Self-Correcting Self-Consuming Loops for Generative Model Training

Nate Gillman [1] **Michael Freeman** [1] **Daksh Aggarwal** [1] **Chia-Hong Hsu** [1]
Calvin Luo [1] **Yonglong Tian** [2] **Chen Sun** [1 2]

Abstract

As synthetic data becomes higher quality and proliferates on the internet, machine learning models are increasingly trained on a mix of human- and machine-generated data. Despite the successful stories of using synthetic data for representation learning, using synthetic data for generative model training creates "self-consuming loops" which may lead to training instability or even collapse, unless certain conditions are met. Our paper aims to stabilize self-consuming generative model training. Our theoretical results demonstrate that by introducing an idealized correction function, which maps a data point to be more likely under the true data distribution, self-consuming loops can be made *exponentially* more stable. We then propose self-correction functions, which rely on expert knowledge (e.g. the laws of physics programmed in a simulator), and aim to approximate the idealized corrector automatically and at scale. We empirically validate the effectiveness of self-correcting self-consuming loops on the challenging human motion synthesis task, and observe that it successfully avoids model collapse, even when the ratio of synthetic data to real data is as high as 100%.

1. Introduction

Generative models have been used to synthesize training data for various learning tasks, to varying degrees of success. For example, for the tasks of image classification and contrastive representation learning, recent work (Azizi et al., 2023; Tian et al., 2023) finds that using data synthesized from generative models rivals using real data. Unfortunately, there is a gloomier outlook when attempting to generalize this framework to generative model training.

[1]Brown University [2]Google DeepMind. Correspondence to: Nate Gillman <nate_gillman@brown.edu>, Chen Sun <chensun@brown.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the authors.

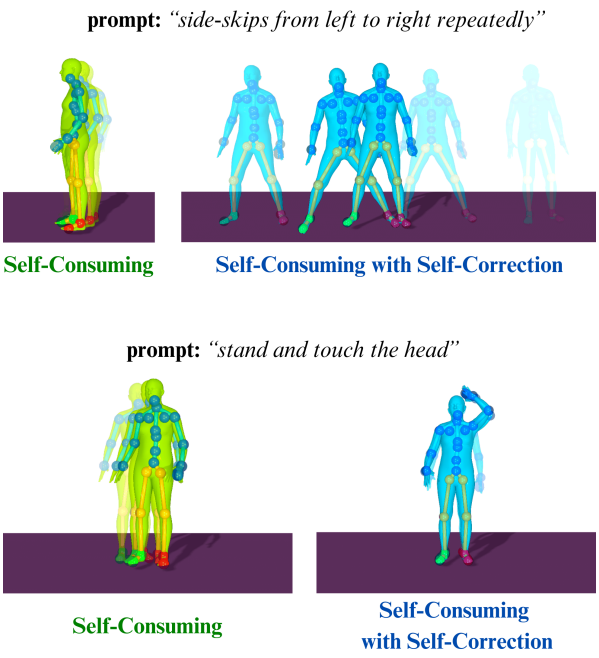


Figure 1. What happens after iteratively training a text-conditioned generative model for human motion synthesis for 50 generations? We simulate a self-consuming loop by creating synthetic data with the latest generative model, and mixing them with the original data to continue training the next generative model. We observe that by self-correcting the synthetic data with a physics simulator, the model can successfully avoid **collapse** and generate **high-quality human motion**. Faded poses represent poses from further back in time. Our paper provides theoretical and empirical justification for the self-correcting self-consuming loop.

On one hand, there is evidence to suggest that training a generative model with its own outputs in a self-consuming manner will lead to collapse (Alemohammad et al., 2024). For example, after 50 iterations of self-consuming training, a human motion diffusion model (Tevet et al., 2023) collapses and fails to follow the text prompts or the laws of physics (see the two examples on the left of Figure 1).

On the other hand, evidence suggests that such a framework could avoid collapse, but only when a "moderate" amount of synthetic data is used (Bertrand et al., 2024). Worse still, this self-consuming scenario might happen without us

knowing, and without us being able to quantify how much synthetic data is being used during training, due to the wide spread of AI generated content on the internet.

Intuitively, model collapse might be delayed or avoided by incorporating higher quality human generated data (Alemohammad et al., 2024), or by manually fixing the "mistakes" in machine created data. Considering the size of datasets used in practice (Schuhmann et al., 2022), neither of these options is a scalable solution.

In this paper, we aim to provide a theoretical analysis of how certain operations would avoid collapse in self-consuming loops, without any assumptions on the "moderateness" of synthetic data corruption. We introduce the mathematical abstraction of a *self-correction operation*. This operation maps synthesized data that are sampled from the generative model to data that are better representatives from the target probability distribution that the model is attempting to approximate. Instead of training on a combination of real data and synthesized data, we propose training on a combination of real data and synthesized *and then self-corrected* data. Note that injecting fresh human generated data can be viewed as a special case of this operation.

Our **main theoretical findings** (Theorem 4.3):

(1) The self-consuming model with self-correction is *exponentially* more stable than the self-consuming model without any self-correction.

(2) The self-correction procedure guarantees less unwanted variance during self-consuming model training.

In our theoretical study, we assume that correction is *ideal* in order to obtain rigorous performance guarantees. In our empirical study, we evaluate whether the same conclusions hold for *noisy* self-correction functions. We propose to automate this "self-correction" process by relying on programmed expert knowledge rather than a human-in-the-loop, such that the function can be applied at scale. We focus on the human motion synthesis task (Guo et al., 2022), and implement the self-correction function with a physics simulator-based imitation model (Luo et al., 2021). Our **empirical results** confirm that our theoretical findings hold in practice:

(1) As illustrated in Figure 1, the self-correcting self-consuming model generates higher-quality human motion than the one without any self-correction.

(2) The self-correction function allows self-consuming loops to avoid collapse even at a high synthetic data to real data ratio (e.g. 100%).

Our theory and experiments suggest that self-correction should stabilize self-consuming model training for any generative modeling task for which there exists a high quality "self-correction" function. We have released all the code associated with this paper.[1]

[1]Project page: https://nategillman.com/sc-sc.html

2. Related Work

2.1. Learning Representations with Synthetic Data

Real curated datasets are costly to obtain, so there has been much interest in generating synthetic data as training data for various vision tasks. Azizi et al. (2023) demonstrates that text-to-image diffusion models such as Imagen (Saharia et al., 2022) can generate synthetic examples that augment the ImageNet dataset for better image classification. He et al. (2023) studies how synthetic data from text-to-image models, when used exclusively, can be used as training data for image recognition tasks. Similarly, Tian et al. (2023) finds that using synthetic outputs from a text-to-image model results in contrastive models whose downstream performance rivals that of CLIP (Radford et al., 2021) on visual recognition tasks, including dense prediction. And the work in Jahanian et al. (2022) explored methods for multi-view representation learning by using the latent space of the generative models to generate multiple "views" of the synthetic data. The above works collectively provide evidence that *some* representation learning tasks, when trained on synthetic data from some *given* generative models, yield excellent results.

2.2. Training Generative Models on Synthetic Data

Another line of reseach investigates the use of synthetic data for training *generative* models. Shumailov et al. (2023) and Martínez et al. (2024) show that the use of model generated content in generative model training results in model degradation, likely because self-consuming loops remove low-density areas from the estimated probability manifold. Alemohammad et al. (2024) formalize three different kinds of self-consuming generative models: the fully synthetic loop, the synthetic augmentation loop, and the fresh data loop. In all of these loops, they iteratively re-train the model from scratch for every new generation. They empirically find that only the fresh data loop avoids model degradation.

Another recent work (Bertrand et al., 2024) considers the problem of iteratively fine-tuning in the context of synthetic augmentation loops. They find that self-consuming augmentation loops do not necessarily collapse, so long as the synthetic augmentation percentage is sufficiently low. The authors use techniques from the field of performative stability (Perdomo et al., 2020) to prove the existence of a convergence phenomenon in the space of model parameters. Our paper differs from prior work as we conduct analysis on self-consuming generative model training when the synthetic data can be optionally *corrected*. The correction can be performed with a human-in-the-loop, or by incorporating learned or programmed expert knowledge, as explored for natural language (Saunders et al., 2022; Welleck et al., 2023; Wu et al., 2023) and human motion (Yuan et al., 2023; Xu et al., 2023). We validate our theory with a practical self-correcting operations designed for image generation

and human motion synthesis tasks.

3. Overall Training Procedure

We describe our proposed procedure in concise language in Algorithm 1, and we explain it in more detail here. We train the zero'th generation from scratch on the ground truth dataset $\mathcal{D}_{\text{real}} := \{x_i\}_{i=1}^n$, and we stop training when the model is close to convergence. For all the following generations, we fine-tune the previous generation's latest checkpoint on a combination of the ground truth dataset $\mathcal{D}_{\text{real}}$, as well as $\lfloor \lambda \cdot n \rfloor$ synthetic data points which are generated from the previous generation's latest checkpoint, and then passed through the *correction function* π_γ.

The correction function π_γ is parameterized by the *correction strength* $\gamma \in \mathbb{R}_{\geq 0}$, which controls how much influence the correction function has on the input data points towards increasing a given point's likelihood with respect to the target distribution. The other main hyperparameter $\lambda \in \mathbb{R}_{\geq 0}$ is the *synthetic augmentation percent*, and it controls the ratio of synthetic data to real data in each iteration of fine-tuning. When $\gamma = 0$, we recover iterative re-training with synthetic augmentation considered in (Bertrand et al., 2024). And if we choose the synthetic augmentation percent to be $\lambda = 0$, then each generation simply corresponds to fine-tuning the model on the same dataset that it was trained on initially.

We now use *iterative fine-tuning* interchangeably with the more general term self-consuming loop. We also consider the idealized correction function for our theoretical analysis, and a broader family of practical *correction functions* for different data types.

4. Theoretical Analysis

4.1. Preliminaries

We mostly follow the notation from (Bertrand et al., 2024), except for introducing the correction function π_γ. Let us denote by p_{data} the ground truth probability distribution that we want to train a generative model to estimate. Suppose we have some dataset $\mathcal{D}_{\text{real}} = \{x_i\}_{i=1}^n$ sampled from p_{data}. We write $\hat{p}_{\text{data}} = (1/n)\sum_{i=1}^n \delta_{x_i}$. More generally, we use a hat to denote the empirical distribution over finitely many samples from the corresponding distribution.

Suppose that we have a class of generative models parameterized by $\Theta \subset \mathbb{R}^d$. We denote by p_θ a probability distribution in this class with model parameters $\theta \in \Theta$. We define the optimal model parameters within this class to be

$$\theta^\star = \arg\max_{\theta' \in \Theta} \mathbb{E}_{x \sim p_{\text{data}}}[\log p_{\theta'}(x)], \tag{1}$$

where we break ties by minimizing $\|\theta^\star\|$. Typically, such optimal parameters yield a model $p_{\theta^\star}$ which closely approximates the oracle ground truth distribution p_{data}, but doesn't equal it exactly; accordingly, we define the Wasserstein-2 distance between the distributions to be

$$\varepsilon := d_W(p_{\theta^\star}, p_{\text{data}}). \tag{2}$$

The model weights for the first generation are naturally defined according to the optimization

$$\theta_0^n := \arg\max_{\theta' \in \Theta} [\mathbb{E}_{x \sim \hat{p}_{\text{data}}}[\log p_{\theta'}(x)]]. \tag{3}$$

This corresponds to training on the finite subset $\mathcal{D}_{\text{real}}$. Next, let us suppose that the model weights from generation t are denoted θ_t^n. We will formalize a procedure for updating these weights for the next generation to obtain θ_{t+1}^n. For this, we need to define our correction function, and then we will use it to define the weight update.

Definition 4.1. For any probability distribution, and for any $\gamma \in \mathbb{R}_{\geq 0}$, we define the *correction of strength* γ of distribution p_θ to be the distribution

$$\pi_\gamma p_\theta(x) := \frac{p_\theta(x) + \gamma p_{\theta^\star}(x)}{1 + \gamma}, \tag{4}$$

where $p_{\theta^\star}$ is defined in (1). For any augmentation percentage $\lambda \geq 0$, we define the *weight update* mapping to be

$$\begin{aligned} \pi_\gamma \mathcal{G}_\lambda^n(\theta) &:= \operatorname*{local\,argmax}_{\theta' \in \Theta} \hat{\mathcal{H}}(\theta, \theta') \\ &:= \operatorname*{local\,argmax}_{\theta' \in \Theta} \Big[\mathbb{E}_{x \sim \hat{p}_{\text{data}}}[\log p_{\theta'}(x)] \\ &\qquad + \lambda \mathbb{E}_{x \sim \widehat{\pi_\gamma p_\theta}}[\log p_{\theta'}(x)] \Big], \end{aligned} \tag{5}$$

where $\hat{p}_{\text{data}}$ and $\widehat{\pi_\gamma p_\theta}$ are empirical distributions of size n and $\lfloor \lambda \cdot n \rfloor$ respectively.

To continue our discussion from before, our iterative weight update is defined as $\theta_{t+1}^n := \pi_\gamma \mathcal{G}_\lambda^n(\theta_t^n)$.

Note that we use an global maximization in (3) when defining the initial parameters θ_0^n, but we use a local maximization when computing our parameter update in (5). This difference is analogous to the differences between how model weights update during initial training, where parameter updates are more global, and during fine-tuning, where parameter updates are more local.

4.1.1. Understanding the correction $\pi_\gamma p_\theta(x)$

For $\gamma = 0$, the correction mapping in (4) simplifies to $\pi_0 p_\theta = p_\theta$, which is just the original distribution; this corresponds to no correction at all. For $\gamma = 1$, it is $\pi_1 p_\theta = (p_\theta + p_{\theta^\star})/2$. And for $\gamma = \infty$, it is $\pi_\infty p_\theta = p_{\theta^\star}$, which corresponds to the optimal distribution. So as γ increases from 0 to ∞, the distribution $\pi_\gamma p_\theta$ has a likelihood profile that matches p_θ less, and $p_{\theta^\star}$ more. As $p_{\theta^\star}$ is the optimal

Algorithm 1 Iterative Fine-tuning of a Generative Model With Correction

Input: $\mathcal{D}_{\text{real}} := \{x_i\}_{i=1}^n$, $\mathcal{A}$, $\mathcal{A}_{\text{ft}}$, π_γ // ground truth data, learning procedure, fine-tuning procedure, correction function
Parameters: T, λ, γ // number of retraining iterations, proportion of generated data, correction strength
$p_{\theta_0} \leftarrow \mathcal{A}(\mathcal{D}_{\text{real}})$ // learn generative model from scratch on true data
for $t = 1$ **to** T **do**
  $\mathcal{D}_{\text{synth}} \leftarrow \{\pi_\gamma(\tilde{x}_i)\}_{i=1}^{\lfloor \lambda \cdot n \rfloor}$, with $\tilde{x}_i \sim p_{\theta_{t-1}}$ // sample $\lfloor \lambda \cdot n \rfloor$ synthetic data points, pass through correction function
  $p_{\theta_t} \leftarrow \mathcal{A}_{\text{ft}}(\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}; p_{\theta_{t-1}})$ // fine-tune previous generation using augmented dataset
end for
Return $[p_{\theta_0}, p_{\theta_1}, p_{\theta_2}, \ldots, p_{\theta_T}]$

model in our generative model class, this means that as γ increases from 0 to ∞, we have that $\pi_\gamma p_\theta(x)$ is a PDF which better represents the target likelihood that we want to estimate through training the generative model.

In our theoretical formulation, we consider correction functions that correct the probability distribution p_θ, rather than the more intuitive (and practical) case of a correction function that corrects individual points that the distribution is defined over. In Appendix C, we specify sufficient conditions under which a pointwise correction function is guaranteed to correspond to a distribution-wise correction function of the same form as those which we consider in our theoretical study and therefore can enjoy the theoretical stability guarantees we prove. We also provide a concrete example of a projection function, in the Gaussian case, which provably satisfies those conditions. We conduct a series of experiments on this toy example in Section 5.

4.1.2. Understanding the weight update $\pi_\gamma \mathcal{G}_\lambda^n(\theta)$

The weight update $\pi_\gamma G_\lambda^n(\theta)$ in (5) is a formalization of the intended output of fine-tuning p_θ on $\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}$, where $\mathcal{D}_{\text{real}} = \{x_i\}_{i=1}^n$ is the ground truth dataset of size n, and $\mathcal{D}_{\text{synth}} = \{\tilde{x}_i : \tilde{x}_i \sim \widehat{\pi_\gamma p_\theta}\}_{i=1}^{\lfloor \lambda \cdot n \rfloor}$ is the synthesized-and-corrected dataset of size $\lfloor \lambda \cdot n \rfloor$. In other words, in an ideal run of stochastic gradient descent fine-tuning, the model weights θ should update to $\pi_\gamma \mathcal{G}_\lambda^n(\theta)$, as defined in (5), when trained on $\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth}}$.

Intuitively, the weight update $\theta \mapsto \pi_\gamma \mathcal{G}_\lambda^n(\theta)$ avoids the loss of variance in the generated data by ensuring that at each step, the model is trained on synthetic data which is likelier to have been sampled from the diverse target distribution. This positive phenomenon is more pronounced when the correction strength γ is larger.

4.2. Assumptions

In order to prove our main result, we need some regularity assumptions about the learning procedure. Informally speaking, we will assume that the class of generative models that we consider is smoothly parameterized by its model weights; the loss landscape is concave near the ideal model weights; and the class of generative models does an increasingly good job approximating the target data distribution as the dataset size increases. We formally quantify and state these hypotheses in Assumption 4.2.

Assumption 4.2. The following are true.

1. There exists some $L > 0$ such that, for all θ sufficiently close to $\theta^\star$, the mapping $x \mapsto \nabla_\theta^2 \log p_\theta(x)$ is L-Lipschitz.
2. The mapping $\theta \mapsto \mathbb{E}_{x \sim p_{\text{data}}}[\log p_\theta(x)]$ is continuously twice differentiable locally around $\theta^\star$, and there exists some $\alpha > 0$ such that $\mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta^2 \log p_\theta(x)\right]|_{\theta^\star} \preceq -\alpha I_d \prec 0$.
3. There exist $a, b, \varepsilon_{\text{OPT}} \geq 0$ and a neighborhood U of $\theta^\star$ such that, for any $\delta \in (0,1)$, with probability $1-\delta$ over the samplings, we have[2]
$$\|\pi_\gamma \mathcal{G}_\lambda^n(\theta) - \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)\| \leq \varepsilon_{\text{OPT}} + \frac{a}{\sqrt{n}} \sqrt{\log \frac{b}{\delta}}. \quad (6)$$
for all $\theta \in U$ and $n \in \mathbb{N}$. Denote this bound by $\tau_n(\delta)$.

In Assumption 4.2 (2), the notation "$\preceq$" corresponds to the Loewner order on symmetric matrices: we write that $A \preceq B$ if $B - A$ is positive semi-definite, and $A \prec B$ if $B - A$ is positive definite. In particular, Assumption 4.2 (2) implies that the matrix $\mathbb{E}_{x \sim p_{\text{data}}}\left[\nabla_\theta^2 \log p_\theta(x)\right]|_{\theta^\star}$ is negative definite, and its largest eigenvalue is at most $-\alpha$. And Assumption 4.2 (3) mirrors the main assumption in (Bertrand et al., 2024); it is motivated by generalization bounds in deep learning, see e.g. (Jakubovitz et al., 2019; Ji et al., 2021). The interested reader can consult Appendix B for more details on this assumption.

4.3. Iterative Fine-Tuning with Correction

We now have the language to state our main result, which essentially says that if the initial parameters θ_0 are suffi-

[2]The map $\pi_\gamma G_\lambda^\infty$ is defined similarly to $\pi_\gamma G_\lambda^n$ in (5), but with $\hat{p}_{\text{data}}$ replaced with p_{data}, and with $\widehat{\pi_\gamma p_\theta}$ replaced with $\pi_\gamma p_\theta$. See Appendix A for more details. This estimate is identical to the analogous Assumption 3 used in (Bertrand et al., 2024), with the only difference being it is applied to our iterative fine-tuning update function. See Appendix B for further discussion.

ciently close to the optimal model parameters $\theta^\star$, and if the augmentation percentage λ is sufficiently small, then under iterative fine-tuning with correction, we can expect our subsequent model parameters to stay close to $\theta^\star$.

Theorem 4.3 (Stability of Iterative Fine-Tuning with Correction)**.** *Fix an augmentation percentage $\lambda \in \mathbb{R}_{>0}$ and a correction strength $\gamma \in \mathbb{R}_{\geq 0}$. Suppose we have an iterative fine-tuning procedure defined by the rule $\theta^n_{t+1} = \pi_\gamma \mathcal{G}^n_\lambda(\theta^n_t)$, and suppose that Assumption 4.2 holds. Define the constant*

$$\rho(\lambda) := \rho(\lambda; \alpha, \varepsilon, L) := \frac{\lambda(\alpha + \varepsilon L)}{\alpha - \lambda(\alpha + \varepsilon L)}$$

and fix any $\delta \in (0, 1)$. If θ_0 is sufficiently close to $\theta^\star$, and if $\lambda\left(1 + \frac{\varepsilon L}{\alpha}\right) < \frac{1+\gamma}{2+\gamma}$, then $\rho(\lambda)/(1+\gamma) < 1$, and it follows that the stability estimate holds with probability $1 - \delta$:

$$\|\theta^n_t - \theta^\star\| \leq \tau_n(\delta/t) \sum_{i=0}^{t} \left(\frac{\rho(\lambda)}{1+\gamma}\right)^i + \left(\frac{\rho(\lambda)}{1+\gamma}\right)^t \|\theta^n_0 - \theta^\star\| \tag{7}$$

for all $t > 0$.

We prove Theorem 4.3 in Appendix A.

Remark 4.4. If we apply Theorem 4.3 with correction strength $\gamma = 0$, then the iterative fine-tuning procedure trains successively on a combination of raw synthetic data that has not been corrected using a correction function and ground truth data. This is exactly the case considered in (Bertrand et al., 2024). Accordingly, the bound in (7), applied with $\gamma = 0$, exactly recovers their result.

Corollary 4.5. *Under the assumptions from Theorem 4.3, iterative fine-tuning with any amount of correction outperforms iterative fine-tuning without correction–in the sense that it is exponentially more stable, and it results in better model weights.*

Proof of Corollary 4.5. We apply Theorem 4.3 with $\gamma = 0$, which corresponds to no correction, as well as with $\gamma > 0$, which corresponds to any amount of correction. For any $\gamma > 0$, we notice that the RHS of (7) is strictly smaller than when $\gamma = 0$. This guarantees better stability as $t \to \infty$, as well as model weights θ^n_t closer to $\theta^\star$. □

Example 4.6. If we apply Theorem 4.3 with correction strength $\gamma \to \infty$, then the bound (7) in Theorem 4.3 limits to $\tau_n(\delta/t)$. This implies that the practical iterate θ^n_t approaches the ideal model paramaters, and is at worst some constant away, that depends on error from the optimization procedure, as well as statistical error from using finitely many ground truth data samples n.

Note that Theorem 4.3 relies on the assumption that the initial model parameters θ_0 are sufficiently close to the ideal model parameters $\theta^\star$, and also that the augmentation percentage λ is sufficiently small. We hypothesize that these assumptions can be relaxed in the case where a correction function participates in the iterative fine-tuning procedure–intuitively, the correction function should compensate for errors that arise from θ^n_0 being worse, as well as errors that arise from incorporating more synthetic data. We frame this in the following conjecture.

Conjecture 4.7. *In the case of iterative fine-tuning with correction, we may relax how close the initial model parameters θ^n_0 need to be to the optimal model parameters $\theta^\star$, as well as choose a larger synthetic augmentation percentage λ, while still retaining the improved stability estimate (7).*

We provide empirical evidence for Conjecture 4.7 in Section 7 on the human motion synthesis task. In fact, Theorem 4.3 represents partial progress towards this conjecture. Namely, according to Theorem 4.3, for large correction strength γ, we can effectively choose a synthetic augmentation percentage that is twice as large as we would be able to without any correction, and still be able to meet the assumptions of the theorem. This is because $\lim_{\gamma\to\infty} \frac{1+\gamma}{2+\gamma} = 1$, which is twice as large as the bound when $\gamma = 0$.

5. Toy Example: Gaussian

We first assume oracle knowledge of the ground truth distribution, and use a toy example to directly demonstrate the impact of the correction strength γ on model performance and stability as stated in Theorem 4.3 and Corollary 4.5. Our ground truth distribution is a 2-dimensional isotropic Gaussian centered at the origin, i.e., $\theta^\star = ((0, 0), I_2)$, and our correction is "distribution-wise" in this idealized scenario. We consider the more practical setting, where we don't have oracle knowledge of the target distribution a priori, and where the data correction is "point-wise", in the empirical studies in the following two sections. Further, in Appendix C, we show that, in theory, sufficiently well-behaved pointwise correction functions indeed correspond to distribution-wise correction functions.

Concretely, our ground truth dataset contains 50 points sampled from the target distribution, which are used to estimate $\theta^{50}_0 = (\mu_0, \Sigma_0) \in \mathbb{R}^6$. We fix our synthetic augmentation percentage to be $\lambda = 0.5$, and inductively synthesize a new dataset $\mathcal{D}_{\text{synth}} = \{y_i \sim \mathcal{N}(\mu_t, \Sigma_t)\}_{i=1}^{25}$. We implement a correction function to map $\mathcal{D}_{\text{synth}}$, which was sampled from $p_{\theta^{50}_t}$, to a dataset $\mathcal{D}_{\text{corrected}}$, which is likelier to have been sampled from the target density $p_{\theta^\star}$. We do this by sampling $\mathcal{D}_{\text{corrected}}$ from the *middle density* corresponding to a given correction strength γ:

$$\pi_\gamma \hat{p}_{\theta^{50}_t}(x) := \frac{\hat{p}_{\theta^{50}_t}(x) + \gamma p_{\theta^\star}(x)}{1+\gamma}, \tag{8}$$

where $\hat{p}_{\theta^{50}_t}$ is the empirical PDF obtained from $\mathcal{D}_{\text{synth}}$.

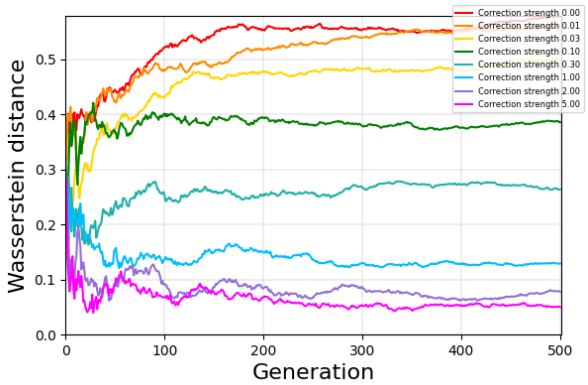


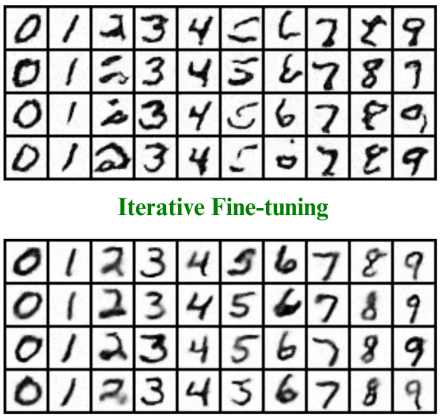
Figure 2. Empirical results from our Gaussian toy example. The graph demonstrates that increasing the correction strength γ, with a fixed augmentation ratio of $\lambda = 0.5$, improves performance and stability after self-consuming iterations.

We logarithmically accrue synthetic data points to simulate the case of fine-tuning. We obtain the updated model parameters θ_{t+1}^{50} by computing the sample mean and covariance on this augmented dataset. In Figure 2, we present the Wasserstein distance between the origin-centered isotropic Gaussian target distribution and the distribution defined by the parameters θ_t^{50} at each iteration t. Our results illustrate how increasing the correction strength γ adds stability and results in convergence near better Wasserstein scores in later generations, in accordance with Theorem 4.3. The experiments also demonstrate how even a very small increase in γ can improve performance over the baseline, in accordance with our claim of exponential improvement in Corollary 4.5.

6. Toy Example: MNIST

Our proof uses the optimal target PDF $p_{\theta^\star}$ to define the correction function π_γ. This is empirically validated by the Gaussian toy experiment, which assumes knowing the true target distribution. In practice, the correction function only depends on the ability to map synthesized data to data which is *likelier* to have been sampled from the ground truth distribution. Crucially, *this can be achieved without having a complete description of the target distribution.* For example, with our human motion experiments, we will demonstrate that point-wise correction based on the laws of physics is one proxy approach to make a sample more likely, without knowing the true target distribution.

One has the freedom to explore alternative approaches to data correction for more general data types, such as images. For example, one simple heuristic is to identify the "anchor" or "exemplar" images, which are intuitively representative and likely. The correction function can then be implemented as mapping or morphing synthesized data towards its nearest anchor, to make the synthesized data more representative and likely. In this section, we implement this approach on MNIST and study its performance.



Iterative Fine-tuning

Iterative Fine-tuning with Self-Correction

Figure 3. Empirical results from our MNIST toy example. These synthesized images demonstrate that after 50 self-consuming iterations at 150% augmentation percentage, the model which is trained using **iterative fine-tuning with self-correction** is able to generate higher quality samples than the model trained using **iterative fine-tuning** without any self-correction.

For our MNIST (LeCun et al., 1998) experiments, we train a diffusion model (Ho et al., 2020) for class-conditional image generation, using a train split of size $n = 12000$. For our iterative fine-tuning experiments, we train the model for 20 epochs, then synthesize $\lambda \cdot 12000/10$ images for each digit, and then augment the ground truth dataset with these to train on for the next generation; every following generation follows the same procedure, but only trains for a single epoch. We vary our experiments over augmentation percentages $\lambda \in \{0.2, 0.5, 1.0, 1.5\}$. To define our self-correction operation, we first compute K-means clusters over the training split for each digit. Our iterative fine-tuning with self-correction experiments use the same setup described above, except instead of training on the synthesized images, we train on the synthesized and then corrected images, where "correcting" an image means finding the nearest centroid in the K centroids for that digit that we computed at the start of training. We swept the values $K \in \{1, 2, 4, \ldots, 1024\}$, and we found that any reasonably large K results in the same general trend where self-correction improves the metrics and stability. We report our results for $K = 16$, which performs the best.

We present images synthesized using our trained models in Figure 3. These synthesized images demonstrate that iterative fine-tuning eventually generates many low quality and illegible digits, and this problem is solved by applying our self-correction operation. Further experiment details, including graphs of the FID metrics for each generation that provide rigorous evidence for this trend across augmentation percentages, can be found in Appendix D. Our empirical results demonstrate that applying self-correction improves performance during iterative fine-tuning for our MNIST image generation task across self-consuming generations, and

this relative performance is amplified when the augmentation percentage is larger. The behavior that we observe is consistent with our theoretical results in Section 4, as well as our human motion experiments in Section 7.

7. Human Motion Synthesis

Theorem 4.3 states that, in theory, iterative fine-tuning with correction should be more stable than iterative fine-tuning without correction. Crucially, the stability estimates that we prove rely on the dataset size, the synthetic augmentation percentage, how expressible the generative model class is, and having an idealized correction function. To validate how our theory works beyond toy examples, we conduct a case study on human motion synthesis with diffusion models (Tevet et al., 2023). We believe this is a natural setting to test our iterative fine-tuning with correction framework, because synthesizing natural motions is a challenging problem, but there is a natural and intuitive way to automatically correct them at scale–namely, using a physics simulator.

7.1. Generative Model

For our generative model, we use the Human Motion Diffusion Model (MDM) (Tevet et al., 2023). This is a classifier-free diffusion-based generative model for the text-to-motion generation task, where the model receives as input a description of a motion sequence (e.g. "get down on all fours and crawl across the floor"), and outputs a sequence of skeleton poses which attempt to embody that prompt. Synthesizing human motion is challenging not only for the diverse and compositional text prompts, but also due to failure of physics obeying-ness (e.g. feet skating, floating, penetrating a surface), which is not explicitly enforced by deep generative models.

7.2. Physics Simulator as Self-Correction Function

For our self-correction function, we use Universal Humanoid Control (UHC) (Luo et al., 2021), which is an imitation policy that operates inside the MuJoCo physics simulator (Todorov et al., 2012). Given an input sequence of humanoid skeleton poses, UHC attempts to imitate the motion sequence, constrained by the laws of physics imposed by the physics simulator, and it outputs a new motion sequence that is the closest possible approximation it can replace it with. For example, if an input motion sequence violates the laws of physics by having a foot penetrate through the floor, then the motion sequence output by UHC will attempt to remove that physically impossible artifact while maintaining the semantic integrity of the original input motion. We use VPoser (Pavlakos et al., 2019) and SMPL (Loper et al., 2015) to translate joint representations between the human motion generator and the physics simulator.

The physics simulator allows us to self-correct a synthesized motion automatically. Our underlying assumption is that by enforcing the physics obeying-ness (via the simulator) and closeness to the synthesized motion (via the imitation objective), the self-correction function would act as similar as an idealized corrector as possible.

7.3. Experimental setup

We preprocess the MoVi (Ghorbani et al., 2021) subset of HumanML3D (Guo et al., 2022) using the official code implementation of HumanML3D. We filter out movements involving interactions with chairs, as UHC by default does not handle human-object interactions. We take as our train split the train split from HumanML3D, intersected with our filtered subset of MoVi, and likewise for the test split. This procedure yields a train set of size $n = 2794$ and a test set of size 546. We further randomly select a smaller training set of $n \in \{64, 128, 256\}$ examples, to simulate the more challenging scenario when the initial generative model is sub-optimal (due to data scarcity). The smaller data also enables us to explore larger synthetic augmentation percentage due to compute constraints. From here, the iterative re-training procedure follows Algorithm 1. We spell it out in this concrete experimental setup.

We first train on the ground truth train split until the model is nearly converged, using all the default hyperparameters from MDM. We evaluate and save this last checkpoint from generation 0. From here, for each generation $t \in \{1, 2, \ldots, 50\}$, we run three sets of experiments.

A. *Baseline*: fine-tune the latest checkpoint from generation $t-1$ for m batches on ground truth dataset $\mathcal{D}_{\text{real}}$.

B. *Iterative fine-tuning:* fine-tune the latest checkpoint from generation $t-1$ on $\mathcal{D}_{\text{real}} \cup \mathcal{D}_{\text{synth},t-1}$ for m batches. Here, $\mathcal{D}_{\text{synth},t-1}$ is a synthetic dataset of size $\lfloor \lambda \cdot n \rfloor$ generated from the checkpoint for generation $t-1$, using randomly chosen prompts from the train split.

C. *Iterative fine-tuning with self-correction:* fine-tune the latest checkpoint from generation $t-1$ on $D_{\text{real}} \cup \text{UHC}(\mathcal{D}_{\text{synth},t-1})$ for m batches. Here, $\text{UHC}(\mathcal{D}_{\text{synth},t-1})$ denotes a synthetic dataset of size $\lfloor \lambda \cdot n \rfloor$ generated from the latest checkpoint for generation $t-1$, using randomly chosen prompts from the train split, which is then corrected by UHC.

We experiment with synthetic augmentation percentages $\lambda \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$ on the larger dataset; we set the number of batches seen during generation 0 to be 3125, and the number of batches seen for each later generation to be $m = 625$. Separately, we experiment with synthetic augmentation percentages $\lambda \in \{0.25, 0.50, 0.75, 1.00\}$ on the smaller datasets; we set the number of batches seen during generation 0 to be $78 * k$ for

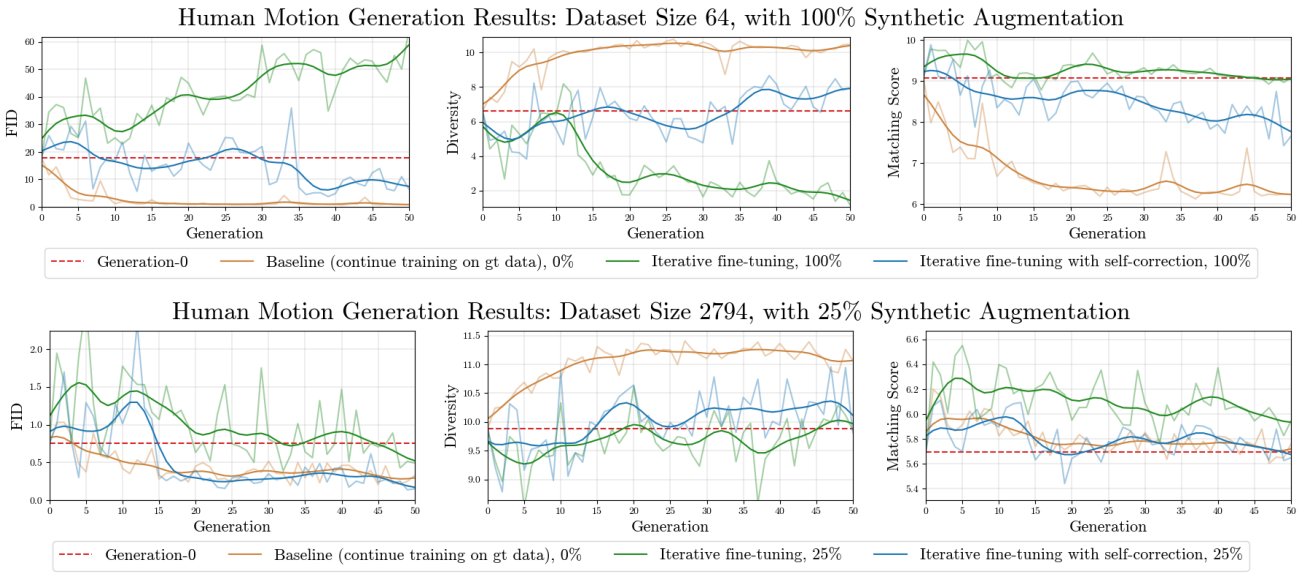


Figure 4. Results from our human motion experiments on iterative fine-tuning with self-correction. These graphs show evaluation metrics for the last checkpoint for every generation. This is the checkpoint used for sampling in the iterative fine-tuning experiments, and it is also the checkpoint where training is resumed with this new partially synthesized dataset. We can see that **with self-correction**, the iterative fine-tuning procedure more stably converges to a better FID score, and more quickly. When the dataset size is smaller ($n = 64$, above) we can see that iterative fine-tuning **with no self-correction** has a flat Matching score, as well as diverging FID and Diversity scores, indicating model collapse. And when the dataset size is larger ($n = 2794$, below), there is less collapse for iterative fine-tuning with no self-correction, although the variance of the FID score is worse, as is the average FID across generations. In both cases, we see that iterative fine-tuning **with self-correction** outperforms iterative fine-tuning **with no self-correction**, and is competitive with the **baseline** after many generations.

dataset size $64 * k$, and the number of batches seen for each later generation $t > 0$ to be $m = 16$. We choose to control how many data points the model sees across each generation, rather than controlling some other quantity like the number of epochs, as this allows each experiment to compare against its baseline in a controlled way, which in turn allows them to compare against each other in a controlled way.

We compute every evaluation one time for each checkpoint using the evaluation script provided in the original MDM codebase. Regardless of the train split size, we perform sampling for evaluation using all 546 motion sequences from the test split, since the FID score is sensitive to generated dataset size. We use the same hyperparameters as those used for MDM, including batch size 64, AdamW (Loshchilov & Hutter, 2019) with learning rate $1e-4$, and classifier-free guidance parameter 2.5. And for UHC we used the `uhc_explicit` model for imitation.

7.4. Quantitative Analysis of Results

For each of these experiments we report the metrics from MDM, as used by (Guo et al., 2022): FID measures how similar the distribution of generated motions is to the ground truth distribution; Diversity measures the variance of the generated motions; and Matching Score measure how well the generated motions embody the given text prompt. In Figure 4 we present results from experiments on our 64-size dataset with 100% synthetic augmentation, as well as our 2794-size dataset with 25% synthetic augmentation.

Our experimental results confirm our theoretical results, that iterative fine-tuning with self-correction outperforms iterative fine-tuning without self-correction, in the sense that the graphs are generally more stable across generations, and approach better evaluation metric values. In particular, Theorem 4.3 and Corollary 4.5 claim that any amount of idealized self-correction will improve the stability bound during iterative fine-tuning. Our results in Figure 4 demonstrate that the FID score is lower and more stable across generations when applying self-correction, and generally higher and less stable than the baseline, where there is no self-consuming training at all. We conduct experiments across multiple seeds, and we find empirically that this general phenomenon holds consistently, where the self-correction technique consistently yields improved training dynamics over iterative fine-tuning with no correction. Graphs from these runs can be found in Appendix G.

Our experimental results also provide empirical evidence for Conjecture 4.7. Observe that in the baseline experiments in Figure 4, the FID score decreases across generations, which indicates that the initial model parameters θ_0^n are not that close to the optimal model parameters $\theta^\star$; additionally, the augmentation percentages considered in the graph

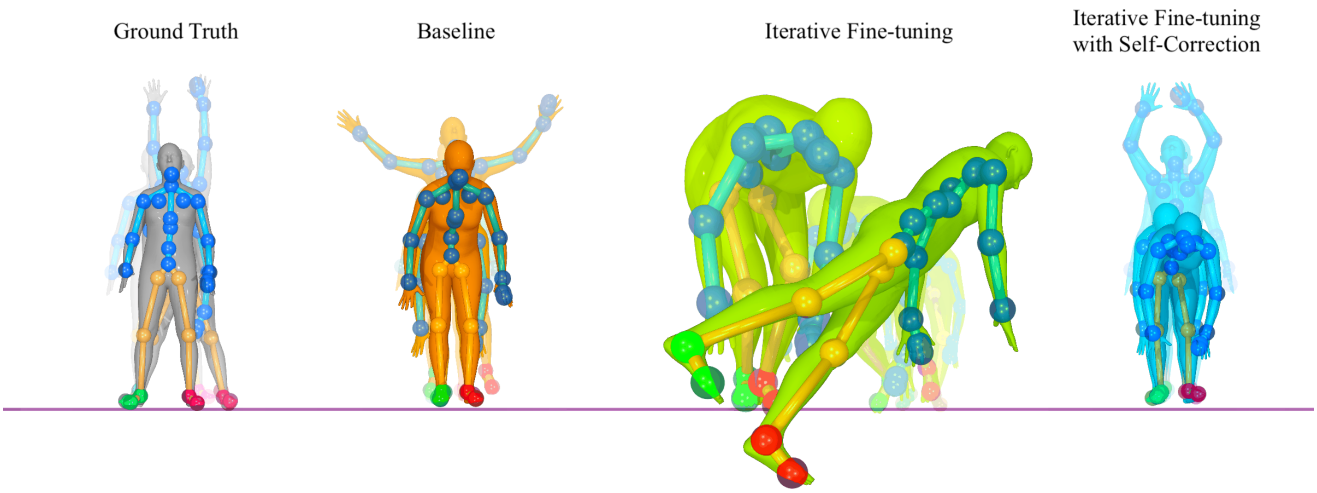


Figure 5. How does the self-correction operation affect iterative fine-tuning, qualitatively? Here we present some visualizations. The prompt which describes the ground truth motion, and which we use to generate the three other motions, is: *"a person stands with feet wide, stretches both hands up over his head and then swings down by the waist and hangs arms down before standing up"*. We can see that the **iterative fine-tuning model** produces a motion where the human moves closer to the camera than the others; this is evidence of model collapse, as moving feet is irrelevant to the prompt. Additionally, this motion produces single frames that suddenly snap to a physically impossible position–note the leg penetration through the ground plane. These negative artifacts do not exist in the motions synthesized from the **ground truth**, **baseline model**, or **iterative fine-tuning with self-correction model**. Lastly, we note that the iterative fine-tuning motion depicted here is semantically similar to crawling. We observe in our experiments with smaller dataset sizes that the iterative fine-tuning model generates less diverse outputs than the baseline model and the iterative fine-tuning with self-correction model, and that this crawling pattern appears more often in the latter. Each snapshot is taken at exactly frame 105 of their respective videos. The two motions on the right come from models that were iteratively fine-tuned for 50 generations, with a train set of size $n = 64$, and a synthetic augmentation percentage of 25%. For all pictures of the human, the camera is fixed at the same position, and for consistency the image is not resized.

are 25% and 100%. Conjecture 4.7 claims that performing self-correction during iterative fine-tuning improves performance, even when the initial model weights are sub-optimal and simultaneously the synthetic augmentation percentage is large. This claim is confirmed by Figure 4. We direct the curious reader to Appendix F, where we present graphs for all of the above listed training set sizes and augmentation percentages, providing additional empirical evidence for Theorem 4.3, Corollary 4.5, and Conjecture 4.7.

7.5. Qualitative Analysis of Results

We visually inspect the generated human motion sequences in order to analyze what concrete effect the self-correction has on iterative fine-tuning. We find that the correctness and diversity of synthesized motions are improved by the self-correction procedure, in agreement with our quantitative analysis in Subsection 7.4. We present snapshots of our synthesized motions in Figure 5, and we analyze the motions in the caption. In short, we find that physics-disobeying artifacts such as floor penetration or floating become more pronounced without the self-correction. We also find that in the model without self-correction, the humanoid sometimes performs movements completely unrelated to the prompt; our model with self-correction fixes these negative phenomena. We direct the curious reader to Appendix E, where we present more examples from our qualitative analysis, as well as our project webpage, where we provide side-by-side video comparisons.

8. Conclusion

Our paper investigates the learning of generative models when the training data includes machine-generated contents. We investigate how self-correction functions, which automatically correct synthesized data points to be more likely under the true data distribution, can stabilize self-consuming generative model training. Our theoretical results show that self-correction leads to exponentially more stable model training and smaller variance, which we illustrate with a Gaussian toy example. We then demonstrate how physics simulators can serve as a self-correction function for the challenging human motion synthesis task, where models trained with our self-correcting self-consuming loops generate higher quality motions, and manage to avoid collapse even at a high synthetic data to real data ratio. Future work includes exploring self-correcting functions for more diverse applications, such as language modeling and text-to-image generation, and investigating when self-consuming training may lead to overall better generative models.

Acknowledgments

We would like to thank Stephen H. Bach, Quentin Bertrand, Carsten Eickhoff, Gauthier Gidel, Jeff Hoffstein, Zhengyi Luo, Singh Saluja, and Ye Yuan for useful discussions. We would also like to thank the anonymous reviewers. This work is supported by the Samsung Advanced Institute of Technology, Honda Research Institute, and a Richard B. Salomon Award for Chen Sun. Our research was conducted using computational resources at the Center for Computation and Visualization at Brown University.

Impact Statement

This paper presents work whose goal is to provide theoretical analysis and practical tools to address the data contamination issue caused by machine-generated content. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Alemohammad, S., Casco-Rodriguez, J., Luzi, L., Humayun, A. I., Babaei, H., LeJeune, D., Siahkoohi, A., and Baraniuk, R. Self-consuming generative models go MAD. In *The Twelfth International Conference on Learning Representations*, 2024.

Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Bertrand, Q., Bose, A. J., Duplessis, A., Jiralerspong, M., and Gidel, G. On the stability of iterative retraining of generative models on their own data. In *The Twelfth International Conference on Learning Representations*, 2024.

Ghorbani, S., Mahdaviani, K., Thaler, A., Kording, K., Cook, D. J., Blohm, G., and Troje, N. F. Movi: A large multi-purpose human motion and video dataset. *Plos one*, 16(6):e0253157, 2021.

Guo, C., Zou, S., Zuo, X., Wang, S., Ji, W., Li, X., and Cheng, L. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5152–5161, 6 2022.

He, R., Sun, S., Yu, X., Xue, C., Zhang, W., Torr, P., Bai, S., and Qi, X. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jahanian, A., Puig, X., Tian, Y., and Isola, P. Generative models as a data source for multiview representation learning. In *International Conference on Learning Representations*, 2022.

Jakubovitz, D., Giryes, R., and Rodrigues, M. R. Generalization error in deep learning. In *Compressed Sensing and Its Applications: Third International MATHEON Conference 2017*, pp. 153–193. Springer, 2019.

Ji, K., Zhou, Y., and Liang, Y. Understanding estimation and generalization error of generative adversarial networks. *IEEE Transactions on Information Theory*, 67(5):3114–3129, 2021.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6): 248:1–248:16, October 2015.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Luo, Z., Hachiuma, R., Yuan, Y., and Kitani, K. Dynamics-regulated kinematic policy for egocentric pose estimation. In *Advances in Neural Information Processing Systems*, 2021.

Martínez, G., Watson, L., Reviriego, P., Hernández, J. A., Juarez, M., and Sarkar, R. Towards understanding the interplay of generative artificial intelligence and the internet. In Cuzzolin, F. and Sultana, M. (eds.), *Epistemic Uncertainty in Artificial Intelligence*, pp. 59–73, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-57963-9.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Perdomo, J., Zrnic, T., Mendler-Dünner, C., and Hardt, M. Performative prediction. In *International Conference on Machine Learning*, pp. 7599–7609. PMLR, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Saunders, W., Yeh, C., Wu, J., Bills, S., Ouyang, L., Ward, J., and Leike, J. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Shumailov, I., Shumaylov, Z., Zhao, Y., Gal, Y., Papernot, N., and Anderson, R. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arxiv:2305.17493*, 2023.

Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-or, D., and Bermano, A. H. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023.

Tian, Y., Fan, L., Chen, K., Katabi, D., Krishnan, D., and Isola, P. Learning vision from models rivals learning vision from data. *arXiv preprint arXiv:2312.17742*, 2023.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Welleck, S., Lu, X., West, P., Brahman, F., Shen, T., Khashabi, D., and Choi, Y. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*, 2023.

Wu, T.-H., Lian, L., Gonzalez, J. E., Li, B., and Darrell, T. Self-correcting llm-controlled diffusion models. *arXiv preprint arXiv:2311.16090*, 2023.

Xu, S., Li, Z., Wang, Y.-X., and Gui, L.-Y. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14928–14940, 2023.

Yuan, Y., Song, J., Iqbal, U., Vahdat, A., and Kautz, J. Physdiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

A. Mathematical Theory: The Proof of Theorem 4.3

In this appendix, we provide a full account of the mathematical details of the theorems and their proofs appearing in the main body of the paper. Our proof technique has the same framework as (Bertrand et al., 2024) because our theoretical analysis generalizes theirs to the case where you have a self-correction function in the self-consuming loop.

A.1. Mathematical Setup and Notation

Definition A.1. Define the optimal model parameters to be

$$\theta^\star \in \arg\max_{\theta' \in \Theta} \mathbb{E}_{x \sim p_{\text{data}}}[\log p_{\theta'}(x)], \tag{9}$$

chosen so that $\|\theta^\star\|$ has minimal norm within this set. Let θ be any model parameters. Then the *correction of strength* γ of distribution p_θ towards $p_{\theta^\star}$ is a new distribution, denoted $\pi_\gamma p_\theta$, defined according to the rule

$$\pi_\gamma p_\theta(x) := \frac{p_\theta(x) + \gamma p_{\theta^\star}(x)}{1+\gamma}.$$

This is illustrated in Figure 6. Let θ_t be the parameters of the model trained after t generations. We define the *iterative fine-tuning with correction update mapping* to be

$$\pi_\gamma \mathcal{G}_\lambda^\infty(\theta) := \text{local}\operatorname*{argmax}_{\theta' \in \Theta} \mathcal{H}(\theta, \theta') := \text{local}\operatorname*{argmax}_{\theta' \in \Theta} [\mathbb{E}_{x \sim p_{\text{data}}}[\log p_{\theta'}(x)]] + \lambda \mathbb{E}_{x \sim \pi_\gamma p_\theta}[\log p_{\theta'}(x)]] \tag{10}$$

$$\pi_\gamma \mathcal{G}_\lambda^n(\theta) := \text{local}\operatorname*{argmax}_{\theta' \in \Theta} \hat{\mathcal{H}}(\theta, \theta') := \text{local}\operatorname*{argmax}_{\theta' \in \Theta} [\mathbb{E}_{x \sim \hat{p}_{\text{data}}}[\log p_{\theta'}(x)]] + \lambda \mathbb{E}_{x \sim \widehat{\pi_\gamma p_\theta}}[\log p_{\theta'}(x)]]. \tag{11}$$

Notice that in the finite case, we're optimizing by taking samples from an empirical distribution. In contrast, in the infinite case, there is zero statistical error, since the parameter update is done with access to an infinite sampling budget at each generation t. The finite case is the more practical case, when we have some statistical error (so we only have access to finite sampling at each generation). Since the parameter space of the generative model class might be limited, there might be a small difference between the distribution corresponding to the optimal parameters and the target distribution p_{data}; we capture this difference via the Wasserstein-2 distance and denote

$$\varepsilon := d_W(p_{\theta^\star}, p_{\text{data}}). \tag{12}$$

Let

$$\mathcal{H}_1(\theta') := \mathbb{E}_{x \sim p_{\text{data}}}[\log p_{\theta'}(x)], \qquad \mathcal{H}_2(\theta, \theta') := \mathbb{E}_{x \sim \pi_\gamma p_\theta}[\log p_{\theta'}(x)]. \tag{13}$$

and note that $\mathcal{H}(\theta, \theta') = \mathcal{H}_1(\theta') + \lambda \mathcal{H}_2(\theta, \theta')$.

We first establish that the correction map is truly a mapping of probability distributions as well as some of its elementary properties.

Lemma A.2. *The correction map has the following properties.*

1. *$\pi_\gamma p_\theta$ is a probability distribution.*
2. *Strengths $0, 1, \infty$ correspond to p_θ, the average of p_θ and $p_{\theta^\star}$, and $p_{\theta^\star}$, respectively.*
3. *For any $x \in \mathbb{R}^n$, if $\gamma > 1$, then*
$$\|\pi_\gamma p_\theta(x) - p_{\theta^\star}(x)\| \leq \|\pi_\gamma p_\theta(x) - p_\theta(x)\|,$$
and if $\gamma < 1$, then the inequality is flipped. In other words, $\pi_\gamma p_\theta$ is a better estimate of the ideal distribution $p_{\theta^\star}$ than p_θ is, precisely when the projection strength is more than 1.

Proof. For the first point, $\pi_\gamma p_\theta$ is a probability distribution because it is a convex combination of probability distributions. For example, we can compute that

$$\int_{\mathbb{R}^d} \pi_\gamma p_\theta dx = \frac{1}{1+\gamma} \int_{\mathbb{R}^d} p_\theta(x) dx + \frac{\gamma}{1+\gamma} \int_{\mathbb{R}^d} p_{\theta^\star}(x) dx = \frac{1}{1+\gamma} \cdot 1 + \frac{\gamma}{1+\gamma} \cdot 1 = 1.$$

The second point follows immediately from the definition of $\pi_\gamma p_\theta$. For the third point, we can estimate that

$$\begin{aligned}
\|\pi_\gamma p_\theta(x) - p_{\theta^\star}(x)\| &= \left\| \frac{p_\theta(x) + \gamma p_{\theta^\star}(x)}{1+\gamma} - \frac{p_{\theta^\star}(x)(1+\gamma)}{1+\gamma} \right\| \\
&= \frac{1}{1+\gamma} \cdot \|p_\theta(x) - p_{\theta^\star}(x)\| \\
&\leq \frac{\gamma}{1+\gamma} \cdot \|p_{\theta^\star}(x) - p_\theta(x)\| \\
&= \left\| \frac{p_\theta(x) + \gamma p_{\theta^\star}(x)}{1+\gamma} - \frac{p_\theta(x)(1+\gamma)}{1+\gamma} \right\| \\
&= \|\pi_\gamma p_\theta(x) - p_\theta(x)\|
\end{aligned}$$

when $\gamma > 1$. The inequality flips when $\gamma < 1$. □

Intuitively, it is clear that we cannot hope to prove general results about generative models without assuming something about the mapping $\theta \mapsto p_\theta$. We now state the two assumptions we require in order to make our theoretical arguments; note that they are precisely the same assumptions made in (Bertrand et al., 2024). The first assumption is a local Lipschitzness property that we will exploit via the Kantorovich-Rubenstein duality:

Assumption A.3. For θ close enough to $\theta^\star$, the mapping $x \mapsto \nabla_\theta \nabla_\theta \log p_\theta(x)$ is L-Lipschitz.

The second assumption is a local regularity and concavity condition:

Assumption A.4. The mapping $\theta \mapsto \mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)]$ is continuously twice differentiable locally around $\theta^\star$ and $\mathbb{E}_{x\sim p_{\text{data}}}\left[\nabla_\theta \nabla_\theta \log p_\theta(x)\right]_{\theta^\star} \preceq -\alpha I_d \prec 0$.

We next show the existence and uniqueness of $\pi_\gamma \mathcal{G}_\lambda^\infty(\infty)$ locally around $\theta^\star$.

Proposition A.5 (The Local Maximum Likelihood Solution is Unique)**.** *The following are true:*

A. *There exists an open neighborhood $U \subset \mathbb{R}^d$ containing $\theta^\star$ and a continuous function $g : U \to \mathbb{R}^d$ such that $g(\theta^\star) = \theta^\star$, and*

$$\nabla_{\theta'} \mathcal{H}(\theta, \theta')|_{\theta, g(\theta)} = 0 \tag{14}$$

for every $\theta \in U$.

B. *Given optimal model parameters $\theta^\star$ as in (9) that follow Assumptions A.3 and A.4, we have that, if $\varepsilon L < \alpha$, then for all $\lambda > 0$ and θ in a small enough neighborhood U around $\theta^\star$, there exists a unique local maximizer $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta)$ in U.*

Proof. **We first prove part A.** It suffices to apply the Implicit Function Theorem to the map

$$\mathbb{R}^{2d} \to \mathbb{R}^d : (\theta, \theta') \mapsto \nabla_{\theta'} \mathcal{H}(\theta, \theta')|_{\theta, \theta'} \tag{15}$$

in an open neighborhood of $(\theta^\star, \theta^\star)$. To do this, we need to show the following:

i) The map vanishes at $(\theta^\star, \theta^\star)$, i.e.

$$\nabla_{\theta'} \mathcal{H}(\theta, \theta')|_{\theta^\star, \theta^\star} = 0. \tag{16}$$

ii) The Jacobian matrix at $(\theta^\star, \theta^\star)$ is invertible, i.e.,

$$\nabla_{\theta'} \nabla_{\theta'} \mathcal{H}(\theta, \theta')|_{\theta^\star, \theta^\star} \qquad \text{is invertible.} \tag{17}$$

We first prove i). Recall from the definition (10) that $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta) = \arg\max_{\theta' \in \Theta} \mathcal{H}(\theta, \theta')$. This means that for any θ, $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta)$ is the choice of θ' which maximizes $\mathcal{H}(\theta, \theta')$. In particular, for $\theta = \theta^\star$, we have that $\theta' = \pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star)$ is the choice which maximizes $\mathcal{H}(\theta^\star, \theta')$. But $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star) = \theta^\star$ by Proposition A.6. This implies that its derivative is zero at $\theta' = \theta^\star$, meaning $\nabla_{\theta'} \mathcal{H}(\theta, \theta')|_{\theta^\star, \theta^\star} = 0$, as needed.

Now we prove ii). In order to show that the matrix (17) is invertible, it suffices to show it is close to another matrix which is invertible. A natural choice is the matrix

$$M = (1+\lambda)\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]|_{\theta^\star}. \tag{18}$$

First of all, note that this matrix indeed exists; by Assumption 2 A.4, we know the map $\theta' \mapsto \mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]$ is continuously twice differentiable locally near $\theta^\star$. We can estimate that the matrices (17) and (18) are indeed close as follows:

$$\begin{aligned}
&\|\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}(\theta,\theta')|_{\theta^\star,\theta^\star} - (1+\lambda)\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad = \|\nabla_{\theta'}\nabla_{\theta'}[\mathbb{E}_{x\sim p_{\text{data}}}\log p_{\theta'}(x) + \lambda\mathbb{E}_{x\sim\pi_\gamma p_\theta} p_{\theta'}(x)]|_{\theta^\star,\theta^\star} - (1+\lambda)\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad = \lambda\|[\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim\pi_\gamma p_\theta}\log p_{\theta'}(x)]|_{\theta^\star,\theta^\star} - \nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad = \lambda\|[\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\theta^\star}}\log p_{\theta'}(x)]_{\theta^\star} - [\nabla_{\theta'}\nabla_{\theta'}\mathbb{E}_{x\sim p_{\text{data}}}\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad = \lambda\|[\mathbb{E}_{x\sim p_{\theta^\star}}\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)]_{\theta^\star} - [\mathbb{E}_{x\sim p_{\text{data}}}\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad \le L\lambda\mathbb{E}_{(x,x')\sim p_{\theta^\star}\times p_{\text{data}}}\|[\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)]_{\theta^\star} - \nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)]_{\theta^\star}\| \\
&\quad \le \lambda\varepsilon L
\end{aligned}$$

where the first equality follows from the definition of $\mathcal{H}$ in (13); the second equality follows from some cancellation; the third equality follows the fact that the derivatives are constant with respect to θ, and $\pi_\gamma p_{\theta^\star} = p_{\theta^\star}$ by Lemma A.2; we exchange the derivative and the expectation in equation 4 using the Dominated Convergence Theorem, since Assumption 1 A.3 says that $x \mapsto \nabla_\theta\nabla_\theta \log p_\theta(x)$ is L-Lipschitz; the fifth estimate follows from Kantorovich-Rubinstein Duality; and the final estimate is the definition of Wasserstein distance (12).

Finally, we verify M is indeed invertible. Assumption 2 A.4 implies that the largest eigenvalue of M is at most $-(1+\lambda)\alpha$. Therefore, since all eigenvalues of M are nonzero, M is invertible. We can now apply the implicit function theorem to (15), and part A follows immediately.

Next, we prove part B. Let $d_U = \sup_{\theta\in U} d_W(p_{\theta^\star}, p_\theta)$. To verify that $g(\theta)$ is a local maximizer of (15), it suffices to show that $\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}(\theta, g(\theta)) \prec 0$. By Assumption 2 A.4, we know $\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(\theta^\star) \prec -\alpha I_d$ and since $\theta' \mapsto \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(\theta')$ is *continuously* twice differentiable locally near $\theta^\star$, we also have $\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(g(\theta)) \prec -\alpha I_d$. Thus, we have

$$\begin{aligned}
\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}(\theta, g(\theta)) &= \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(g(\theta')) + \lambda\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta, g(\theta)) \\
&= (1+\lambda)\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(g(\theta)) + \lambda(\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta, g(\theta)) - \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(g(\theta))) \\
&\preceq -\alpha(1+\lambda)I_d + \lambda L\left(\frac{1}{1+\gamma}d_W(p_\theta, p_{\theta^\star}) + \varepsilon\right)I_d,
\end{aligned}$$

where the last step follows from Kantorovich-Rubsenstein duality:

$$\begin{aligned}
&\|\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta,\theta') - \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(\theta')\| \\
&\quad \le \|\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta,\theta') - \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta^\star,\theta')\| + \|\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_2(\theta^\star,\theta') - \nabla_{\theta'}\nabla_{\theta'}\mathcal{H}_1(\theta')\| \\
&\quad = \|\int_{\mathbb{R}^d}\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)\frac{p_\theta(x)+\gamma p_{\theta^\star}(x)}{1+\gamma}\,dx - \int_{\mathbb{R}^d}\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)p_{\theta^\star}(x)\,dx\| \\
&\qquad + \|\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)] - \mathbb{E}_{x\sim p_{\theta^\star}}[\log p_{\theta'}(x)]\| \\
&\quad \le \frac{1}{1+\gamma}\|\int_{\mathbb{R}^d}\nabla_{\theta'}\nabla_{\theta'}\log p_{\theta'}(x)\left(p_\theta(x) - p_{\theta^\star}(x)\right)\,dx\| + L\varepsilon \\
&\quad = \frac{1}{1+\gamma}\|\mathbb{E}_{x\sim p_\theta}[\log p_{\theta'}(x)] - \mathbb{E}_{x\sim p_{\theta^\star}}[\log p_{\theta'}(x)]\| + L\varepsilon \\
&\quad \le \frac{L}{1+\gamma}d_W(p_\theta, p_{\theta^\star}) + L\varepsilon \\
&\quad \le \frac{L}{1+\gamma}d_U + L\varepsilon
\end{aligned}$$

Thus, to have $\nabla_{\theta'}\nabla_{\theta'}\mathcal{H}(\theta, g(\theta)) \prec 0$, it is sufficient that

$$-\alpha(1+\lambda) + \lambda L\left(\frac{1}{1+\gamma}d_U + \varepsilon\right) < 0,$$

which is guaranteed for all $\lambda > 0$ by $\alpha > L\varepsilon$ and $d_U \leq \frac{\alpha(1+\gamma)}{\lambda}$. This concludes the proof. □

Further, as we would expect, $\theta^\star$ is a fixed point of $\pi_\gamma \mathcal{G}_\lambda^\infty$:

Proposition A.6 (The optimal parametric generative model is a fixed point)**.** *For any given data distribution p_{data}, any $\theta^\star$ as defined by (9), and for all $\lambda > 0$, we have $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star) = \theta^\star$.*

Proof. Unpacking definition (10) shows that $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star) = \mathcal{G}_\lambda^\infty(\theta^\star)$, and we know by Proposition 4 from (Bertrand et al., 2024) that $\mathcal{G}_\lambda^\infty(\theta^\star) = \theta^\star$. □

A.2. Convergence of Iterative Fine-tuning with Correction for Infinite Sampling

We now have the required setup to state and prove a convergence result for iterative fine-tuning assuming infinite access to underlying probablity distributions. We need the following result, which is a technical lemma that provides a computation of the Jacobian of $\pi_\gamma G_\lambda^\infty$ at $\theta^\star$ as well as a spectral bound, both essential for the proof of Theorem A.8.

Lemma A.7. *We define the matrices*

$$A := (\nabla^2_{\theta',\theta'} \mathcal{H}_1(\theta'))|_{\theta^\star} \tag{19}$$

$$B := \nabla^2_{\theta,\theta'} \mathbb{E}_{x\sim p_\theta}[\log p_{\theta'}(x)]\big|_{\theta^\star,\theta^\star} \tag{20}$$

$$C := \nabla^2_{\theta',\theta'} \mathbb{E}_{x\sim p_\theta}[\log p_{\theta'}(x)]\big|_{\theta^\star,\theta^\star} \tag{21}$$

Recall the definition of $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta)$ from (10). Since γ and λ are fixed, denote $\pi\mathcal{G}(\theta) = \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)$. Finally, let $\mathcal{J}(\pi\mathcal{G}(\theta)) := \nabla_\theta \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)|_\theta$ denote the Jacobian of $\pi_\gamma \mathcal{G}_\lambda^\infty(\theta)$.

I. There exists an open neighborhood $U \subseteq \Theta$ containing $\theta^\star$ such that for all $\theta \in U$, we have

$$\mathcal{J}(\pi\mathcal{G}(\theta)) = -\left(\nabla^2_{\theta',\theta'} \mathcal{H}(\theta, \pi\mathcal{G}(\theta))\right)^{-1} \cdot \lambda \nabla^2_{\theta,\theta'} \mathcal{H}_2(\theta, \pi\mathcal{G}(\theta)). \tag{22}$$

II. We have that $\nabla^2_{\theta,\theta'} \mathcal{H}_2(\theta^\star, \theta^\star) = \frac{B}{1+\gamma}$, and $B = -C$, so the Jacobian of $\pi\mathcal{G}$ at $\theta^\star$ is

$$\mathcal{J}(\pi\mathcal{G}(\theta^\star)) = (I + \lambda A^{-1}C)^{-1} \cdot \frac{\lambda}{1+\gamma} A^{-1}C \tag{23}$$

III. The spectral norm of $A^{-1}C$ can be bounded as

$$\|A^{-1}C\| \leq 1 + \frac{L\varepsilon}{\alpha}. \tag{24}$$

Proof. **We first prove I.** We apply Proposition A.5. Part A of that proposition gives us a function $g : U \to \mathbb{R}^d$ such that $\nabla_{\theta'} \mathcal{H}(\theta, \theta')_{\theta, g(\theta)} = 0$. But part B of that proposition says that there exists a unique local maximizer inside U, and this local maximizer is $\pi_\gamma \mathcal{G}_\lambda^\infty$. This implies that $\nabla_{\theta'} \mathcal{H}(\theta, \theta')_{\theta, \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)} = 0$. Next, we implicitly differentiate this equation with respect to θ. Recall that when you have an equation of the form $f(x, y) = 0$, and implicitly differentiate it in the form $f(x, g(x)) = 0$ with respect to x, you obtain $\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial x} = 0$, and solving for $\frac{\partial g}{\partial x}$ yields $\frac{\partial g}{\partial x} = -\left(\frac{\partial f}{\partial y}\right)^{-1} \frac{\partial f}{\partial x}$. We apply this formula with

$$(x, f, g) = (\theta, \theta \mapsto \nabla_{\theta'} \mathcal{H}(\theta, \theta')_{\theta, \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)}, \theta \mapsto \pi_\gamma \mathcal{G}_\lambda^\infty(\theta))$$

and obtain (22), as desired.

Now we prove II. We can compute that

$$\nabla^2_{\theta',\theta}\mathcal{H}_2(\theta,\theta') = \nabla_{\theta'}\nabla_\theta \mathbb{E}_{x\sim\pi_\gamma p_\theta}[\log p_{\theta'}(x)] \tag{25}$$

$$= \nabla_{\theta'}\nabla_\theta \int_{x\in\mathbb{R}^d} \log p_{\theta'}(x)\left(\frac{p_\theta(x)+\gamma p_{\theta^\star}(x)}{1+\gamma}\right)dx \tag{26}$$

$$= \frac{1}{1+\gamma}\nabla_{\theta'}\nabla_\theta \int_{x\in\mathbb{R}^d} \log p_{\theta'}(x)p_\theta(x)dx \tag{27}$$

$$= \frac{1}{1+\gamma}\nabla^2_{\theta',\theta}\mathbb{E}_{x\sim p_\theta}[\log p_{\theta'}(x)] \tag{28}$$

$$= \frac{1}{1+\gamma}B \tag{29}$$

where the third equality holds because the integral containing $p_{\theta^\star}$ is constant with respect to θ. Next, we can compute that

$$B = \int_X \nabla_{\theta'}\log p_{\theta'}(x)\nabla_\theta p_\theta(x)dx\Big|_{\theta^*,\theta^*} \tag{30}$$

$$= \int_X [\nabla_\theta \log p_\theta(x)][\nabla_\theta p_\theta(x)]dx\Big|_{\theta^*,\theta^*} \tag{31}$$

$$= \int_X \nabla_\theta[p_\theta(x)\nabla_\theta\log p_\theta(x)]dx\Big|_{\theta^*,\theta^*} - \int_X p_\theta(x)(\nabla_\theta\nabla_\theta\log p_\theta(x))dx\Big|_{\theta^*,\theta^*} \tag{32}$$

$$= \int_X \nabla_\theta\left[p_\theta(x)\frac{\nabla_\theta p_\theta(x)}{p_\theta(x)}\right]dx\Big|_{\theta^*,\theta^*} - \nabla^2_{\theta',\theta'}\mathbb{E}_{x\sim p_\theta}[\log p_{\theta'}(x)]\Big|_{\theta^*,\theta^*} \tag{33}$$

$$= -C, \tag{34}$$

where the third equality follows from the product rule for gradients,

$$\nabla_\theta[p_\theta(x)\nabla_\theta\log p_\theta(x)] = p_\theta(x)(\nabla_\theta\nabla_\theta\log p_\theta(x)) + [\nabla_\theta p_\theta(x)][\nabla_\theta\log p_\theta(x)]. \tag{35}$$

Finally, we will prove the formula (23) by manipulating (22). We begin with the rightmost factor in (22). If we apply these equalities that we just obtained, then we get

$$\begin{aligned}
\mathcal{J}(\pi\mathcal{G}(\theta^\star)) &= -\left(\nabla^2_{\theta',\theta'}\mathcal{H}(\theta^\star,\theta^\star)\right)^{-1}\cdot\lambda\nabla^2_{\theta',\theta}\mathcal{H}_2(\theta^\star,\theta^\star)\\
&= -(A+\lambda C)^{-1}\cdot\frac{\lambda}{1+\gamma}B\\
&= -(I+\lambda A^{-1}C)^{-1}\cdot\frac{\lambda}{1+\gamma}A^{-1}B\\
&= (I+\lambda A^{-1}C)^{-1}\cdot\frac{\lambda}{1+\gamma}A^{-1}C
\end{aligned}$$

where the first equality follows from (23) along with the fixed point Proposition A.6, and we are using that A is invertible by Assumption 2 A.4, which implies all eigenvalues of A are nonzero; in the fourth step we used that $B = -C$. This proves part II.

Now we prove III. We can bound the operator norm $\|A^{-1}C\|$ as follows:

$$\|A^{-1}C\| = \|I + A^{-1}(C-A)\| \le \|I\| + \|A^{-1}\|\cdot\|C-A\| \le 1 + \alpha^{-1}\|C-A\|, \tag{36}$$

where the first estimate comes from subadditivity and submultiplicativity, and the second comes from the fact that, since A is symmetric, $\|A\| = \max_{\lambda\in\sigma(A)}|\lambda|$, where $\sigma(A)$ is the spectrum of A. Formally, we know by Assumption A.4 that A has eigenvalues $e_1 < e_2 < \cdots < e_n \le -\alpha < 0$ and so $|e_n| > \alpha$. Therefore, A^{-1} has eigenvalues $1/e_n < 1/e_{n-1} < \cdots < 1/e_1 < 0$ and thus $1/|e_n| > 1/|e_{n-1}| > \cdots > 1/|e_1|$, which gives us the bound $\|A^{-1}\| = 1/|e_n| < 1/\alpha$ on the matrix

norm. Next, we can estimate that

$$\begin{aligned}||C - A|| &= \|\nabla^2_{\theta',\theta'}\mathbb{E}_{x\sim p_{\theta^\star}}[\log p_{\theta'}(x)]|_{\theta^\star} - \nabla^2_{\theta',\theta'}\mathbb{E}_{x\sim p_{\text{data}}}[\log p_{\theta'}(x)]|_{\theta^\star}\| \\ &= \|\mathbb{E}_{x\sim p_{\theta^\star}}[\nabla^2_{\theta',\theta'}\log p_{\theta^\star}(x)] - \mathbb{E}_{x\sim p_{\text{data}}}[\nabla^2_{\theta',\theta'}\log p_{\theta^\star}(x)]\| \\ &\leq L d_W(p_{\theta^\star}, p_{\text{data}}) \\ &= L\varepsilon,\end{aligned}$$

where in the second equality we exchange the derivative and the expectation in equation 4 using the Dominated Convergence Theorem, since Assumption 1 A.3 says that $x \mapsto \nabla_\theta \nabla_\theta \log p_\theta(x)$ is L-Lipschitz; and in the last estimate, we used Kantorovich-Rubenstein duality. This, combined with the estimate (36), yields the bound in (24). □

We are finally ready to prove our theorem that guarantees convergence to the optimal parameters in the infinite sampling case under certain assumptions, one being the that the initial model parameters θ_0 are sufficiently close to $\theta^\star$:

Theorem A.8 (Convergence of Iterative Fine-tuning, Infinite Sampling Case)**.** *Suppose we have an iterative fine-tuning procedure defined by the rule $\theta^\infty_{t+1} = \pi_\gamma \mathcal{G}^\infty_\lambda(\theta^\infty_t)$. Let $\theta^\star$ be the parameter vector for the optimal generative model, as in (9). We assume that $\theta^\star$ follows Assumptions A.3 and A.4 from (Bertrand et al., 2024). Suppose also that $\lambda\left(1 + \frac{\varepsilon L}{\alpha}\right) < \frac{1+\gamma}{2+\gamma}$. Then, the Jacobian of $\pi_\gamma G^\infty_\lambda$ satisfies the following bound:*

$$\|\nabla_\theta \pi_\gamma \mathcal{G}^\infty_\lambda(\theta^\star)\|_2 \leq \frac{1}{1+\gamma} \cdot \frac{\lambda(\alpha + \varepsilon L)}{\alpha - \lambda(\alpha + \varepsilon L)} < 1. \tag{37}$$

Consequently, there exists a $\delta > 0$ such if $\theta_0 \in \Theta$ satisfies $\|\theta_0 - \theta^\star\| \leq \delta$, then starting training at θ_0 and having $\theta_{t+1} = \pi_\gamma \mathcal{G}^\infty_\lambda(\theta_t)$, we have that $\lim_{t\to\infty} \theta_t \to \theta^\star$. Furthermore, if we define

$$\rho(\lambda) = \frac{\lambda(\alpha + \varepsilon L)}{\alpha - \lambda(\alpha + \varepsilon L)}, \tag{38}$$

then we obtain the asymptotic stability estimate[3]

$$\|\theta_t - \theta^\star\| \leq \left(\frac{\rho(\lambda)}{1+\gamma}\right)^t \|\theta_0 - \theta^\star\|. \tag{39}$$

Proof. **We first prove the Jacobian bound** (37)**.** By hypothesis, we know $\lambda(1 + \frac{L\varepsilon}{\alpha}) < 1$, so by Lemma A.7(III), we have $\lambda||A^{-1}C|| < 1$. Thus, we can write

$$(I + \lambda A^{-1}C)^{-1} = \sum_{k=0}^{\infty}(-\lambda A^{-1}C)^k$$

and so

$$\|(I + \lambda A^{-1}C)^{-1}\| \leq \sum_{k=0}^{\infty} \lambda^k ||A^{-1}C||^k = \frac{1}{1 - \lambda||A^{-1}C||}.$$

Applying Lemma A.7(2), we get

$$||\mathcal{J}(G(\theta^\star))|| \leq ||(I + \lambda A^{-1}C)^{-1}|| \cdot \frac{\lambda}{1+\gamma}||A^{-1}C|| \leq \frac{\lambda}{1+\gamma} \cdot \frac{||A^{-1}C||}{1 - \lambda||A^{-1}C||}.$$

Now, it is straightforward to see the RHS above is at most the bound in (37) if and only if $\alpha\|A^{-1}C\| < \alpha + \varepsilon L$. But this bound holds because of Lemma A.7(III). This proves the Jacobian bound (37), but does not prove that the bound is less than 1. For this, we must show that

$$\frac{1}{1+\gamma} \cdot \frac{\lambda(\alpha + \varepsilon L)}{\alpha - \lambda(\alpha + \varepsilon L)} < 1. \tag{40}$$

[3](Bertrand et al., 2024) could have presented their results in this stronger form, without the big O notation, with very little extra work.

By clearing denominators and grouping like terms, we can see that this is equivalent to

$$\lambda\left(1+\frac{\varepsilon L}{\alpha}\right)<\frac{1+\gamma}{2+\gamma}, \tag{41}$$

which is precisely guaranteed by our hypothesis.

We now apply the the Jacobian bound (37) to prove the asymptotic stability estimate (39). Assume λ is sufficiently small so that $\rho(\lambda)/(1+\gamma)<1$. Then for every $\rho'\in(\rho(\lambda)/(1+\gamma),1)$, there exists $\delta>0$ sufficiently small so that every $\theta_0\in\Theta$ which satisfies $\|\theta_0-\theta^\star\|<\delta$ has the property that $\|\nabla_\theta\pi_\gamma G_\lambda^\infty(\theta_0)\|_2<\rho'$. Because the map $\pi_\gamma G_\lambda^\infty$ has Jacobian matrix norm less than 1 in the δ-ball around $\theta^\star$, it is a contraction mapping in this neighborhood. Concretely, this means that

$$\|\pi_\gamma\mathcal{G}_\lambda^\infty(\theta)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta')\|\leq\frac{\rho(\lambda)}{1+\gamma}\|\theta-\theta'\|, \tag{42}$$

for every θ,θ' in the δ-ball around $\theta^\star$. In particular, for $(\theta,\theta')=(\theta_t,\theta^\star)$ we obtain

$$\|\theta_{t+1}-\theta^\star\|=\|\pi_\gamma\theta_t-\theta^\star\|=\|\pi_\gamma\mathcal{G}_\lambda^\infty(\theta_t)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta^\star)\|\leq\frac{\rho(\lambda)}{1+\gamma}\cdot\|\theta_t-\theta^\star\|.$$

By induction, the above estimate implies that if θ_0 is in a δ-ball around $\theta^\star$, then so is every successive θ_t. Therefore the desired estimate (39) now follows by induction on t. □

Remark A.9. Taking $\gamma=0$ recovers exactly the result in (Bertrand et al., 2024). Importantly, the correction function π_γ provides leverage in determining how large the augmentation percentage λ can be: choosing a larger correction strength γ allows us to choose a larger augmentation percentage λ while still retaining theoretical guarantees for convergence. Additionally, for the same choice of augmentation percentage λ, a larger correction strength γ provides a guarantee for an improved rate of convergence. See Conjecture 4.7.

A.3. Stability of Iterative Fine-tuning with Correction for Finite Sampling

Finally, we prove a stability result for iterative fine-tuning with correction in the presence of statistical error. To do this, we require an assumption that essentially provides probabilistic guarantee that the chosen generative model learns the underlying distribution increasingly better if it has access to more samples:

Assumption A.10. There exist $a,b,\varepsilon_{\text{OPT}}\geq 0$ and a neighborhood U of $\theta^\star$ such that, for any $\delta\in(0,1)$, with probability $1-\delta$ over the samplings, we have

$$(\forall\theta\in U)(\forall n\in\mathbb{N})\qquad\|\pi_\gamma\mathcal{G}_\lambda^n(\theta)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta)\|\leq\varepsilon_{\text{OPT}}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}. \tag{43}$$

See Appendix B for a discussion about this assumption; we investigated whether to assume a similar bound to the one they assumed in (Bertrand et al., 2024), or prove our bound from theirs. In fact, we prove in Appendix B that you can in fact deduce something nearly as strong as Assumption A.10 from Assumption 3 in their paper, so we made Assumption A.10 for the sake of a cleaner, more parallel exposition.

Theorem A.11 (Iterative Fine-Tuning Stability Under Correction)**.** *Suppose we have an iterative fine-tuning procedure defined by the rule $\theta_{t+1}^n=\pi_\gamma\mathcal{G}_\lambda^n(\theta_t^n)$. In words, this means that the augmentation percentage is $\lambda\in(0,\infty)$ and the correction strength is $\gamma\in[0,\infty)$. Under the same assumptions of Theorem A.8 and Assumption A.10, there exist $0<\rho<1$ and $\delta_1>0$ such that if $\|\theta_0^n-\theta^\star\|\leq\delta_1$, then for any $\delta_2\in(0,1)$, with probability $1-\delta_2$, we have*

$$\|\theta_t^n-\theta^\star\|\leq\left(\varepsilon_{OPT}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{bt}{\delta}}\right)\sum_{i=0}^{t}\left(\frac{\rho(\lambda)}{1+\gamma}\right)^i+\left(\frac{\rho(\lambda)}{1+\gamma}\right)^t\|\theta_0^n-\theta^\star\|. \tag{44}$$

Proof. By the triangle inequality, we can estimate that

$$\begin{aligned}\|\theta_t^n-\theta^\star\|&\leq\|\theta_t^n-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta_{t-1}^n))\|+\|\pi_\gamma\mathcal{G}_\lambda^\infty(\theta_{t-1}^n)-\theta^\star\|\\&=\|\pi_\gamma\mathcal{G}_\lambda^n(\theta_{t-1}^n)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta_{t-1}^n)\|+\|\pi_\gamma\mathcal{G}_\lambda^\infty(\theta_{t-1}^n)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta^\star)\|,\end{aligned} \tag{45}$$

where we applied the fixed point Proposition A.6. By Assumption A.10, the left summand in (45) is at most $\varepsilon_{\text{OPT}}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}$, with probability $1-\delta$. Next, recall that in (42) in the proof of Theorem A.8, we proved that that $\pi_\gamma G_\lambda^\infty$ is a contraction mapping of factor $\rho(\lambda)/(1+\gamma)$ sufficiently close to U; this implies that the right summand in (45) is at most $\frac{\rho(\lambda)}{1+\gamma}\|\theta_{t-1}^n-\theta^\star\|$. Together, these yield the recurrence estimate

$$\mathbb{P}\left(\|\theta_t^n-\theta^\star\|\leq\varepsilon_{\text{OPT}}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}+\frac{\rho(\lambda)}{1+\gamma}\|\theta_{t-1}^n-\theta^\star\|\right)\geq 1-\delta. \tag{46}$$

Iterating this recurrence for successive time steps yields

$$\mathbb{P}\left(\|\theta_t^n-\theta^\star\|\leq\left(\varepsilon_{\text{OPT}}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}\right)\sum_{i=0}^{t}\left(\frac{\rho(\lambda)}{1+\gamma}\right)^i+\left(\frac{\rho(\lambda)}{1+\gamma}\right)^t\|\theta_0^n-\theta^\star\|\right)\geq(1-\delta)^t. \tag{47}$$

Note that (47) holds for any $\delta\in(0,1)$. In particular, we can apply (47) with $\delta:=\delta/t$. In this case, the Bernoulli inequality lets us estimate that $(1-\delta/t)^t\geq 1-\delta$. This completes the proof, with $\delta_2=\delta$. □

Remark A.12. Theorem A.11 recovers the result from (Bertrand et al., 2024) in the case where the correction strength is $\gamma=0$. But for a fixed augmentation percentage λ, for any correction strength $\gamma>0$, this gives stronger stability guarantees than in (Bertrand et al., 2024).

Remark A.13. In a previous version of this manuscript, we claimed that there was an error in the statement of the corresponding theorem in (Bertrand et al., 2024). In this version, we retract that claim; we have corresponded with those authors, and they updated their manuscript with additional details to justify their statement.

A.4. Discussion: The Main Limitation

Our empirical results are for generative modeling tasks where we have access to some "self-correction" operation that is easy to compute, as well as automatic; see Sections 6 and 7 for more details about these correction functions. Therefore, the main limitation of our work is that one can only hope to use this self-correction procedure to stabilize training in scenarios where there is some "self-correction" function. For our MNIST experiments, we built a self-correction function from scratch using clustering statistics. And for our human motion experiments, we used an off-the-shelf human motion imitation model that other researchers built.

B. Discussion about Assumption 4.2

In this section, we show how with a mild boundedness assumption on our generative model parameter update function, we can deduce our Assumption A.10 (which is the same as Assumption 4.2, part 3) from the following assumption used in (Bertrand et al., 2024).

Assumption B.1. There exist $a,b,\varepsilon_{\text{OPT}}\geq 0$ and a neighborhood U of $\theta^\star$ such that, for any $\delta\in(0,1)$, with probability $1-\delta$ over the samplings, we have

$$(\forall\theta\in U)(\forall n\in\mathbb{N})\qquad\|\mathcal{G}_\lambda^n(\theta)-\mathcal{G}_\lambda^\infty(\theta)\|\leq\varepsilon_{\text{OPT}}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}. \tag{48}$$

Now, if we make the additional assumption that our generative model parameter update function is locally bounded near $\theta^\star$ then we obtain the following.

Proposition B.2. *Suppose Assumption B.1 holds. Suppose also that there exists $B<\infty$ such that for all $n>0$ and θ sufficiently close to $\theta^\star$,*

$$\|\mathcal{G}_\lambda^n(\theta)-\mathcal{G}_\lambda^n(\theta^\star)\|<B\|\theta-\theta^\star\|.$$

Then there exist $a,b,c,\varepsilon_{OPT}\geq 0$ and a neighborhood U of $\theta^\star$ such that, for any $\delta\in(0,1)$, with probability $1-\delta$ over the samplings, we have

$$(\forall\theta\in U)(\forall n\in\mathbb{N})\qquad\|\pi_\gamma\mathcal{G}_\lambda^n(\theta)-\pi_\gamma\mathcal{G}_\lambda^\infty(\theta)\|\leq c\cdot d_U+\varepsilon_{OPT}+\frac{a}{\sqrt{n}}\sqrt{\log\frac{b}{\delta}}, \tag{49}$$

where $d_U=\sup_{\theta\in U}\|\theta-\theta^\star\|$.

Proof. By the triangle inequality, we have

$$\|\pi_\gamma \mathcal{G}_\lambda^n(\theta) - \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)\| \leq \|\pi_\gamma \mathcal{G}_\lambda^n(\theta) - \mathcal{G}_\lambda^n(\theta)\| + \|\mathcal{G}_\lambda^n(\theta) - \mathcal{G}_\lambda^\infty(\theta)\| + \|\mathcal{G}_\lambda^\infty(\theta) - \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)\|. \tag{50}$$

We bound each term in the RHS: firstly, note the middle term is bounded by Assumption B.1.The first term is bounded as follows:

$$\begin{aligned}\|\mathcal{G}_\lambda^n(\theta) - \pi_\gamma \mathcal{G}_\lambda^n(\theta)\| &\leq \|\mathcal{G}_\lambda^n(\theta) - \mathcal{G}_\lambda^n(\theta^\star)\| + \|\pi_\gamma \mathcal{G}_\lambda^n(\theta^\star) - \pi_\gamma \mathcal{G}_\lambda^n(\theta)\| \\ &\leq B\|\theta - \theta^\star\| + B\|\theta - \theta^\star\| \\ &\leq 2Bd_U,\end{aligned}$$

where in the first step we used that $\mathcal{G}_\lambda^\infty(\theta^\star) = \pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star)$. Similarly, the last term is bounded as follows:

$$\begin{aligned}\|\mathcal{G}_\lambda^\infty(\theta) - \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)\| &\leq \|\mathcal{G}_\lambda^\infty(\theta) - \mathcal{G}_\lambda^\infty(\theta^\star)\| + \|\pi_\gamma \mathcal{G}_\lambda^\infty(\theta^\star) - \pi_\gamma \mathcal{G}_\lambda^\infty(\theta)\| \\ &\leq \rho(\lambda)\|\theta - \theta^\star\| + \frac{\rho(\lambda)}{1+\gamma}\|\theta - \theta^\star\| \\ &= \rho(\lambda)\frac{2+\gamma}{1+\gamma}\|\theta - \theta^\star\| \\ &\leq \rho(\lambda)\frac{2+\gamma}{1+\gamma}d_U,\end{aligned}$$

where in the second step we applied (42). Using these bounds in (50) and taking $c = 2B + \rho(\lambda)\frac{2+\gamma}{1+\gamma}$ completes the proof. □

Note that the constant $c \cdot d_U < c$ (for U sufficiently small) can really be viewed as a part of the optimization constant ε_{OPT} since it is controlled by the choice of generative model class.

C. Point-wise correction corresponds to distribution-wise correction

In this section we provide a sufficient condition under which you can associate a distribution-wise correction mapping (like the one we consider in the paper, π_γ) to a point-wise correction mapping (which is the one you are more likely to find in the wild).

Definition C.1. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^m$ and define the *empirical cumulative distribution* function Φ_X by

$$\Phi_X(v) := \Phi_X(v; \{x_1, \ldots, x_n\}) := \frac{1}{n}\sum_{i=1}^n \chi_v(x_i),$$

where for $v \in \mathbb{R}^m$, $\chi_v : \mathbb{R}^m \to \{0, 1\}$ is the indicator function for the set $\prod_{i=1}^n (-\infty, v_i]$. For a continuous distribution, the cumulative distribution function is defined in the usual way.

Definition C.2. Suppose that we have a model p_θ and an arbitrary function $\Pi : \mathbb{R}^m \to \mathbb{R}^m$. Then we say that Π is a *valid point-wise correction function* for p_θ if there exists a $\gamma \in [0, \infty]$ such that

$$\lim_{n\to\infty}\left(\mathbb{E}_{X^n \sim p_\theta^n} \sup_{v\in\mathbb{R}^m} \|\Phi_{\Pi(X^n)}(v) - \Phi_{\pi_\gamma p_\theta}(v)\|\right) \to 0, \tag{51}$$

almost surely, where the expectation is over all samplings $X^n = \{x_1, \ldots, x_n\}$ of size n from p_θ.

Intuition C.3. This is saying that the CDFs for $\pi_\gamma p_\theta$ and $\Pi(X \sim p_\theta^n)$ are equal in expectation, for large enough n. This is one way of saying that $\pi_\gamma p_\theta$ and $\Pi(X \sim p_\theta^n)$, for large enough n, are nearly identical probability distributions.

Definition C.4. If the limit in (51) exists, then we define the *distribution-wise projection function* corresponding to Π to be

$$\pi_\gamma p_\theta = \frac{1}{1+\gamma}p_\theta + \frac{\gamma}{1+\gamma}p_{\theta^\star}, \tag{52}$$

and we define the *projection strength of the point-wise correction function* Π to be γ. Recall that $\pi_\gamma p_\theta = \frac{1}{1+\gamma}p_\theta + \frac{\gamma}{1+\gamma}p_{\theta^\star}$. So intuitively, (51) implies that the projection function Π maps samples from p_θ to a different space such that they look like they come from a combination of the original distribution p_θ and $p_{\theta^\star}$, at least at the level of CDFs.

Remark C.5. Such a γ, if it exists, is unique. Furthermore, if $p_\theta = p_{\theta^\star}$, then $\gamma = \infty$.

The limit condition in Definition C.2 is abstract, and can be hard to swallow. We present an example of a simple point-wise correction for the Gaussian toy example that we consider in Section 5, whose corresponding distribution-wise correction is exactly one would expect it to be–the weighted average of the corresponding Gaussians. Recall that we demonstrated empirically in Figure 2 that Theorem 4.3 holds for that example. The projection function is depicted in Figure 6.

Example C.6. Let $G_1(x)$ be the pdf of $\mathcal{N}(0, \sigma_1^2 I_d)$ (initial distribution, corresponds to θ) and $G_2(x)$ the pdf of $\mathcal{N}(0, \sigma_2^2 I_d)$ (target distribution, corresponds to $\theta^\star$). Given $x_1, \ldots, x_n \sim G_1$, we define Π^γ as follows: Fix any $\gamma \in \mathbb{R}_{\geq 0}$, and let $y_1, \ldots, y_n \sim (\hat{G}_1^{(n)}(x) + \gamma G_2(x))/(1+\gamma)$, where $\hat{G}_1^{(n)}$ is the PDF of the empirical distribution defined by $\{x_1, \ldots, x_n\}$; in practice we implement $\hat{G}_1^{(n)}$ as a histogram. Then choose a random $\sigma \in S_n$ (S_n = group of permutations on n symbols). Finally, we define $\Pi^\gamma(x_i) := y_{\sigma(i)}$ for $1 \leq i \leq n$.

Next, we define the projection set $\Pi X^{(n)} := \{\Pi^\gamma(x_i)\}_{1 \leq i \leq n}$, and define the PDF $\pi_\gamma \hat{G}_1^{(n)}(x) := \frac{1}{1+\gamma}\hat{G}_1^{(n)}(x) + \frac{\gamma}{1+\gamma}G_2(x)$, and let $\Phi_{\pi_\gamma \hat{G}_1^{(n)}}$ represent the cumulative distribution function of the Gaussian $\pi_\gamma \hat{G}_1^{(n)}$. Then, since $\Pi^\gamma(x_i) \sim \pi_\gamma \hat{G}_1^{(n)}$, we have by the uniform law of large numbers that

$$\lim_{n\to\infty} \left(\mathbb{E}_{\{x_i \sim G_1\}_{i=1}^n} \sup_{v \in \mathbb{R}^m} \left\|\Phi_{\Pi X^{(n)}}(v) - \Phi_{\pi_\gamma G_1}(v)\right\|\right) \to 0 \tag{53}$$

almost surely. Therefore Π^γ is a valid point-wise correction function, and its corresponding distribution-wise projection function is π_γ.

Remark C.7. In the example we considered in Section 5, we could have included a total distance traveled minimization condition, but here for this proof we don't even need to use that hypothesis. (In the proof, this would have corresponded to the additional assumption that we've chosen a $\sigma \in S_n$ such that $\sum_{i=1}^n \|x_i - y_{\sigma(i)}\|$ is minimized.) This implies that different point-wise correction functions can correspond to the same distribution-wise correction function.

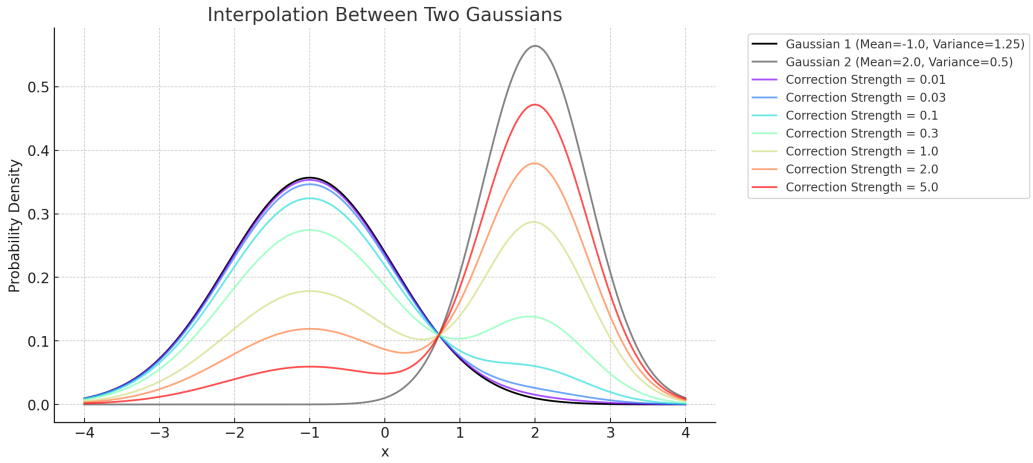


Figure 6. Illustration of the distribution-wise projection function, like in our Gaussian toy example. Correcting one Gaussian in the direction of another, like we consider in Section 5, corresponds to finding the "(weighted) average Gaussian" that lives between the two.

D. More MNIST Experiment Details

We train a Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) on the 20% of the MNIST dataset (LeCun et al., 1998). We use classifier-free guidance (Ho & Salimans, 2021) with guidance parameter 0.5, and 400 diffusion steps. We used a batch size of 256. We train generation 0 for 20 epochs, with a linear decay learning rate schedule starting at $1e-4$ and ending at $(1e-4)/20$. We train each following generations for a single epoch, with a fixed learning rate of $(1e-4)/20^2$.

To compute our metrics, we first train a LeNet model (LeCun et al., 1998) on MNIST, and then we sample an equal number of digits from each class using the checkpoint that we're trying to evaluate. To compute the FID score, we extract embeddings

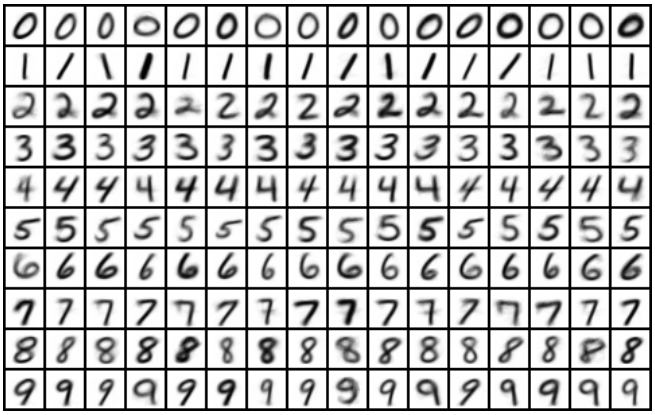


Figure 7. For every digit, we perform K-means clustering with K=16. We show here the cluster centroids, which intuitively are anchor images within the manifold of all possible images.

from the last fully connected LeNet layer for the synthesized examples, as well as for the held out test examples, and compute FID score as normal, by computing the Wasserstein distance between the Gaussians. Note that we use embeddings for LeNet trained on MNIST, rather than the Inception network trained on ImageNet, because MNIST isn't comprised of natural images. This is consistent with the convention in (Alemohammad et al., 2024).

For the self-correction operation, we compute the K-means clusters, with $K = 16$, once at the start of training. And we "correct" a synthesized motion by mapping it to the nearest cluster mean corresponding to its digit. In Figure 7 we present the clusters, and we present graphs of our FID scores across augmentation percentages in Figure 8.

E. Additional Human Motion Generation Qualitative Results

In Figures 9, 10, and 11, we present additional qualitative observations and analysis of our synthesized motions. We present more evidence that iterative fine-tuning with self-correction yields physically plausible motions comparable to the baseline, whereas iterative fine-tuning without self-correction yields motions that are incorrect for various reasons. See the captions of the referenced figures for analysis of some characteristic failure modes of the iterative fine-tuning loop *without* self-correction.

A technical note: for all figures, we render the motions from the same environment and camera position. We consolidate each render into the same image *without* resizing it. This means that if a figure appears larger relative to the others, the human moved closer to the camera. Some motions will have transparent frames of past positions; the more transparent the image, the farther back in the past it was in the motion sequence. Finally, in each figure, the text prompt for all generated motions was the same –the prompt being the one associated with the ground truth motion in the HumanML3D (Guo et al., 2022) training data, which we also visualize. Note that the coloring in the humanoid figures corresponds to the coloring in the graphs.

F. Additional Human Motion Generation Quantitative Results

See Figures 12, 13, 14 for results when the dataset size is $n \in \{64, 128, 256\}$ and the synthetic augmentation percentage is $\lambda \in \{0.25, 0.50, 0.75, 1.00\}$. And see Figures 15 and 16 for additional results on our iterative fine-tuning experiments when the dataset size is $n = 2794$ and the synthetic augmentation percentage is $\lambda \in \{0.05, 0.10, 0.15, 0.20, 0.25\}$. The graphs provide evidence across 17 experiment settings that our iterative fine-tuning procedure with self-correction yields better training performance than iterative fine-tuning with no self-correction for the motion synthesis task, in accordance with Theorem 4.3.

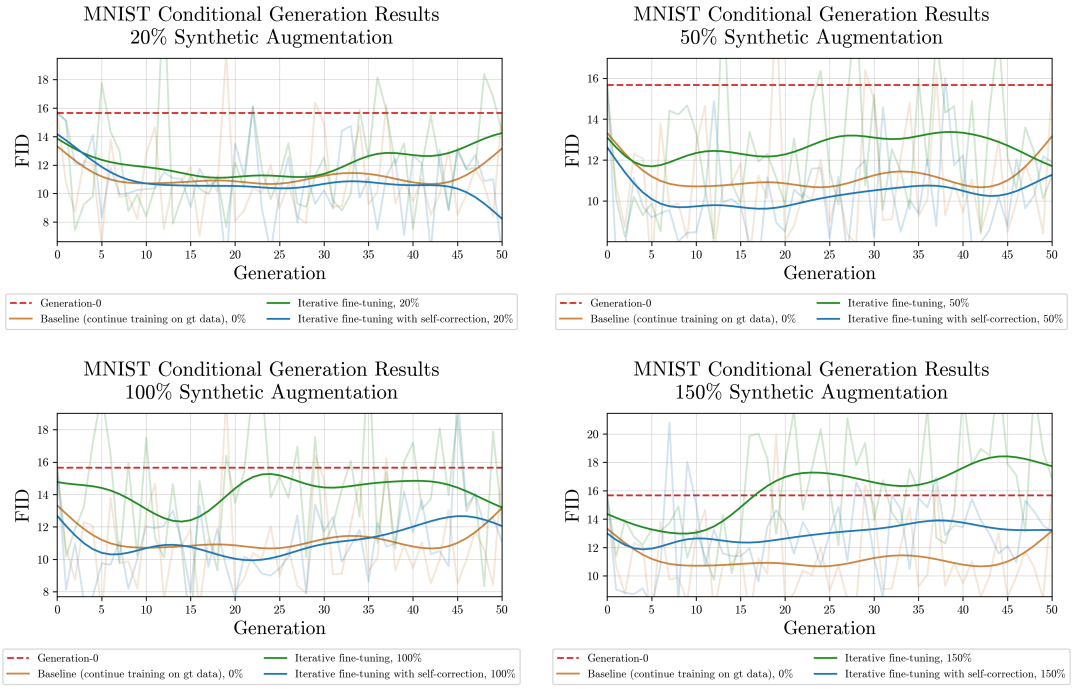


Figure 8. Results from MNIST experiments with iterative fine-tuning with and without self-correction. These graphs show the FID score on the last checkpoint for every generation; this is the checkpoint used for sampling in the self-consuming loop experiments, and it is also the checkpoint where training is resumed with this new partially synthesized dataset. These results demonstrate that **iterative fine-tuning with self-correction** generally outperforms **iterative fine-tuning.**

G. Consistency Across Seeds: Additional Human Motion Generation Quantitative Results

In Figures 17, 18, 19, and 20, we present experimental results from runs across three more seeds for our human motion experiments when the dataset size is $n = 64$. We find that the self-correction technique consistently yields improved training dynamics over iterative fine-tuning without correction.

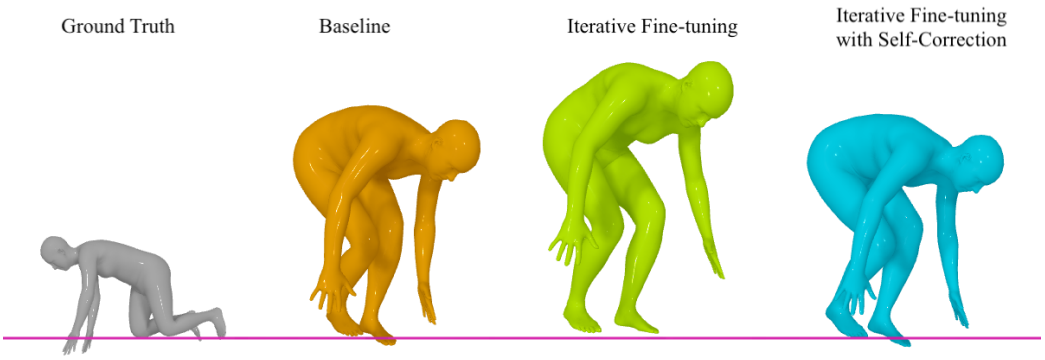


Figure 9. Here we see the negative *floating* phenomenon exacerbated by **iterative fine-tuning**, whereas **iterative fine-tuning with self-correction** generates a motion with floor contact integrity comparable to the **ground truth** and **baseline**. The floatic metric is formally defined in (Yuan et al., 2023) as the distance between the lowest vertex on the human mesh and the floor plane. All three sequences were generated using the same prompt: *person got down and is crawling across the floor.* Each snapshot was taken at exactly frame 87. The green figure appears larger than the other two only because it is closer to the camera. The two motions on the right were synthesized after 50 generations training with 25% synthetic augmentation, trained on $n = 64$ data points.

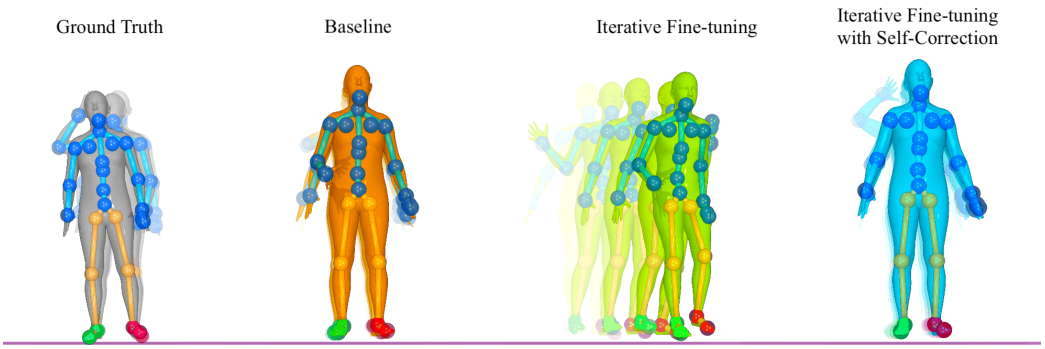


Figure 10. All four of the above motions correspond to the prompt: *a person raises right hand to face looks around and puts hand down back to side.*. The model which is trained with **iterative fine-tuning** outputs spurious motion that slides the figure to the right. And in the video for this example, the human rotates their forearm unnaturally and forcefully. In contrast, the **baseline** and **iterative fine-tuning with self-correction** models' motions both accurately embody the prompt. Each generated snapshot is taken at exactly frame 142 while the ground truth's image is frame 70 in its sequence. The two motions on the right were synthesized after 42 generations with 10% synthetic augmentation, where the ground truth dataset has size $n = 2794$.

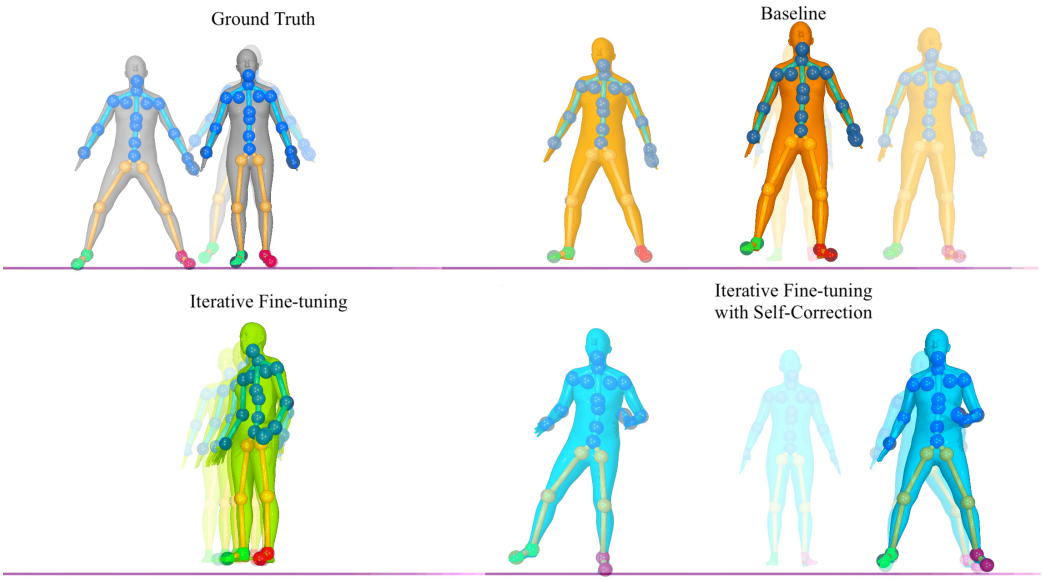


Figure 11. Here we observe that **iterative fine-tuning** fails to produce any meaningful motion sequence, but the **iterative fine-tuning with self-correction** and **baseline** models generate results consistent with their prompt: *walks side ways but back and forth*. Each snapshot for the generated motions was taken at exactly frame 120 while the ground truth image is a snapshot from frame 69. These images were synthesized after 50 generation of the model that was trained on $n = 64$ data points at 25% synthetic augmentation.

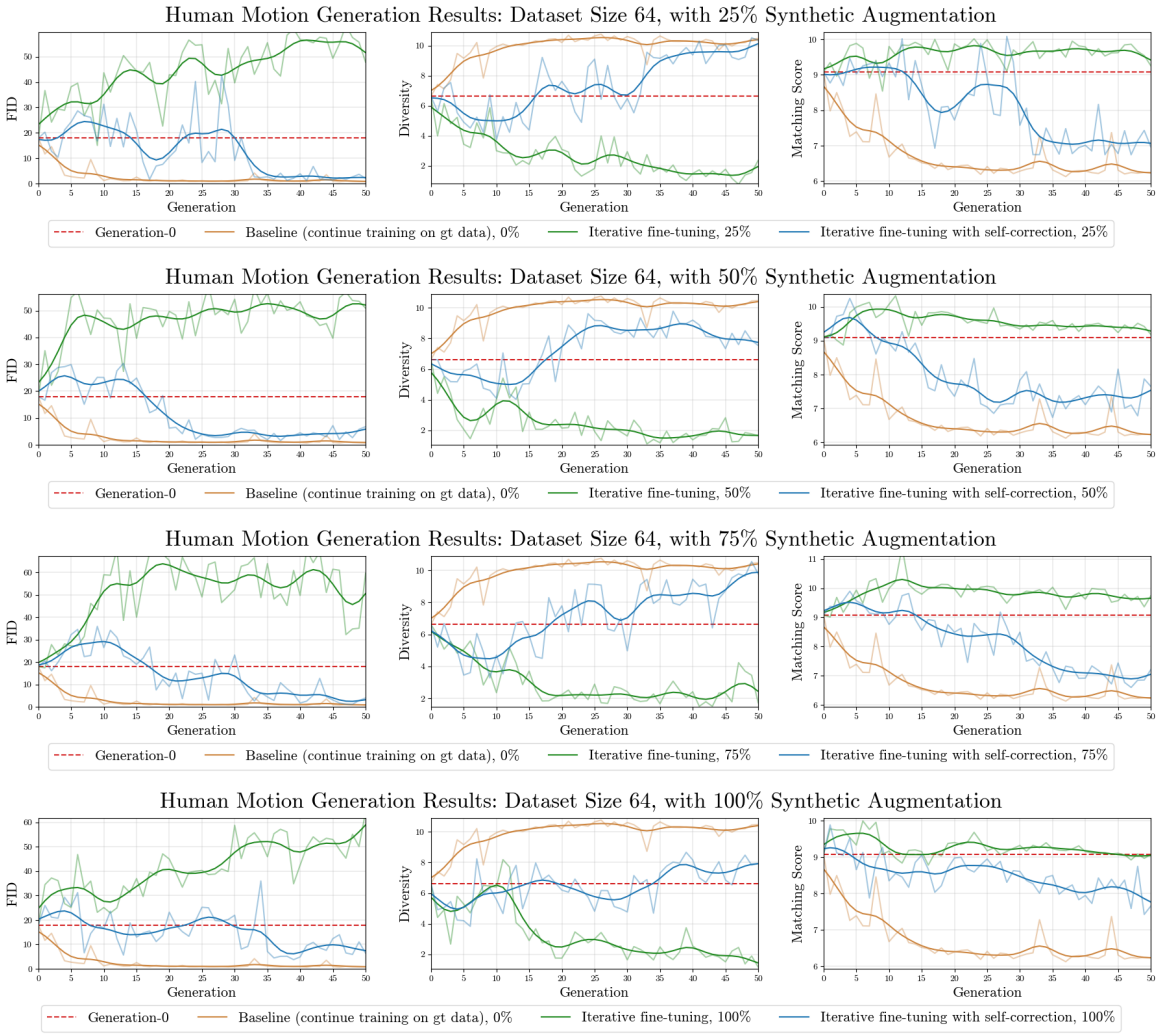


Figure 12. Results from our human motion experiments with iterative fine-tuning with and without self-correction, where the training set has size 64. These are graphs for evaluation metrics on the last checkpoint for every generation; this is the checkpoint used for sampling in the self-consuming loop experiments, and it is also the checkpoint where training is resumed with this new partially synthesized dataset. These results demonstrate that **iterative fine-tuning with self-correction** generally outperforms **iterative fine-tuning**, and is sometimes even competitive with **baseline** performance.

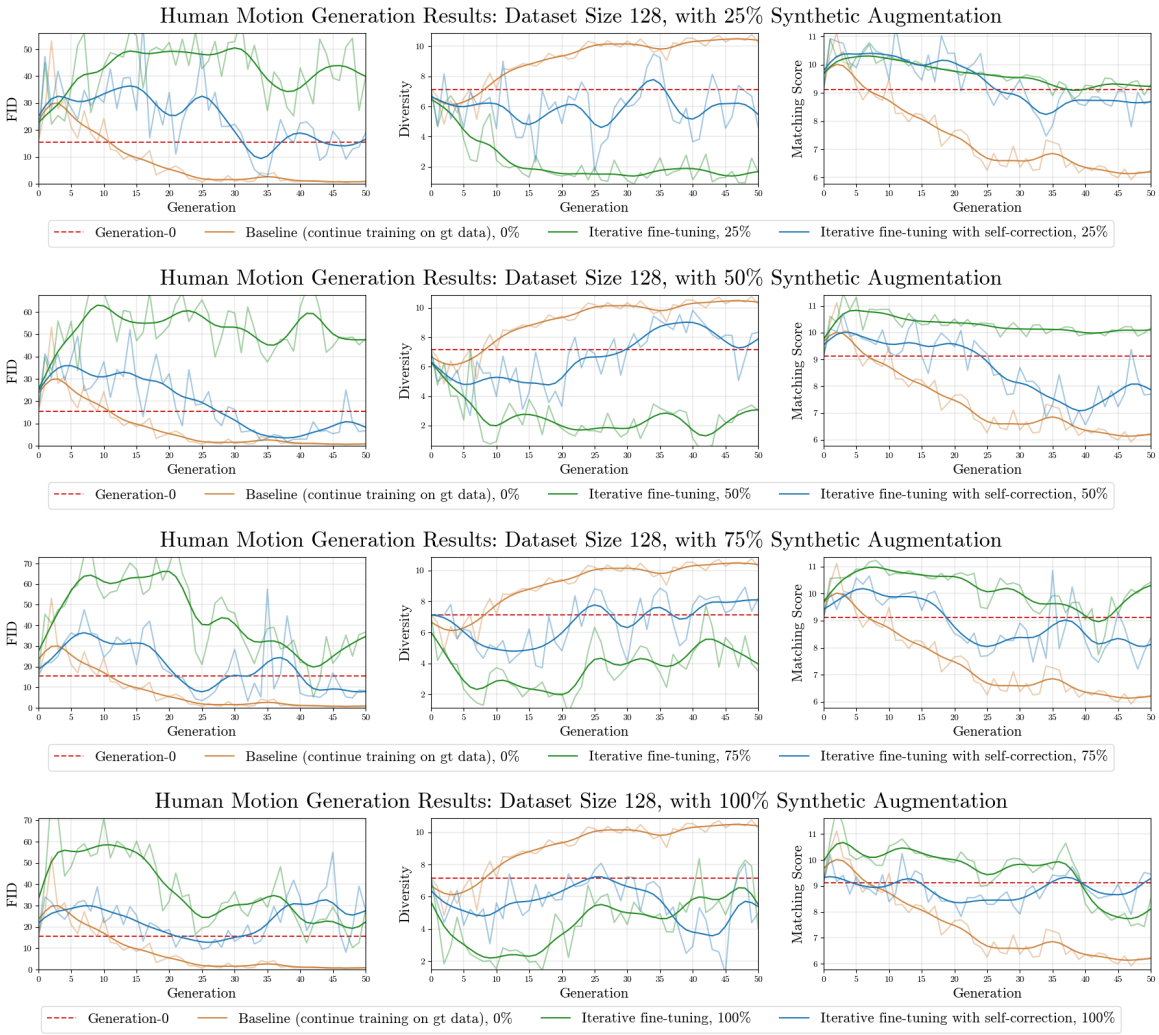


Figure 13. Results from our human motion experiments with iterative fine-tuning with and without self-correction, where the training set has size 128. These are graphs for evaluation metrics on the last checkpoint for every generation; this is the checkpoint used for sampling in the self-consuming loop experiments, and it is also the checkpoint where training is resumed with this new partially synthesized dataset. These results demonstrate that **iterative fine-tuning with self-correction** generally outperforms **iterative fine-tuning**, and is sometimes even competitive with **baseline** performance. Notably, the performance gain of **iterative fine-tuning with self-correction** over **iterative fine-tuning** is less pronounced than when the dataset size is $n = 64$.

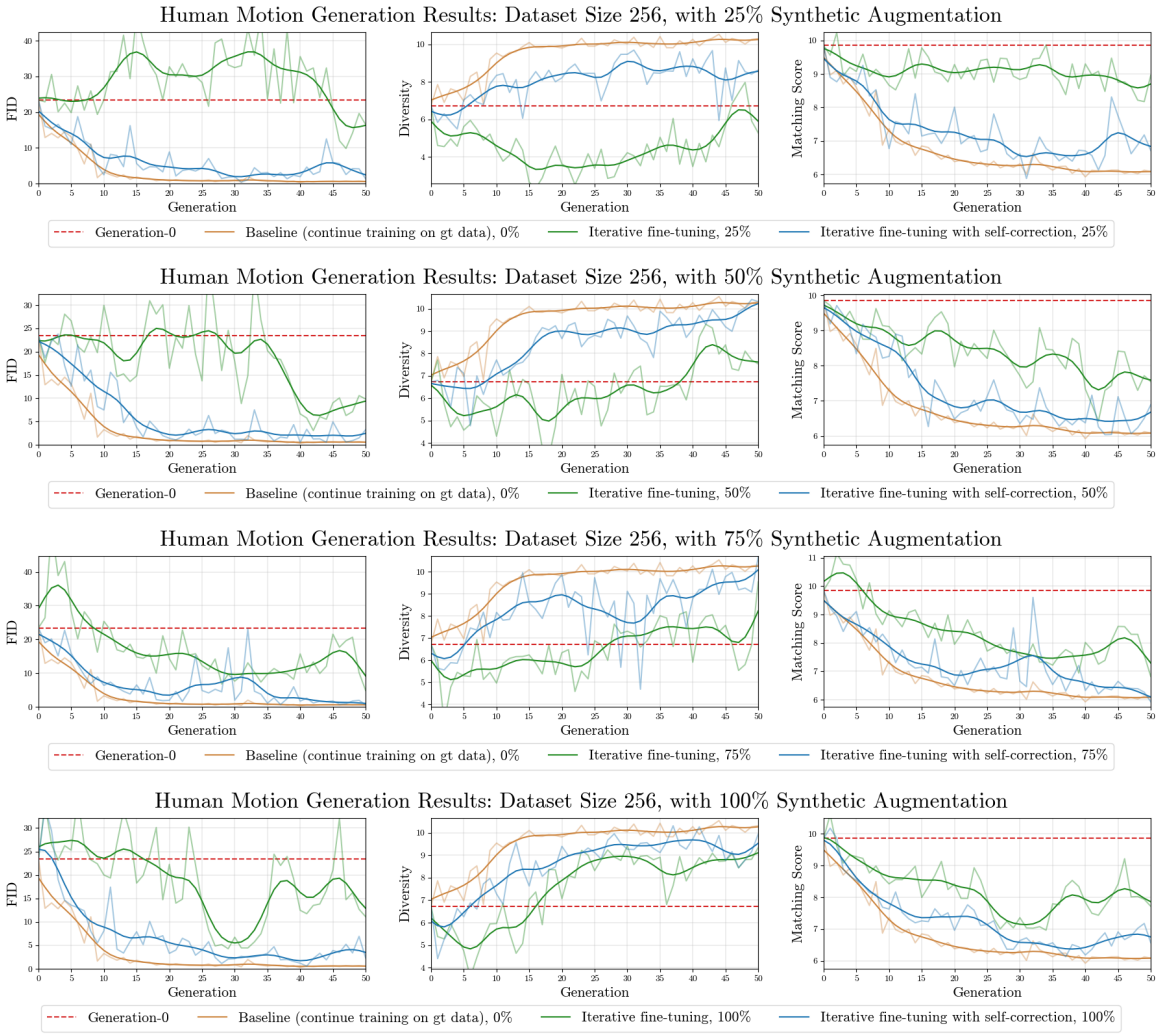


Figure 14. Results from our human motion experiments with iterative fine-tuning with and without self-correction, where the training set has size 256. These are graphs for evaluation metrics on the last checkpoint for every generation; this is the checkpoint used for sampling in the self-consuming loop experiments, and it is also the checkpoint where training is resumed with this new partially synthesized dataset. These results demonstrate that **iterative fine-tuning with self-correction** generally outperforms **iterative fine-tuning**, and is sometimes even competitive with **baseline** performance.

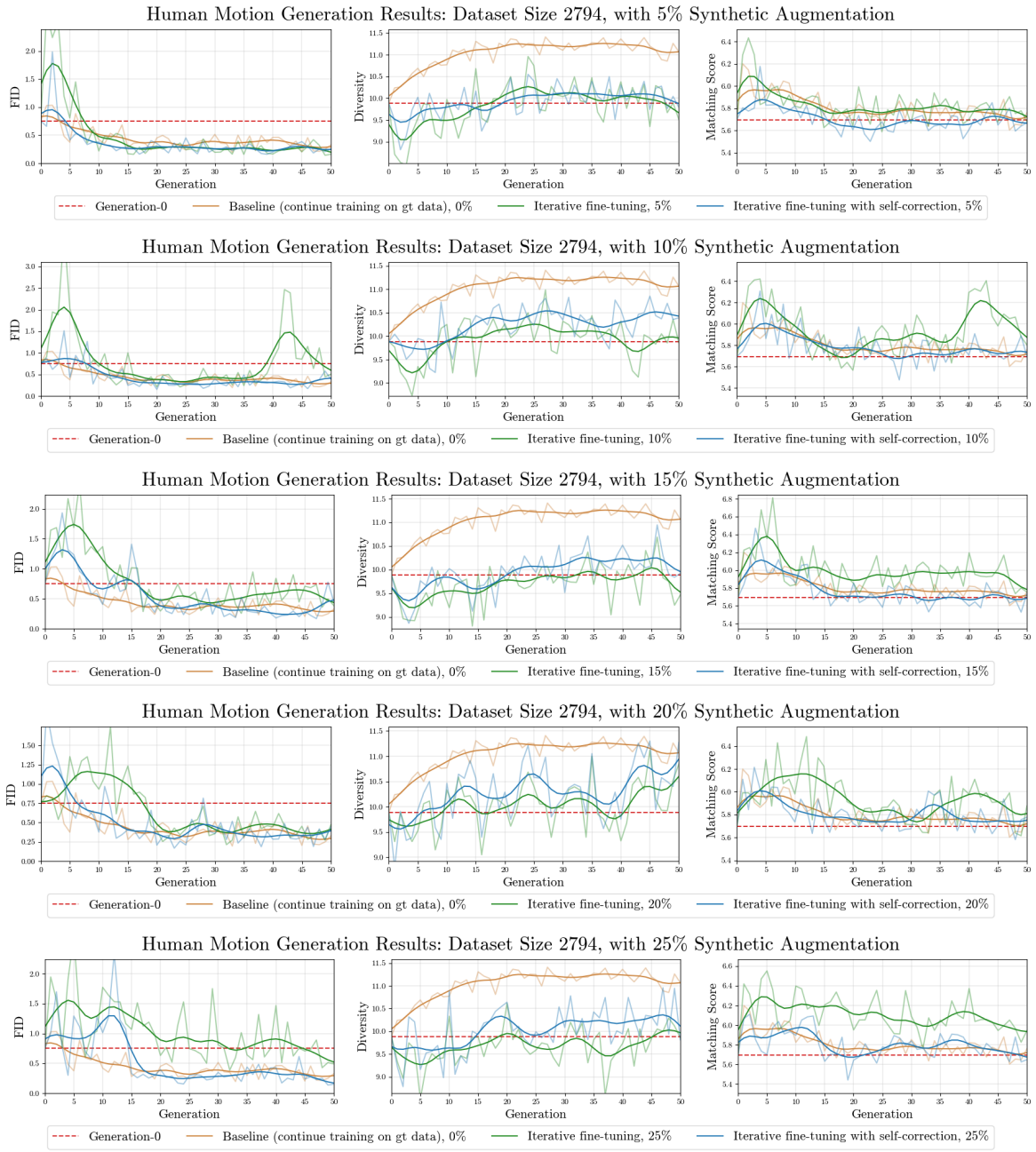


Figure 15. Results from our human motion experiments on iterative fine-tuning with dataset size $n = 2794$. These are graphs for evaluation metrics on the last checkpoint for every generation; this is the checkpoint used for sampling in the augmentation loop experiments, and it is also the checkpoint where training is resumed with this new synthesized dataset. In these results, it appears as though iterative fine-tuning **with self-correction** has less variance during training than iterative fine-tuning with **with no self-correction**, and generally has better FID scores later in training. Notably, the these two curves are closer together than they were in the cases $n \in \{64, 128, 256\}$.

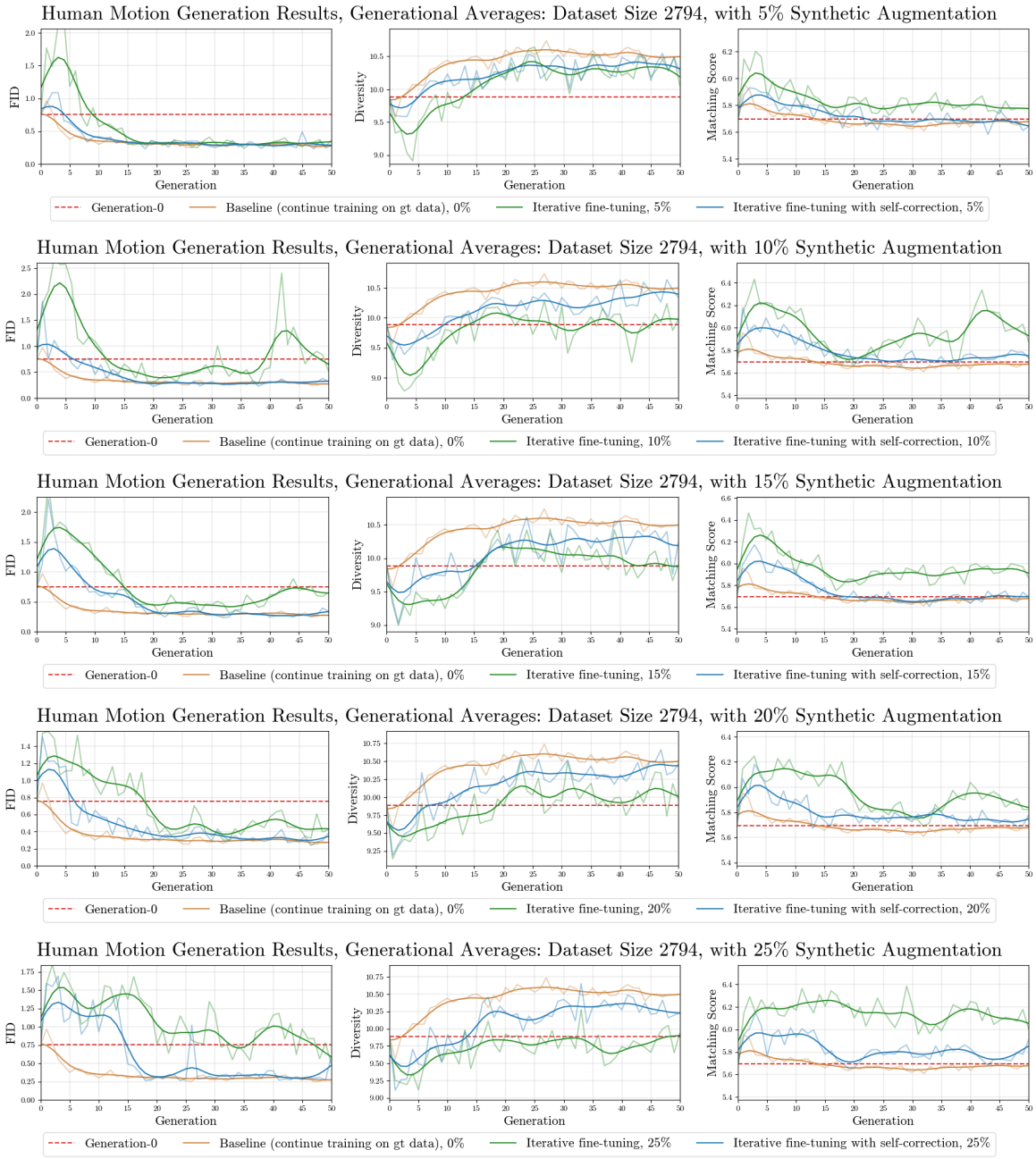


Figure 16. Results from our human motion experiments on iterative fine-tuning with dataset size $n = 2794$. These are graphs of the average evaluation metrics for every generation. Graphing the average evaluation metrics makes the training dynamics trend over time more clear. With this additional smoothing, it is more clear that iterative fine-tuning **with self-correction** outperforms iterative fine-tuning **with no self-correction**, and is competitive with the **baseline** after many generations; in fact, it appears to converge to the baseline (on average) for every synthetic augmentation percentage that we considered.

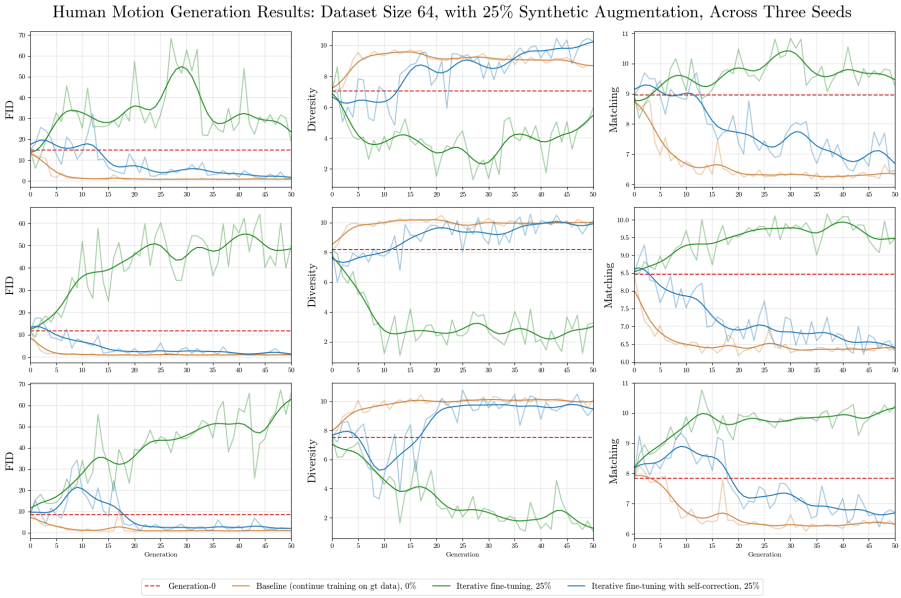


Figure 17. Results from our human motion experiments on iterative fine-tuning, with dataset size $n = 64$ and 25% augmentation percentage. Each row corresponds to a different random seed. We can see that iterative fine-tuning **with self-correction** consistently outperforms iterative fine-tuning **with no self-correction**, and the FID score appears to converge to the **baseline** after many generations.

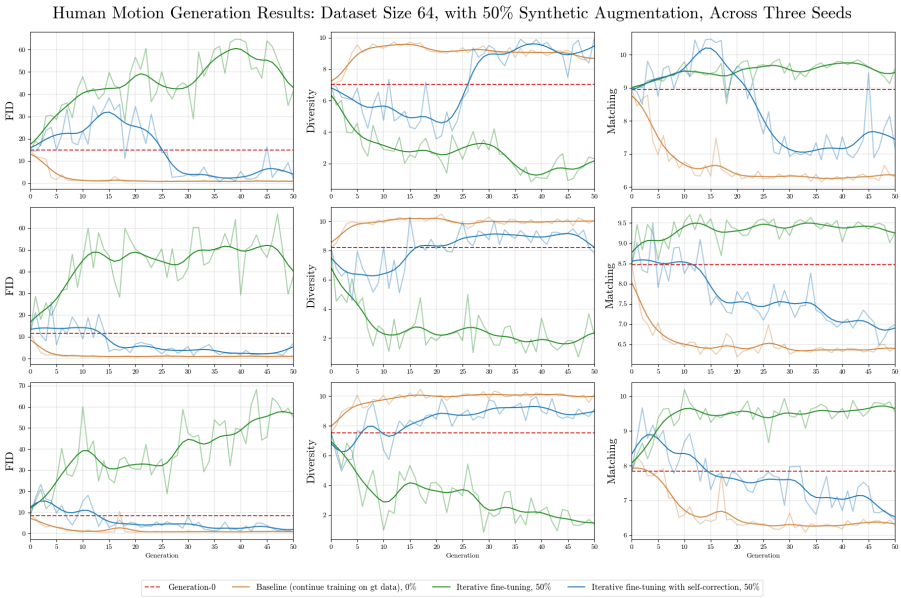


Figure 18. Results from our human motion experiments on iterative fine-tuning, with dataset size $n = 64$ and 50% augmentation percentage. Each row corresponds to a different random seed. We can see that iterative fine-tuning **with self-correction** consistently outperforms iterative fine-tuning **with no self-correction**, and the FID score appears to converge to the **baseline** after many generations.

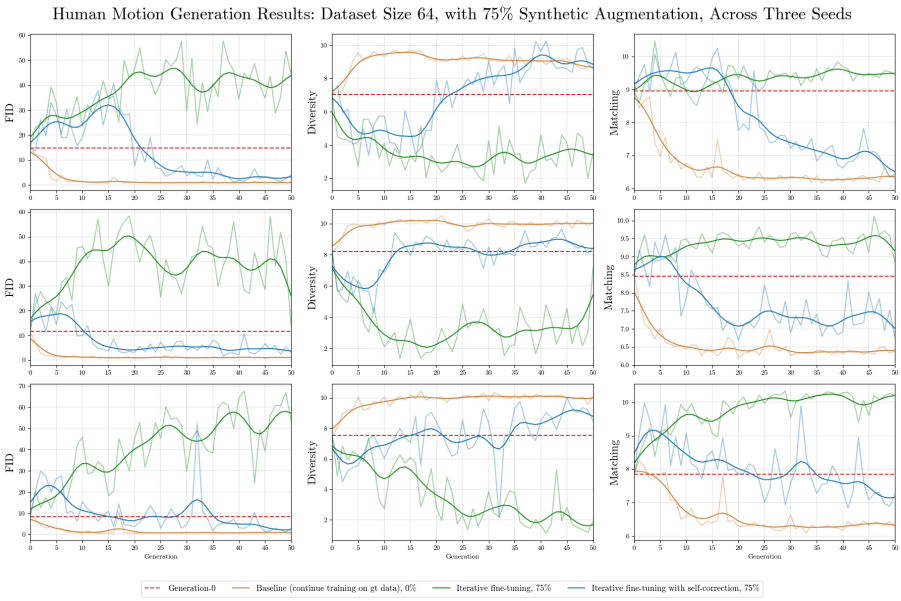


Figure 19. Results from our human motion experiments on iterative fine-tuning, with dataset size $n = 64$ and 75% augmentation percentage. Each row corresponds to a different random seed. We can see that iterative fine-tuning **with self-correction** consistently outperforms iterative fine-tuning **with no self-correction**, and the FID score appears to converge near the **baseline** after many generations.

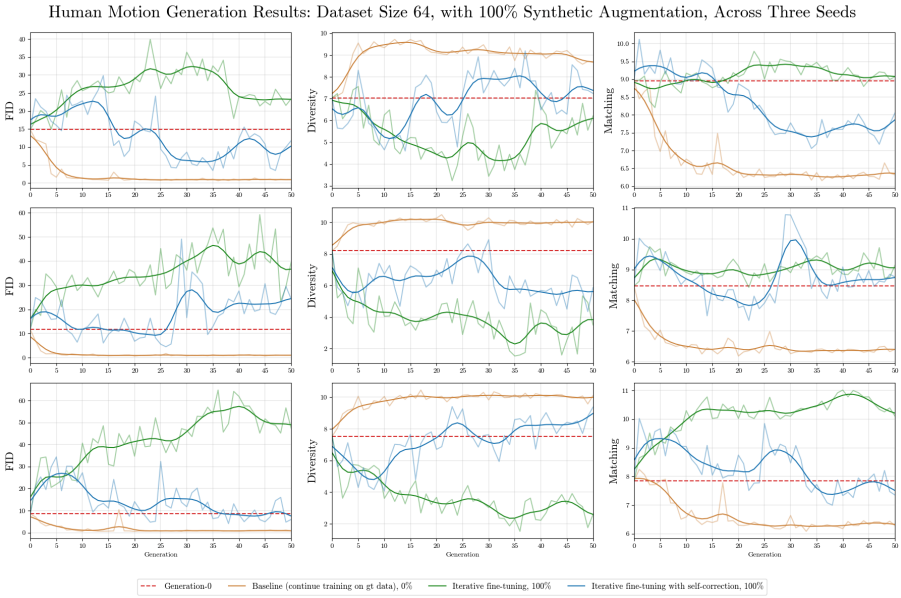


Figure 20. Results from our human motion experiments on iterative fine-tuning, with dataset size $n = 64$ and 100% augmentation percentage. Each row corresponds to a different random seed. We can see that iterative fine-tuning **with self-correction** consistently outperforms iterative fine-tuning **with no self-correction**. However, we see less stability than in the runs with a lower augmentation percentage. This is in accordance with Theorem 4.3.