* We'd like to point out that DDGAN appeared in ArXiv in December 2021, only about six weeks before the ICML submission deadline, and its code is not publicly available until April 5, 2022. Thus we only quoted the result of DDGAN with $T$=4 into comparison (our own implementation of DDGAN was not that successful). We also note while DiffuseVAE of Pandey et al. (2022) appeared in ArXiv less than one month before the ICML submission deadline and is outperformed by TDPM, we did include its results into Table 1.

**Reviewer #1** **Q1:** *Sampling diversity and training stability of GAN in TDPM.* **A1:** Not only does GAN help shorten the diffusion chain, the diffusion part also helps stabilize GAN. As illustrated by our visualizations (especially multi-mode cases in Figs. 6-7), increasing the number of diffusion steps helps diffuse the modes towards a more uni-modal distribution, which makes the GAN easier to train and mitigates the concern on mode collapse, especially if given a large enough $T$. Following your suggestion, we measure the recall on CIFAR-10 (the rebuttal time is too short for us to finish ImageNet experiments) and get 0.57 for both T=4 and 50, indicating the diversity is on par with DDPM(T=1000). **Q2:** *Inability of GANs on high-frequency details.* **A2:** $\mathbf{x}_T$ is a white Gaussian noise corrupted image, which actually loses increasingly more high-frequency details as $T$ increases due to noise corruption, making it easier to train a GAN. We note StyleGAN2 w/ ADA has utilized various data augmentation to boost its performance; a fairer comparison is with StyleGAN2 w/o ADA. **Q3:** *# of parameters.* **A3:** Please see **A0.2**. **Q4:** *Experiments on NLL-favor setting.* **A4:** Incorporating NLL related loss into TDPM is interesting but beyond the scope of this paper.

**Reviewer #2** **Q1:** *DDGAN and DiffuseVAE.* **A1:** Please see **A0.3**. **Q2:** *The extreme case of T=1 and 2.* On CIFAR-10, the FIDs under $T = 1$ (no diffusion) and $T = 2$ (one diffusion step) are 6.94 and 4.61, respectively. **Q3:** *# of samples used in NLL and KL computation.* **A3:** We follow DDPM to use the testing set (10k samples) to calculate NLL. The empirical KL divergence in toy experiments are calculated with 2,000 samples, using 20 grids within $[-10, 10]$; note this is not required for model training. **Q4:** *Loss and Gain.* **A4:** Main Loss: the training becomes more expensive, as shown in A0.2; Main Gain: the diffusion chain is significantly shortened for much faster generation.

**Reviewer #3** **Q1:** *Computation time.* **A1:** The time is comparable given same $T$, but TDPM can work well with a significantly smaller $T$. **Q2:** *Requirement on $D$.* We attribute the low sensitivity to the choice of $D$ to: 1) a diffused distribution is more uni-modal and hence easier to fit, and 2) the initial generation is refined during reverse diffusion. **Q3:** *$1/T$ and $\lambda$.* **A3:** Since $\mathcal{L}_{simple}$ is written as an expectation rather than a summation of $T$ terms, we add $\frac{1}{T}$ in front of $\mathcal{L}_T$. We set $\lambda = T$ to more strongly emphasize the training at step $T$. Tuning it could lead to further improved performance. **Q4:** *Results in Table 2.* **A4:** Overall we expect a larger $T$ to have a better FID, but there could be exceptions. **Q5:** *Line 136.* **A5:** DDGAN takes this approach and a key challenge is that the generator needs to fit different multimodal distributions at different $t$, which is probably why its $T$ is restricted to be small ($e.g.$, $T = 4$).

**Reviewer #4** Please see **A0.1-3** for your Q1-3 and two highlighted concerns. **Q4:** *$T$ and baselines.* **A4:** We choose these $T$ and group with similar NFE to facilitate comparisons with baselines. We will add TDPM ($T$=1) and re-organize Table 1. Considering our model mainly aims to improve DDPM that already performs very well, we mainly compare with DDPM on higher resolution images as a further justification besides CIFAR-10 (Note diffusion models demand costly computation under high-resolution).