GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium

Martin Heusel **Hubert Ramsauer** **Thomas Unterthiner** **Bernhard Nessler**

Günter Klambauer **Sepp Hochreiter**

LIT AI Lab & Institute of Bioinformatics,
Johannes Kepler University Linz
A-4040 Linz, Austria
{mhe,ramsauer,unterthiner,nessler,klambauer,hochreit}@bioinf.jku.at

Abstract

Generative Adversarial Networks (GANs) excel at creating realistic images with complex models for which maximum likelihood is infeasible. However, the convergence of GAN training has still not been proved. We propose a two time-scale update rule (TTUR) for training GANs with stochastic gradient descent that has an individual learning rate for both the discriminator and the generator. We prove that the TTUR converges under mild assumptions to a stationary Nash equilibrium. The convergence carries over to the popular Adam optimization, for which we prove that it follows the dynamics of a heavy ball with friction and thus prefers flat minima in the objective landscape. For the evaluation of the performance of GANs at image generation, we introduce the "Fréchet Inception Distance" (FID) which captures the similarity of generated images to real ones better than the Inception Score. In experiments, TTUR improves learning for DCGANs, improved Wasserstein GANs, and BEGANs, outperforming conventional GAN training on CelebA, One Billion Word Benchmark, and LSUN bedrooms. Implementations are available at: https://github.com/bioinf-jku/TTUR.

Introduction

Generative adversarial networks (GANs) [17] have achieved outstanding results in generating realistic images [46, 36, 27, 2, 4] and producing text [22]. GANs can learn complex generative models for which maximum likelihood or a variational approximations are infeasible. Instead of the likelihood, a discriminator network serves as objective for the generative model, that is, the generator. GAN learning is a game between the generator, which constructs synthetic data from random variables, and the discriminator, which separates synthetic data from real world data. The generator's goal is construct data in such a way that the discriminator cannot tell them apart from real world data. Thus, the discriminator tries to minimize the synthetic-real discrimination error while the generator tries to maximize this error. Since training GANs is a game and its solution is a Nash equilibrium, gradient descent may fail to converge [48, 17, 19]. For special GAN variants, convergence can be proved under certain assumptions [9, 21], as can local stability [41]. To characterize the convergence properties of training general GANs is still an open challenge [18, 19]. In an actor-critic setting, where critic learns faster than the actor, a two time-scale update rule can ensure that training reaches a stationary Nash equilibrium [45]. Convergence is proved by deriving an ordinary differential equation

(ODE), whose stable limit points coincide with stationary Nash equilibria of the underlying stochastic game. We follow the same approach.

We prove that GAN converge to a Nash equilibrium when trained by a two-time scale update rule (TTUR), i.e., when discriminator and generator have separate learning rates. This also leads to better results in practice. The main premise is that the discriminator converges to a local minimum when the generator is fixed. If the generator changes slowly enough, then the discriminator still converges, since the perturbations by the generator are small. The discriminator is tracking the generator while the former converges. Besides ensuring convergence, the results also may improve since the discriminator must first learn new patterns before they are transferred to the generator. In contrast, a generator which is overly fast, drives the discriminator steadily into new regions without having captured the generator's shortcomings. In recent GAN implementations, the discriminator often learned faster than the generator. A new objective slowed down the generator to prevent it from overtraining on the current discriminator [48]. The Wasserstein GAN algorithm uses more update steps for the discriminator than for the generator [2]. We demonstrate the learning behavior of GAN training with TTUR in contrast to standard GAN training. Figure 1 shows at the left panel a typical stochastic gradient example on MNIST for original GAN training (orig), which often leads to oscillations, and the TTUR. On the right panel an example of a 4 node network flow problem of Zhang et al. [54] is shown. The distance between the actual parameter and its optimum for an one time-scale update rule is shown across iterates. When the upper bounds on the errors are small, the iterates return to a neighborhood of the optimal solution, while for large errors the iterates may diverge (see Appendix Section A2.3).

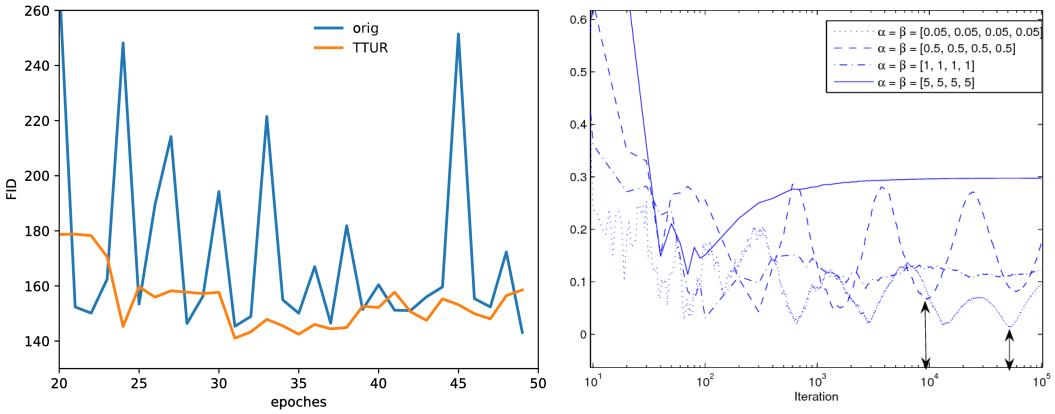

Figure 1: **Left:** Typical case of original vs. TTUR GAN training on MNIST. **Right:** Figure from Zhang 2007 [54] which shows the distance of parameter from the optimum for a one time-scale update of a 4 node network flow problem. When the upper bounds on the errors (α, β) are small, the iterates return to a neighborhood of the optimal solution (see Appendix Section A2.3). However, when the upper bounds on the errors are large, the recurrent behavior of the iterates may not occur, and the iterates may diverge.

Our novel contributions in this paper are

- the two time-scale update rule (TTUR) for GANs,
- the proof that GANs trained with TTUR converge to a stationary Nash equilibrium,
- the description of Adam as heavy ball with friction and the resulting second order differential equation,
- the convergence of GANs trained with TTUR and Adam to a stationary Nash equilibrium,
- the "Fréchet Inception Distance" (FID) to evaluate GANs, which is more consistent than the Inception Score.

Two Time-Scale Update Rule for GANs

We formulate the GAN update rules according to Goodfellow et al. [17] with discriminator $D(.; \boldsymbol{w})$ with parameter vector $\boldsymbol{w}$ and generator $G(.; \boldsymbol{\theta})$ with parameter vector $\boldsymbol{\theta}$. First we define the gradient $\tilde{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w})$ of discriminator's loss function and the gradient $\tilde{\boldsymbol{h}}(\boldsymbol{\theta}, \boldsymbol{w})$ of the generator's loss function:

$$\tilde{\boldsymbol{g}}\big(\boldsymbol{\theta}, \boldsymbol{w}\big) \;=\; \nabla_w \left[\frac{1}{m} \sum_{i=1}^{m} \left(\log D\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right) \;+\; \log\left(1 \;-\; D\left(G\left(\boldsymbol{z}^{(i)}; \boldsymbol{\theta}\right); \boldsymbol{w}\right)\right)\right)\right] , \quad (1)$$

$$\tilde{\boldsymbol{h}}\big(\boldsymbol{\theta}, \boldsymbol{w}\big) \;= \nabla_\theta \left[- \frac{1}{m} \sum_{i=1}^{m} \log\left(1 \;-\; D\left(G\left(\boldsymbol{z}^{(i)}; \boldsymbol{\theta}\right); \boldsymbol{w}\right)\right)\right] . \quad (2)$$

In the generator gradient formula Eq. (2), $\log(1 - D(.))$ is often replaced by $-\log D(.)$ to speed up learning at the beginning [19]. The Wasserstein GAN has the same gradients but without "log" and the resulting constants [2].

The gradients Eq. (1) and Eq. (2) use mini-batches of m real world samples $\boldsymbol{x}^{(i)}, 1 \leqslant i \leqslant m$ and m synthetic samples $\boldsymbol{z}^{(i)}, 1 \leqslant i \leqslant m$ which are randomly chosen. Therefore both gradients are stochastic. If the true gradients are $\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{w})$ and $\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{w})$, then we can define $\tilde{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}) = \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{w}) + \boldsymbol{M}^{(w)}$ and $\tilde{\boldsymbol{h}}(\boldsymbol{\theta}, \boldsymbol{w}) = \boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{w}) + \boldsymbol{M}^{(\theta)}$ with random variables $\boldsymbol{M}^{(w)}$ and $\boldsymbol{M}^{(\theta)}$. Thus, the update rules Eq. (1) and Eq. (2) are stochastic approximations algorithms. Recently GANs have been analyzed using stochastic approximation algorithms [41]. We analyze GANs as two time-scale stochastic approximations algorithms. For a two time-scale update rule (TTUR), we use the learning rates $b(n)$ and $a(n)$ for the discriminator and the generator update, respectively:

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n) \left(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{M}_n^{(w)}\right), \; \boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n) \left(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{M}_n^{(\theta)}\right) . \quad (3)$$

For more details on the following convergence proof and its assumptions, see Appendix Section A2.1. To prove convergence of GANs learned by TTUR, we make the following assumptions (The actual assumption is ended by ◀, the following text are just comments and explanations):

- **(A1)** The gradients $\boldsymbol{h}$ and $\boldsymbol{g}$ are Lipschitz. ◀ Consequently, networks with Lipschitz smooth activation functions like ELUs ($\alpha = 1$) [12] fulfill the assumption but not ReLU networks.
- **(A2)** $\sum_n a(n) = \infty$, $\sum_n a^2(n) < \infty$, $\sum_n b(n) = \infty$, $\sum_n b^2(n) < \infty$, $a(n) \;=\; \mathrm{o}(b(n))$◀
- **(A3)** The stochastic gradient errors $\{\boldsymbol{M}_n^{(\theta)}\}$ and $\{\boldsymbol{M}_n^{(w)}\}$ are martingale difference sequences w.r.t. the increasing σ-field $\mathcal{F}_n \;=\; \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, l \;\leqslant\; n), n \;\geqslant\; 0$ with $\mathrm{E}\left[\|\boldsymbol{M}_n^{(\theta)}\|^2 \mid \mathcal{F}_n^{(\theta)}\right] \leqslant B_1$ and $\mathrm{E}\left[\|\boldsymbol{M}_n^{(w)}\|^2 \mid \mathcal{F}_n^{(w)}\right] \leqslant B_2$, where B_1 and B_2 are positive deterministic constants.◀ The original Assumption (A3) from Borkar 1997 follows from Lemma 2 in [5] (see also [47]). The assumption is fulfilled in the Robbins-Monro setting, where mini-batches are randomly sampled and the gradients are bounded.
- **(A4)** For each $\boldsymbol{\theta}$, the ODE $\dot{\boldsymbol{w}}(t) \;=\; \boldsymbol{g}\big(\boldsymbol{\theta}, \boldsymbol{w}(t)\big)$ has a local asymptotically stable attractor $\boldsymbol{\lambda}(\boldsymbol{\theta})$ within a domain of attraction G_θ such that $\boldsymbol{\lambda}$ is Lipschitz. The ODE $\dot{\boldsymbol{\theta}}(t) \;=\; \boldsymbol{h}\big(\boldsymbol{\theta}(t), \boldsymbol{\lambda}(\boldsymbol{\theta}(t))\big)$ has a local asymptotically stable attractor $\boldsymbol{\theta}^*$ within a domain of attraction.◀ The discriminator must converge to a minimum for fixed generator parameters and the generator, in turn, must converge to a minimum for this fixed discriminator minimum. Borkar 1997 required unique global asymptotically stable equilibria [7]. The assumption of global attractors was relaxed to local attractors via Assumption (A6)' and Theorem 2.7 in Karmakar & Bhatnagar [28]. See for more details Assumption (A6) in Appendix Section A2.1.3. Here, the GAN objectives may serve as Lyapunov functions. Recently it has been shown that the traditional GAN formulation is locally asymptotically stable [41].
- **(A5)** $\sup_n \|\boldsymbol{\theta}_n\| < \infty$ and $\sup_n \|\boldsymbol{w}_n\| < \infty$.◀ Typically ensured by objective or weight decay. The parameters can be projected to a box which leads to a projected stochastic approximation. Theorem 5.3.1 on page 191 of Kushner & Clark [35] states convergence for projected stochastic approximations for single iterates (see Appendix E of Bhatnagar, Prasad, & Prashanth 2013 [6]).

The next theorem has been proved in the seminal paper of Borkar 1997 [7].

Theorem 1 (Borkar)**.** *If the assumptions are satisfied, then the updates Eq.* (3) *converge to* $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*))$ *a.s.*

The solution $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*))$ is a stationary Nash equilibrium [45], since $\boldsymbol{\theta}^*$ as well as $\boldsymbol{\lambda}(\boldsymbol{\theta}^*)$ are local asymptotically stable attractors with $\boldsymbol{g}\big(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*)\big) = \mathbf{0}$ and $\boldsymbol{h}\big(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*)\big) = \mathbf{0}$. An alternative approach to the proof of convergence using the Poisson equation for ensuring a solution to the fast update rule can be found in Appendix Section A2.1.2. This approach assumes a linear update function in the fast update rule which, however, can be a linear approximation to a nonlinear gradient [31, 33]. For the rate of convergence see Appendix Section A2.2, where Section A2.2.1 focuses on linear and Section A2.2.2 on non-linear updates. Recently the local stability of GANs and their convergence rates have been investigated [41].

For equal time scales it can only be proved that the updates revisit an environment of the solution infinitely often, which, however, can be very large [54, 13]. For more details on the analysis of equal time scales see Appendix Section A2.3. The main idea in the proof of Borkar [7] is to use (T, δ) perturbed ODEs according to Hirsch 1989 [23] (see also Appendix C of Bhatnagar, Prasad, & Prashanth 2013 [6]). The proof relies on the fact that there eventually is a time point when the perturbation of the slow update rule is small enough (given by δ) to allow the fast update rule to converge. For experiments with TTUR, we aim at finding learning rates such that the slow update is small enough to allow the fast to converge. Typically, the slow update is the generator and the fast update the discriminator. We have to adjust the two learning rates such that the generator does not affect discriminator learning in a undesired way and perturb it too much. However, even a larger learning rate for the generator than for the discriminator may ensure that the discriminator has low perturbations. Learning rates cannot be translated into perturbation since the perturbation of discriminator by the generator is different from the perturbation of the generator by the discriminator.

Two Time-Scale Update Rule for Adam

GANs suffer from "mode collapse", where large masses of probability are mapped onto a few modes that cover only small regions of the entire data distribution. While these regions represent meaningful samples, the variety of the real world data is lost and only a few prototype samples are generated. Different methods have been proposed to avoid mode collapsing [10, 39]. We observed that models trained with Adam [30] are less prone to mode collapse than pure SGD. We hypothesize that Adam reduces the risk of mode collapse because it follows the dynamics of a Heavy Ball with Friction (HBF) (see below). The HBF dynamics stem from the averages over past gradients. This averaging corresponds to a velocity that makes the generator resistant to getting pushed into small regions. Adam as an HBF method typically overshoots small local minima that correspond to model collapse and can find flat minima which generalize well [25]. Figure 2 depicts the dynamics of HBF, where the ball settles at a flat minimum. Next, we analyze whether GANs trained with TTUR converge when using Adam. For more details see Appendix Section A3.

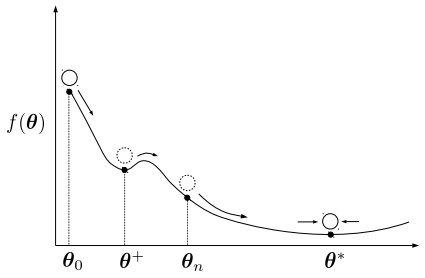

Figure 2: Heavy Ball with Friction, where the ball with mass overshoots the local minimum $\boldsymbol{\theta}^+$ and settles at the flat minimum $\boldsymbol{\theta}^*$.

We recapitulate the Adam update rule at step n, with learning rate a, exponential averaging factors β_1 for the first and β_2 for the second moment of the gradient $\nabla f(\boldsymbol{\theta}_{n-1})$:

$$
\begin{aligned}
\boldsymbol{g}_n &\longleftarrow \nabla f(\boldsymbol{\theta}_{n-1}) \\
\boldsymbol{m}_n &\longleftarrow (\beta_1/(1-\beta_1^n))\, \boldsymbol{m}_{n-1} \,+\, ((1-\beta_1)/(1-\beta_1^n))\, \boldsymbol{g}_n \\
\boldsymbol{v}_n &\longleftarrow (\beta_2/(1-\beta_2^n))\, \boldsymbol{v}_{n-1} \,+\, ((1-\beta_2)/(1-\beta_2^n))\, \boldsymbol{g}_n \odot \boldsymbol{g}_n \\
\boldsymbol{\theta}_n &\longleftarrow \boldsymbol{\theta}_{n-1} \,-\, a\, \boldsymbol{m}_n/(\sqrt{\boldsymbol{v}_n} + \epsilon)\,,
\end{aligned}
\tag{4}
$$

where following operations are meant componentwise: the product $\odot$, the square root $\sqrt{.}$, and the division $/$ in the last line.

Instead of learning rate a, we introduce the damping coefficient $a(n)$ with $a(n) = an^{-\tau}$ for $\tau \in (0, 1]$. Adam has parameters β_1 for averaging the gradient and β_2 for averaging the squared gradient. These parameters can be considered as defining a memory for Adam. To characterize β_1 and β_2 in the following, we define the exponential memory $r(n) = r$ and the polynomial memory $r(n) = r/\sum_{l=1}^{n} a(l)$ for some positive constant r. The next theorem describes Adam by a differential equation, which in turn allows to apply the idea of (T, δ) perturbed ODEs to TTUR. Consequently, learning GANs with TTUR and Adam converges.

Theorem 2. *If Adam is used with $\beta_1 = 1 - a(n+1)r(n)$, $\beta_2 = 1 - \alpha a(n+1)r(n)$ and with ∇f as the full gradient of the lower bounded, continuously differentiable objective f, then for stationary second moments of the gradient, Adam follows the differential equation for Heavy Ball with Friction (HBF):*

$$\ddot{\boldsymbol{\theta}}_t \ + \ a(t)\, \dot{\boldsymbol{\theta}}_t \ + \ \nabla f(\boldsymbol{\theta}_t) \ = \ \mathbf{0} \, . \tag{5}$$

Adam converges for gradients ∇f that are L-Lipschitz.

Proof. Gadat et al. derived a discrete and stochastic version of Polyak's Heavy Ball method [44], the Heavy Ball with Friction (HBF) [16]:

$$\begin{aligned} \boldsymbol{\theta}_{n+1} \ &= \ \boldsymbol{\theta}_n \ - \ a(n+1)\, \boldsymbol{m}_n \, , \\ \boldsymbol{m}_{n+1} \ &= \ \left(1 \ - \ a(n+1)\, r(n)\right) \boldsymbol{m}_n \ + \ a(n+1)\, r(n) \, \left(\nabla f(\boldsymbol{\theta}_n) \ + \ \boldsymbol{M}_{n+1}\right) . \end{aligned} \tag{6}$$

These update rules are the first moment update rules of Adam [30]. The HBF can be formulated as the differential equation Eq. (5) [16]. Gadat et al. showed that the update rules Eq. (6) converge for loss functions f with at most quadratic grow and stated that convergence can be proofed for ∇f that are L-Lipschitz [16]. Convergence has been proved for continuously differentiable f that is quasiconvex (Theorem 3 in Goudou & Munier [20]). Convergence has been proved for ∇f that is L-Lipschitz and bounded from below (Theorem 3.1 in Attouch et al. [3]).

Adam normalizes the average $\boldsymbol{m}_n$ by the second moments $\boldsymbol{v}_n$ of of the gradient $\boldsymbol{g}_n$: $\boldsymbol{v}_n = \mathrm{E}\left[\boldsymbol{g}_n \odot \boldsymbol{g}_n\right]$. $\boldsymbol{m}_n$ is componentwise divided by the square root of the components of $\boldsymbol{v}_n$. We assume that the second moments of $\boldsymbol{g}_n$ are stationary, i.e., $\boldsymbol{v} = \mathrm{E}\left[\boldsymbol{g}_n \odot \boldsymbol{g}_n\right]$. In this case the normalization can be considered as additional noise since the normalization factor randomly deviates from its mean. In the HBF interpretation the normalization by $\sqrt{\boldsymbol{v}}$ corresponds to introducing gravitation. We obtain

$$\boldsymbol{v}_n \ = \ \frac{1 \ - \ \beta_2}{1 \ - \ \beta_2^n} \ \sum_{l=1}^{n} \beta_2^{n-l} \ \boldsymbol{g}_l \odot \boldsymbol{g}_l \ , \quad \Delta \boldsymbol{v}_n \ = \ \boldsymbol{v}_n \ - \ \boldsymbol{v} \ = \ \frac{1 \ - \ \beta_2}{1 \ - \ \beta_2^n} \ \sum_{l=1}^{n} \beta_2^{n-l} \ \left(\boldsymbol{g}_l \odot \boldsymbol{g}_l \ - \ \boldsymbol{v}\right) \, . \tag{7}$$

For a stationary second moment $\boldsymbol{v}$ and $\beta_2 = 1 - \alpha a(n+1)r(n)$, we have $\Delta \boldsymbol{v}_n \propto a(n+1)r(n)$. We use a componentwise linear approximation to Adam's second moment normalization $1/\sqrt{\boldsymbol{v} + \Delta \boldsymbol{v}_n} \approx 1/\sqrt{\boldsymbol{v}} - (1/(2\boldsymbol{v} \odot \sqrt{\boldsymbol{v}})) \odot \Delta \boldsymbol{v}_n + \mathrm{O}(\Delta^2 \boldsymbol{v}_n)$, where all operations are meant componentwise. If we set $\boldsymbol{M}_{n+1}^{(v)} = -(\boldsymbol{m}_n \odot \Delta \boldsymbol{v}_n)/(2\boldsymbol{v} \odot \sqrt{\boldsymbol{v}} a(n+1)r(n))$, then $\boldsymbol{m}_n/\sqrt{\boldsymbol{v}_n} \approx \boldsymbol{m}_n/\sqrt{\boldsymbol{v}} + a(n+1)r(n)\boldsymbol{M}_{n+1}^{(v)}$ and $\mathrm{E}\left[\boldsymbol{M}_{n+1}^{(v)}\right] = \mathbf{0}$, since $\mathrm{E}\left[\boldsymbol{g}_l \odot \boldsymbol{g}_l - \boldsymbol{v}\right] = \mathbf{0}$. For a stationary second moment $\boldsymbol{v}$, the random variable $\{\boldsymbol{M}_n^{(v)}\}$ is a martingale difference sequence with a bounded second moment. Therefore $\{\boldsymbol{M}_{n+1}^{(v)}\}$ can be subsumed into $\{\boldsymbol{M}_{n+1}\}$ in update rules Eq. (6). The factor $1/\sqrt{\boldsymbol{v}}$ can be componentwise incorporated into the gradient $\boldsymbol{g}$ which corresponds to rescaling the parameters without changing the minimum. □

According to Attouch et al. [3] the energy, that is, a Lyapunov function, is $E(t) = 1/2|\dot{\boldsymbol{\theta}}(t)|^2 + f(\boldsymbol{\theta}(t))$ and $\dot{E}(t) = -a\ |\dot{\boldsymbol{\theta}}(t)|^2 < 0$. Since Adam can be expressed as differential equation and has a Lyapunov function, the idea of (T, δ) perturbed ODEs [7, 23, 8] carries over to Adam. Therefore the convergence of Adam with TTUR can be proved via two time-scale stochastic approximation analysis like in Borkar [7] for stationary second moments of the gradient.

Experiments

Performance Measure

Before presenting the experiments, we introduce a quality measure for models learned by GANs. The objective of generative learning is that the model produces data which matches the observed data. Therefore, each distance between the probability of observing real world data $p_w(.)$ and the probability of generating model data $p(.)$ can serve as performance measure for generative models. However, defining appropriate performance measures for generative models is difficult [50]. The best known measure is the likelihood, which can be estimated by annealed importance sampling [52]. However, the likelihood heavily depends on the noise assumptions for the real data and can be dominated by single samples [50]. Other approaches like density estimates have drawbacks as well [50].

A well-performing approach to measure the performance of GANs is the "Inception Score" which correlates with human judgment [48]. Generated samples are fed into a Inception model that was trained on ImageNet. Images with meaningful objects are supposed to have low label (output) entropy, that is, they belong to few object classes. On the other hand, the entropy across images should be high, that is, the variance over the images should be large. A drawback of the Inception Score is that the statistics of real world samples are not used and compared to the statistics of synthetic samples. Therefore, we improve the Inception Score. The equality $p(.) = p_w(.)$ holds except for a non-measurable set if and only if $\int p(.)f(x)dx = \int p_w(.)f(x)dx$ for a basis $f(.)$ spanning the function space in which $p(.)$ and $p_w(.)$ live. These equalities of expectations are used to describe distributions by moments or cumulants, where $f(x)$ are polynoms of the data x. We generalize these polynoms by replacing x by the coding layer of an Inception model in order to obtain vision-relevant features. For practical reasons we only consider the first two polynoms, that is, the first two moments: mean and covariance. The Gaussian is the maximum entropy distribution for given mean and covariance, therefore we assume the coding units to follow a multidimensional Gaussian. The difference of two Gaussians (synthetic and real-world images) is measured by the Fréchet distance [15] also known as Wasserstein-2 distance [51]. We call the Fréchet distance $d(.,.)$ between the Gaussian with mean and covariance $(\boldsymbol{m}, \boldsymbol{C})$ obtained from $p(.)$ and the Gaussian $(\boldsymbol{m}_w, \boldsymbol{C}_w)$ obtained from $p_w(.)$ the "Fréchet Inception Distance" (FID), which is given by [14]:

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\big(\boldsymbol{C} + \boldsymbol{C}_w - 2\big(\boldsymbol{C}\boldsymbol{C}_w\big)^{1/2}\big). \tag{8}$$

Next we show that the FID is consistent with human judgment and increasing disturbances. Figure 3 displays the evaluation of the FID for Gaussian noise, Gaussian blur, implanted black rectangles, swirled images, salt and pepper noise, and CelebA dataset contaminated by ImageNet images. The FID captures the disturbance level well. Therefore, we used the FID to evaluate the performance of GANs in the experiments. For more details and a comparison between FID and Inception Score see Appendix Section A1, where we show that FID is more consistent with the noise level than the Inception Score.

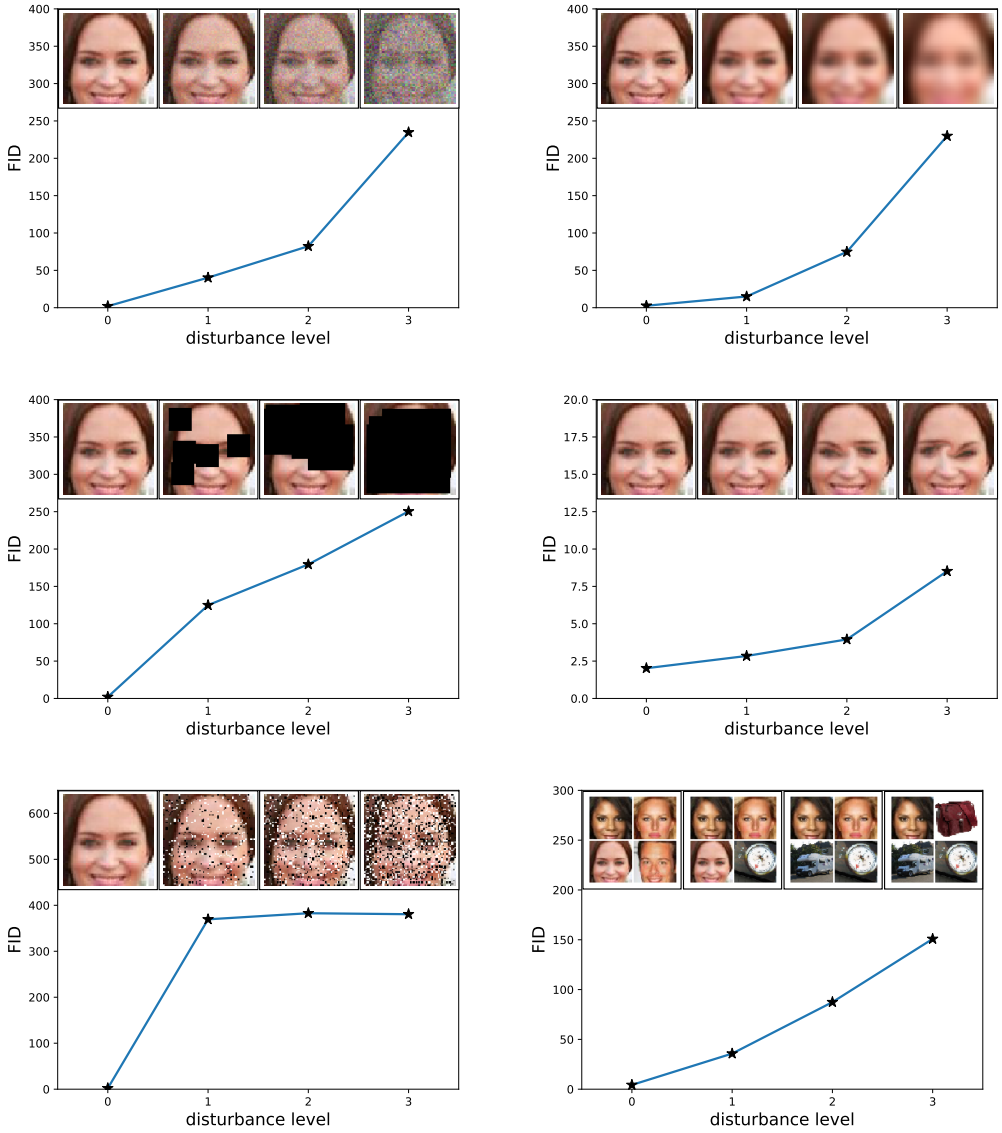

Figure 3: FID is evaluated for Gaussian noise (upper left), Gaussian blur (upper right), implanted black rectangles (middle left), swirled images (middle right), salt and pepper noise (lower left), and CelebA dataset contaminated by ImageNet images (lower right). Left is the smallest disturbance level of zero, which increases to the highest level at right. The FID captures the disturbance level very well by monotonically increasing.

Model Selection and Evaluation

We have selected Adam stochastic optimization to reduce the risk of mode collapsing. The advantage of Adam has been confirmed by MNIST experiments, where Adam indeed considerably reduced the cases for which we observed mode collapsing. Although TTUR ensures that the discriminator converges during learning, practicable learning rates must be found for each experiment. We face a trade-off since the learning rates should be small enough (e.g. for the generator) to ensure convergence but at the same time should be large enough to allow fast learning. For each of the experiments, the learning rates have been optimized to be large while still ensuring stable training which is indicated by a decreasing FID or Jensen-Shannon-divergence. We further fixed the time point for stopping training to the update step when the FID or Jensen-Shannon-divergence of the best models was no longer decreasing. For some models, we observed that the FID diverges or starts to increase at a certain time point. An example of this behaviour is shown in Figure 4.

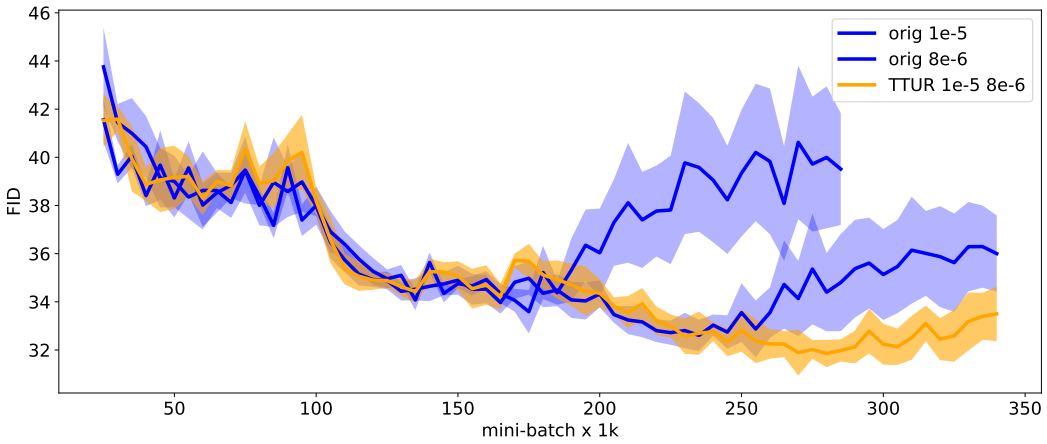

Figure 4: The FID for BEGAN on CelebA along training time given as 1k mini-batch updates. The FID of the GANs diverges at different time points, where TTUR training diverges later than the original GANs. TTUR has lower variance of its FID compared to the original training method, which indicates more stable learning.

The performance of generative models is evaluated via the Fréchet Inception Distance (FID) introduced above. For the Billon Word experiment, the normalized Jensen-Shannon-divergence served as performance measure.

For computing the FID, we propagated 100,000 random images from the CelebA or LSUN bedroom dataset through the pretrained Inception-v3 model (for the pretrained Inception model see Section A6). Following the computation of the Inception Score [48], we use the `pool_3:0` tensor as coding layer. For this coding layer, we calculated the mean $\boldsymbol{m}_w$ and the covariance matrix $\boldsymbol{C}_w$ for the 100,000 images. Thus, we approximate the first and second central moment of the function given by the Inception coding layer under the real world distribution. To approximate these moments also for the model distribution, we generate 5,000 images from the model, propagate them through the Inception-v3 model, and compute mean $\boldsymbol{m}$ and the covariance matrix $\boldsymbol{C}$ for these 5,000 images. For computational efficiency, we only evaluate the FID every 5,000 mini-batch updates.

For more details, used implementations and further results Appendix Section A4.

Results

CelebA. The Large-scale CelebFaces Attributes (CelebA) dataset [37] has been used to evaluate GANs [24]. We used images that were center cropped to 64×64 pixels. We trained Boundary Equilibrium GANs (BEGAN) [4] with their original training procedure and with TTUR. Figure 5 (left panel) shows training performance across mini-batch updates. We report the average FID and

standard deviation for 8 runs for TTUR and the training procedure every 5,000 mini-batches. Figure 5 (right panel) shows the FID at the end of the training. TTUR outperformed the original training starting from mini-batch update around 100k. The best FIDs that could be obtained with the original BEGANs and TTUR trained BEGANs are 28.55 and 26.19, respectively (see Table 1). Examples of CelebA images generated by BEGAN trained with the original procedure and TTUR are given in Figure 6 for a FID around 48 and in Figure 7 for a FID of 26.

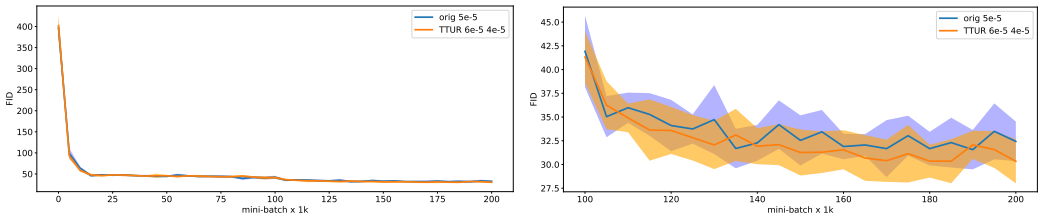

Figure 5: FID for BEGAN trained on CelebA. The learning rate a for the original training method is given as "orig a". TTUR learning rates are given as pairs (b, a) of discriminator learning rate b and generator learning rate a: "TTUR b a". The shaded region indicates the standard deviation for 8 runs for each experiment. **Left:** Training curves for the whole training from 0 to 200k updates. **Right:** Training curves zoomed in on the region from 100k to 200k updates. At the end of the learning process, BEGAN trained with TTUR exhibit an improved FID.

Table 1: The performance of BEGAN trained with the original procedure and with TTUR on CelebA. We compare the networks performance with respect to the FID at the optimal number of updates during training. BEGAN trained with TTUR exhibit a better FID.

method	learning rates	updates	FID
BEGAN TTUR	6e-5, 4e-5	175,000	26.19
BEGAN orig	5e-5	170,000	28.55

Figure 6: Examples of CelebA images generated by BEGANs with a FID around 48 trained with the original training method (top two rows) and TTUR (lower two rows).

In the next experiment, we test TTUR for the deep convolutional GAN (DCGAN) [46] at the CelebA dataset. Figure 8 shows the FID during learning DCGAN with the original learning method and with TTUR, each averaged over five runs and surrounded with one standard deviation. The original

Figure 7: Examples of CelebA images generated by BEGANs with a FID of 28 trained with the original training method (top two rows) and FID 26 with TTUR (lower two rows).

training method is faster at the beginning, but TTUR finally achieves better performance. Figure 8 (right panel) zooms in at the region of 10k to 50k mini-batch updates of Figure 8 (left panel) to show the difference between TTUR and original learning of DCGANs. Overall, DCGAN achieves a lower FID than BEGAN, which we attribute to higher variety of the generated images. DCGAN trained TTUR reaches a lower FID than the original method. The best FIDs that could be obtained with original DCGANs and TTUR trained DCGANs over all runs are 20.09 and 17.88, respectively (see Table 2). Note, that in this case the learning rate of the generator is larger then that of the discriminator. This is not contradictory to the theory, for more details see A5. Examples of CelebA images generated by DCGAN trained with the original procedure and TTUR are given in Figure 9.

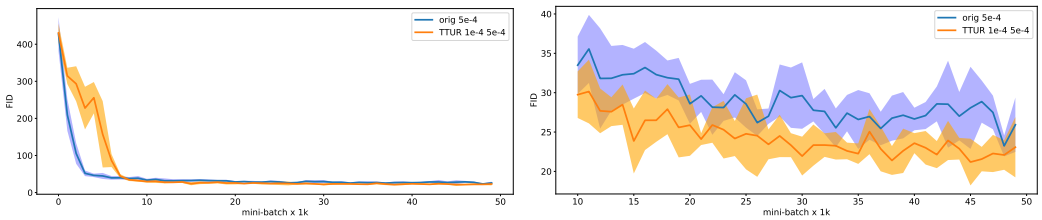

Figure 8: FID for DCGAN trained on CelebA. TTUR learning rates are given as pairs (b, a) of discriminator learning rate b and generator learning rate a: "TTUR b a". **Left:** Training curves for the whole training from 0 to 50k updates. **Right:** Training curves zoomed in on the region from 10k to 50k updates. While the original training procedure is faster at the beginning, training with TTUR leads to a better FID.

Table 2: The performance of DCGAN trained with the original procedure and with TTUR on CelebA. We compare the networks performance with respect to the FID at the optimal number of updates during training. DCGAN trained with TTUR exhibit a lower FID.

method	learning rates	updates	FID
DCGAN TTUR	1e-4, 1e-3	45,000	17.88
DCGAN orig	5e-4	49,000	20.09

Figure 9: CelebA samples generated by DCGANs with TTUR (left) and original procedure (right) respectively.

One Billion Word. The One Billion Word Benchmark [11] serves to compare TTUR to other GAN training methods. We used the Improved WGAN model [22]. The character-level generative language model is a 1D convolutional neural network (CNN) which maps a latent vector into a sequence of one-hot character vectors of dimension 32 given by the maximum of a softmax output. The discriminator is also a 1D CNN applied to sequences of one-hot vectors of 32 characters. Since the FID criterium only works for images, we measured the performance by the Jensen-Shannon-divergence (JSD) between the model and the real world distribution as has been done previously [22]. In contrast to the original code in which the critic is trained 10 times for each generator update, TTUR updates the discriminator only once, therefore we align the training progress with wall-clock time. The learning rate for the original training was optimized to be large but leads to stable learning. TTUR can use a higher learning rate for the discriminator since TTUR stabilizes learning. We report the normalized mean JSD for 10 runs for original training and TTUR training in Figure 10 for 4-gram and 6-gram word evaluation. TTUR outperforms the standard training procedure for both evaluation measures. The improvement of TTUR on the 6-gram statistics over original training shows that TTUR enables to learn to generate more subtle pseudo-words which better resembles real words. The best JSD that could be obtained with original and TTUR trained Improved WGANs over all runs for the 4-gram evaluation are 0.374 and 0.351, respectively and 0.756 and 0.736 for the 6-gram evaluation (see Table 3). Training Improved WGANs with TTUR improved their performance with respect to both 4-gram and 6-gram JSD. In Table 4 we show randomly chosen samples from models trained with original method and TTUR.

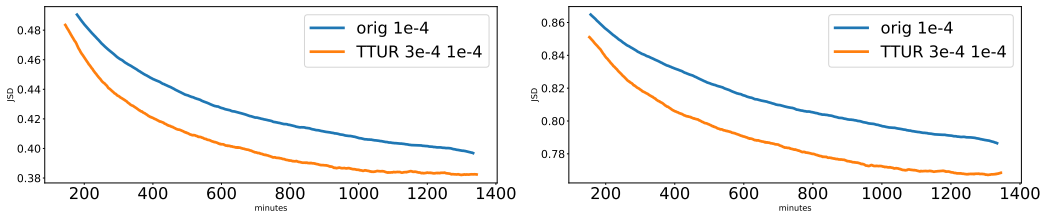

Figure 10: Performance of improved WGAN models trained with the original (orig) and our TTUR method on the Billion Word benchmark. The performance is measured by the normalized Jensen-Shannon-divergence based on 4-grams (left) and 6-grams (right) and averaged over 10 runs. TTUR learning clearly improved the original learning which is more prominent at 6-gram than at 4-gram.

Table 3: Performance of Improved WGAN trained with the original procedure and with TTUR on the One Billion Word Benchmark. We compare the networks performance with respect to JSD score at the optimal number of updates during training. Improved WGAN trained with TTUR exhibit a better FID.

	4-gram			6-gram		
method	learning rates	wall-clock minutes	JSD	learning rates	wall-clock minutes	JSD
Improved WGAN TTUR	3e-4, 1e-4	1300	0.351	3e-4, 1e-4	1400	0.736
Improved WGAN orig	1e-4	1380	0.374	1e-4	1400	0.756

Table 4: Samples of Billion Word data generated by improved WGAN trained with TTUR (left) the original method (right).

```
Dry Hall Sitning tven the concer
There are court phinchs hasffort
He scores a supponied foutver il
Bartfol reportings ane the depor
Seu hid , it 's watter 's remold
Later fasted the store the inste
Indiwezal deducated belenseous K
Starfers on Rbama 's all is lead
Inverdick oper , caldawho 's non
She said , five by theically rec
RichI , Learly said remain .‘‘‘‘
Reforded live for they were like
The plane was git finally fuels
The skip lifely will neek by the
SEW McHardy Berfect was luadingu
But I pol rated Franclezt is the
```

```
No say that tent Franstal at Bra
Caulh Paphionars tven got corfle
Resumaly , braaky facting he at
On toipe also houd , aid of sole
When Barrysels commono toprel to
The Moster suprr tent Elay diccu
The new vebators are demases to
Many 's lore wockerssaow 2 2 ) A
Andly , has le wordd Uold steali
But be the firmoters is no 200 s
Jermueciored a noval wan 't mar
Onles that his boud-park , the g
ISLUN , The crather wilh a them
Fow 22o2 surgeedeto , theirestra
Make Sebages of intarmamates , a
Gullla " has cautaria Thoug ly t
```

LSUN Bedrooms. We compare TTUR to the original GAN training for BEGANs [4] on the bedrooms category of the large scale image database (LSUN) [53]. Figure 11 shows the training curves of BEGAN trained with the original procedure and with TTUR and original runs. BEGAN trained with TTUR show a lower FID from around 40k mini-batch updates and maintains a better performance until the end of the training. However, the best FID that could be obtained with original BEGANs and TTUR trained BEGANs over all runs are similar with 112.8 and 112.0, respectively (see Table 5). Figure 12 shows examples of samples of BEGAN trained with the original method and with TTUR.

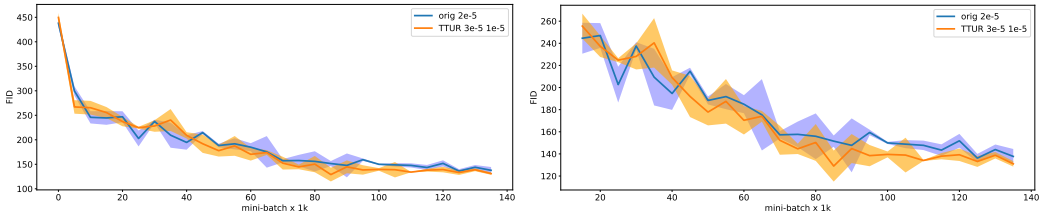

Figure 11: FID of BEGANs on the LSUN bedroom dataset trained with the original method (orig) and TTUR. TTUR learning rates are given as pairs (b, a) of discriminator learning rate b and generator learning rate a: "TTUR b a". The curves are averages over 3 runs, the shaded region indicates the standard deviation. **Left:** Training curves for the whole training from 0 to 140k updates. **Right:** Training curves zoomed in on the region from 20k to 140k updates.

Table 5: The performance of BEGAN trained with the original procedure and with TTUR on LSUN. We compare the networks performance with respect to the FID at the optimal number of updates during training. BEGAN trained with TTUR exhibit a similar FID as with the original training.

method	learning rates	updates	FID
BEGAN TTUR	3e-5, 1e-5	85,000	112.0
BEGAN orig	2e-5	96,000	112.8

Figure 12: LSUN bedroom samples generated by BEGANs trained with the original method (left) and the TTUR method (right).

We compare TTUR to the original GAN training procedure for DCGANs at the LSUN dataset. Figure 13 shows the training curves of DCGAN with the original learning method and with TTUR. The original training method improves the FID faster at the beginning, but TTUR finally achieves better performance. The right plot in Figure 13 zooms in at the region of 10k to 50k mini-batch updates of Figure 13 to show the difference between TTUR and original learning of DCGANs. DCGAN achieves a lower FID than BEGAN, therefore gives better results, which we attribute to higher variety of the generated images. DCGAN trained with TTUR reaches a lower FID than the original method. The best FIDs that could be obtained with original DCGANs and TTUR trained DCGANs are 69.7 and 68.3, respectively (see Table 6). Similar to TTUR DCGAN training on CelebA, the learning rate of the generator is larger than that of the discriminator. To see that this is not contradictory to the theory, we again refer to A5. Figure 14 shows examples of samples of DCGAN trained with the original method and with TTUR.

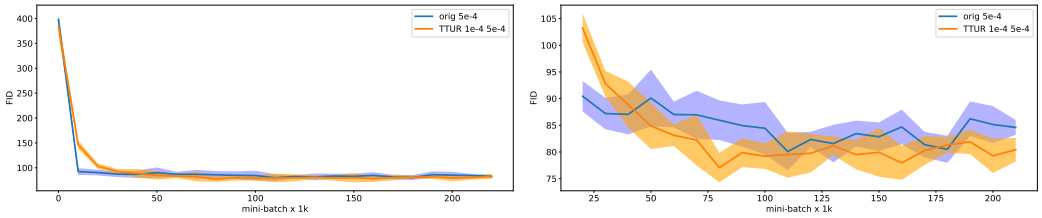

Figure 13: FID of DCGANs on the LSUN bedroom dataset trained with the original method (orig) and TTUR. TTUR learning rates are given as pairs (b, a) of discriminator learning rate b and generator learning rate a: "TTUR b a". The curves are averages over 5 runs. **Left:** Training curves for the whole training from 0 to 200k updates. **Right:** Training curves zoomed in on the region from 25k to 200k updates. Again, TTUR leads to lower FID and improved over the original learning method.

Table 6: The performance of DCGAN trained with the original procedure and with TTUR on LSUN. We compare the networks performance with respect to the FID at the optimal number of updates during training. DCGAN trained with TTUR exhibit a better FID.

method	learning rates	updates	FID
DCGAN TTUR	1e-4, 5e-4	150,000	68.3
DCGAN orig	5e-4	130,000	69.7

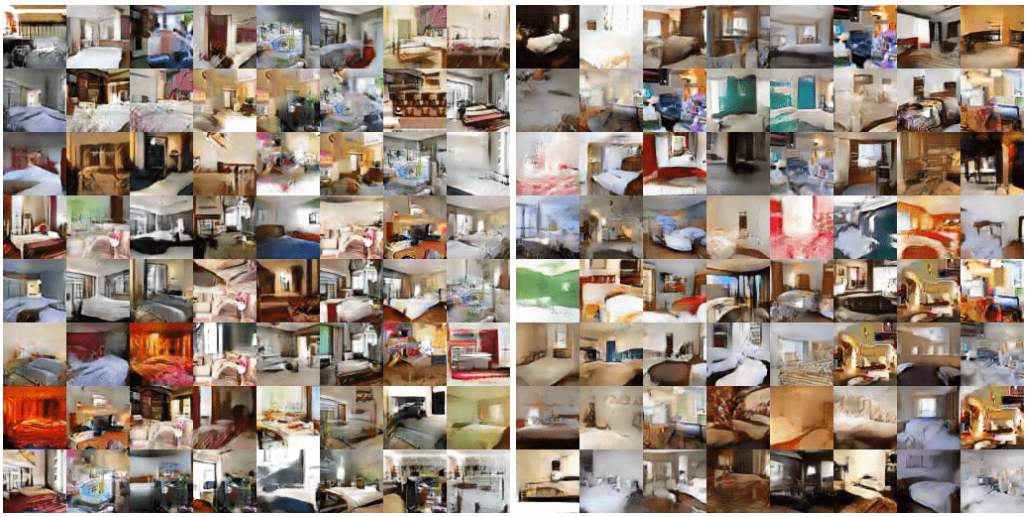

Figure 14: LSUN bedroom samples generated by DCGANs trained with the original method (left) and the TTUR method (right).

Conclusion

For learning GANs, we have introduced the two time-scale update rule (TTUR) which we have proved to converge to a stationary Nash equilibrium. We then described Adam stochastic optimization as a heavy ball with friction (HBF) dynamics, which shows that Adam converges and that Adam tends to find flat minima while missing small local minima. A second order differential equation describes the learning dynamics of Adam as a HBF system. Via this differential equation, the convergence of GANs trained with TTUR to a stationary Nash equilibrium can be extended to Adam. In experiments, we have compared GANs trained with TTUR to conventional GAN training on CelebA, the One Billion Word Benchmark, and the LSUN bedroom category. TTUR outperforms conventional GAN training with respect to performance.

References

The references are provided in Section A7.

Acknowledgment

This work was supported by NVIDIA Corporation, Zalando SE with Research Agreement 01/2016, Audi.JKU Deep Learning Center, Audi Electronic Venture GmbH, IWT research grant IWT150865 (Exaptation), H2020 project grant 671555 (ExCAPE) and FWF grant P 28660-N31.

Appendix

Contents

A1 Fréchet Inception Distance (FID)

We improve the Inception score for comparing the results of GANs [48]. The Inception score has the disadvantage that it does not use the statistics of real world samples and compared them to the statistics of synthetic samples. Let $p(.)$ be the distribution of model samples and $p_w(.)$ the distribution of the samples from real world. The equality $p(.) = p_w(.)$ holds except for a non-measurable set if and only if $\int p(.)f(x)dx = \int p_w(.)f(x)dx$ for a basis $f(.)$ spanning the function space in which $p(.)$ and $p_w(.)$ live. These equalities of expectations are used to describe distributions by moments or cumulants, where $f(x)$ are polynoms of the data x. We replacing x by the coding layer of an Inception model in order to obtain vision-relevant features and consider polynoms of the coding unit functions. For practical reasons we only consider the first two polynoms, that is, the first two moments: mean and covariance. The Gaussian is the maximum entropy distribution for given mean and covariance, therefore we assume the coding units to follow a multidimensional Gaussian. The difference of two Gaussians is measured by the Fréchet distance [15] also known as Wasserstein-2 distance [51]. The Fréchet distance $d(.,.)$ between the Gaussian with mean and covariance $(\boldsymbol{m}, \boldsymbol{C})$ obtained from $p(.)$ and the Gaussian $(\boldsymbol{m}_w, \boldsymbol{C}_w)$ obtained from $p_w(.)$ is called the "Fréchet Inception Distance" (FID), which is given by [14]:

$$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \operatorname{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right) . \tag{9}$$

Next we show that the FID is consistent with increasing disturbances and human judgment on the CelebA dataset. We computed the $(\boldsymbol{m}_w, \boldsymbol{C}_w)$ on 100,000 randomly chosen CelebA images, while for computing $(\boldsymbol{m}, \boldsymbol{C})$ we used 5,000 randomly selected samples. We considered following disturbances of the image $\boldsymbol{X}$:

1. **Gaussian noise**: We constructed a matrix $\boldsymbol{N}$ with Gaussian noise scaled to $[0, 255]$. The noisy image is computed as $(1 - \alpha)\boldsymbol{X} + \alpha\boldsymbol{N}$ for $\alpha \in \{0, 0.25, 0.5, 0.75\}$. The larger α is, the larger is the noise added to the image, the larger is the disturbance of the image.
2. **Gaussian blur**: The image is convolved with a Gaussian kernel with standard deviation $\alpha \in \{0, 1, 2, 4\}$. The larger α is, the larger is the disturbance of the image, that is, the more the image is smoothed.
3. **Black rectangles**: To an image five black rectangles are are added at randomly chosen locations. The rectangles cover parts of the image. The size of the rectangles is αimagesize with $\alpha \in \{0, 0.25, 0.5, 0.75\}$. The larger α is, the larger is the disturbance of the image, that is, the more of the image is covered by black rectangles.

4. **Swirl**: Parts of the image are transformed as a spiral, that is, as a swirl (whirlpool effect). Consider the coordinate (x, y) in the noisy (swirled) image for which we want to find the color. Toward this end we need the reverse mapping for the swirl transformation which gives the location which is mapped to (x, y). We first compute polar coordinates relative to a center (x_0, y_0) given by the angle $\theta = \arctan((y - y_0)/(x - x_0))$ and the radius $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$. We transform them according to $\theta' = \theta + \alpha e^{-5r/(\ln 2\rho)}$. Here α is a parameter for the amount of swirl and ρ indicates the swirl extent in pixels. The original coordinates, where the color for (x, y) can be found, are $x_{\text{org}} = x_0 + r\cos(\theta')$ and $y_{\text{org}} = y_0 + r\sin(\theta')$. We set (x_0, y_0) to the center of the image and $\rho = 25$. The disturbance level is given by the amount of swirl $\alpha \in \{0, 1, 2, 4\}$. The larger α is, the larger is the disturbance of the image via the amount of swirl.
5. **Salt and pepper noise**: Some pixels of the image are set to black or white, where black is chosen with 50% probability (same for white). Pixels are randomly chosen for being flipped to white or black, where the ratio of pixel flipped to white or black is given by the noise level $\alpha \in \{0, 0.1, 0.2, 0.3\}$. The larger α is, the larger is the noise added to the image via flipping pixels to white or black, the larger is the disturbance level.
6. **ImageNet contamination**: From each of the 1,000 ImageNet classes, 5 images are randomly chosen, which gives 5,000 ImageNet images. The images are ensured to be RGB and to have a minimal size of 256x256. A percentage of $\alpha \in \{0, 0.25, 0.5, 0.75\}$ of the CelebA images has been replaced by ImageNet images. $\alpha = 0$ means all images are from CelebA, $\alpha = 0.25$ means that 75% of the images are from CelebA and 25% from ImageNet etc. The larger α is, the larger is the disturbance of the CelebA dataset by contaminating it by ImageNet images. The larger the disturbance level is, the more the dataset deviates from the reference real world dataset.

We compare the Inception Score [48] with the FID. The Inception Score with m samples and K classes is

$$\exp\big(\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} p(y_k \mid \boldsymbol{X}_i)\log\frac{p(y_k \mid \boldsymbol{X}_i)}{p(y_k)}\big)\,. \tag{10}$$

The FID is a distance, while the Inception Score is a score. To compare FID and Inception Score, we transform the Inception Score to a distance, which we call "Inception Distance" (IND). This transformation to a distance is possible since the Inception Score has a maximal value. For zero probability $p(y_k \mid \boldsymbol{X}_i) = 0$, we set the value $p(y_k \mid \boldsymbol{X}_i)\log\frac{p(y_k|\boldsymbol{X}_i)}{p(y_k)} = 0$. We can bound the log-term by

$$\log\frac{p(y_k \mid \boldsymbol{X}_i)}{p(y_k)} \;\leqslant\; \log\frac{1}{1/m} \;=\; \log m\,. \tag{11}$$

Using this bound, we obtain an upper bound on the Inception Score:

$$\exp\big(\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} p(y_k \mid \boldsymbol{X}_i)\log\frac{p(y_k \mid \boldsymbol{X}_i)}{p(y_k)}\big) \tag{12}$$

$$\leqslant\; \exp\big(\log m\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} p(y_k \mid \boldsymbol{X}_i)\big) \tag{13}$$

$$=\; \exp\big(\log m\frac{1}{m}\sum_{i=1}^{m} 1\big) \;=\; m\,. \tag{14}$$

The upper bound is tight and achieved if $m \leqslant K$ and every sample is from a different class and the sample is classified correctly with probability 1. The IND is computed "IND = m - Inception Score", therefore the IND is zero for a perfect subset of the ImageNet with $m < K$ samples, where each sample stems from a different class. Therefore both distances should increase with increasing disturbance level. In the following we present the evaluation for each kind of disturbance. The larger the disturbance level is, the larger the FID and IND should be. We consider following disturbances:

1. **Gaussian noise**: Figure A15 shows the FID and the IND for Gaussian noise with different disturbance levels.

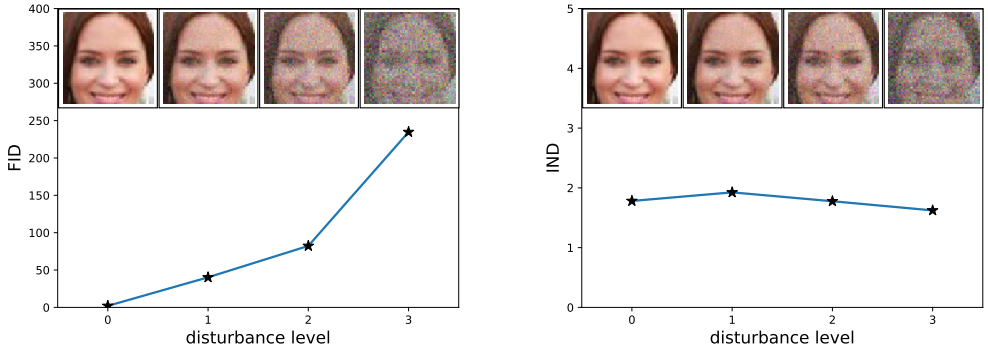

Figure A15: For Gaussian noise the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

2. **Gaussian blur**: Figure A16 shows the FID and the Id for Gaussian blur with different disturbance levels.

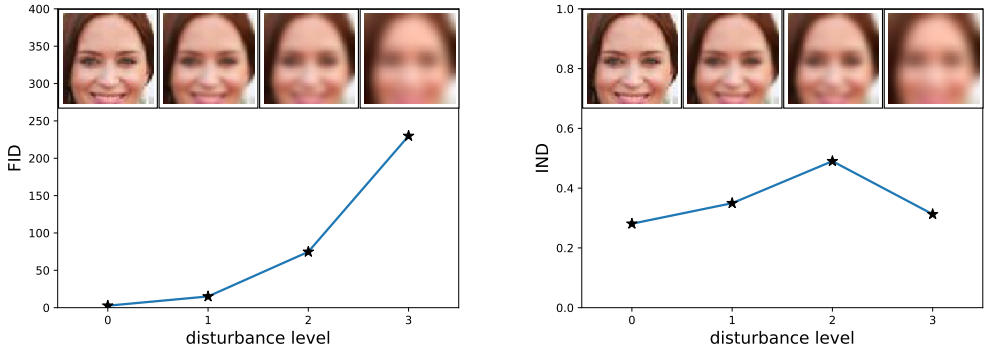

Figure A16: For Gaussian blur the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

3. **Black rectangles**: Figure A17 shows the FID and the IND for implanted rectangles with different disturbance levels.

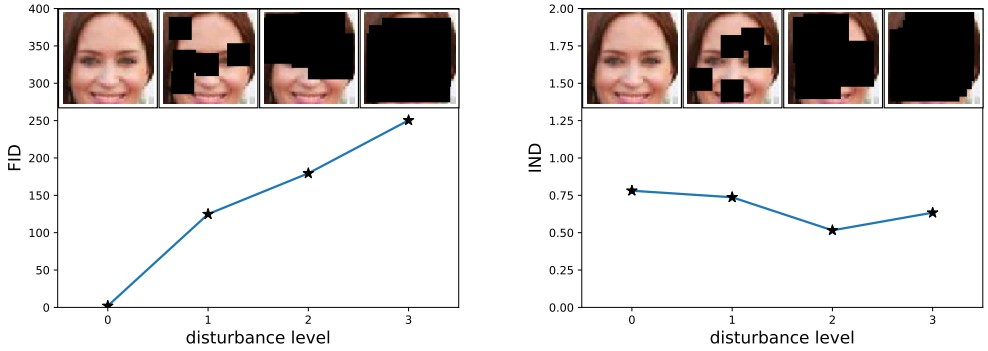

Figure A17: For implanted black rectangles the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

4. **Swirl**: Figure A18 shows the FID and the IND for swirls with different disturbance levels.

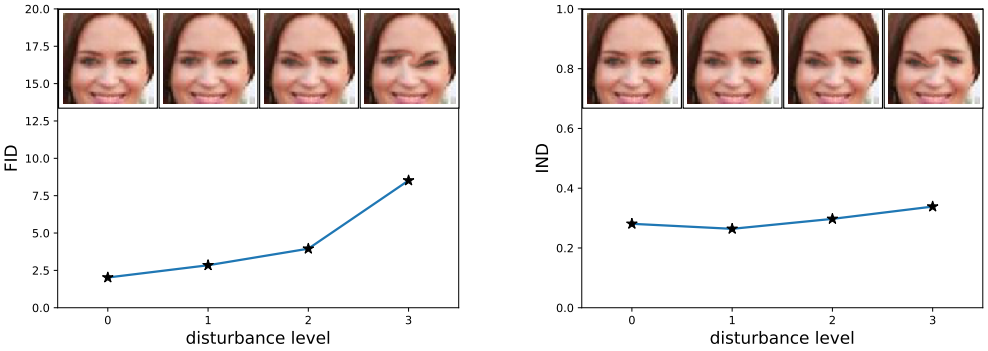

Figure A18: For swirls the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

5. **Salt and pepper noise**: Figure A19 shows the FID and the IND for salt and pepper noise with different disturbance levels.

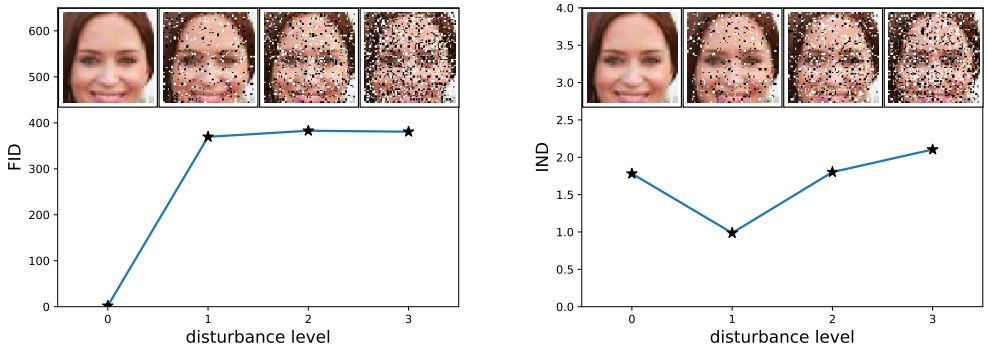

Figure A19: For salt and pepper noise the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

6. **ImageNet contamination**: Figure A20 shows the FID and the IND for ImageNet contaminations with different contamination / disturbance levels.

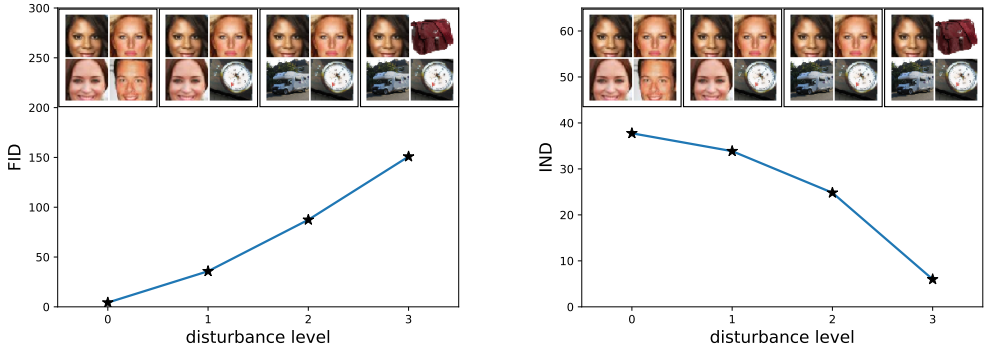

Figure A20: For ImageNet contamination the FID (left) and the IND (right) are given for different disturbance levels. The disturbance level increases from zero (left) to maximal value (right), thus a good quality measure should increase from left to right.

The following Table A7 reports the FID and the IND for different disturbances at different levels. The FID is consistent with the disturbance level while the IND is not always consistent with the disturbance level. The FID is clearly a better evaluation criteria than the Inception Score.

noise	measure	none	weak	medium	strong
Gaussian	FID	2.0	40.2	82.3	234
Gaussian	IND	2.2	2.08	2.22	2.38
Gaussian blur	FID	2.0	16.7	77.8	233
Gaussian blur	INC	2.2	2.15	2.01	2.19
Black rectangles	FID	2.0	124	179	250
Black rectangles	IND	2.2	2.26	2.48	2.37
Swirl	FID	2.0	2.8	4.0	8.5
Swirl	IND	2.2	2.24	2.20	2.16
Salt and pepper	FID	2.0	369	382	380
Salt and pepper	IND	2.2	3.00	2.19	1.91
ImageNet	FID	2.0	37.5	87.9	149
ImageNet	IND	2.2	6.14	15.2	34.0

Table A7: The FID and IND are given for different disturbances at different levels (none, weak, medium, strong). The larger the disturbance level is, the larger the FID and IND should be. The FID is consistent with the disturbance level while the IND is not always consistent with the disturbance level. The FID is clearly a better evaluation criteria than the Inception Score.

To show how the FID and the visual impression is related we show generated samples from the CelebA dataset during training with BEGAN and samples from LSUN bedroom during training with DCGAN. Both implementations maintain fixed noise vectors before the training starts and used them for generating samples from the generator afterwards, therefore it's possible to see how the generators vary their output given the fixed inputs while learning from the discriminator. For details about DCGAN and BEGAN references, architectures and training see A4.1.2 and A4.1.1 in the experiment section.

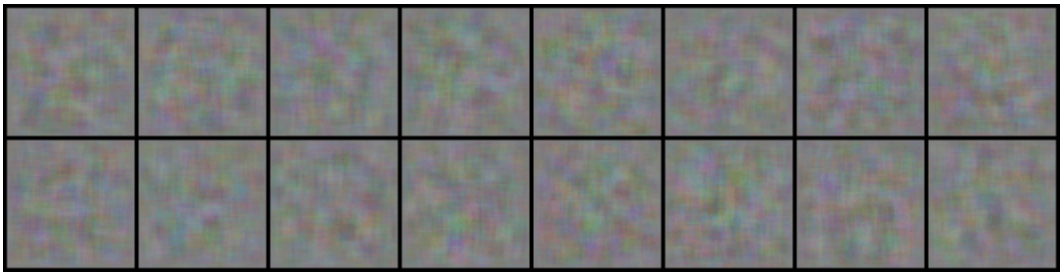

Figure A21: CelebA BEGAN mini-batch 0 FID 403.

Figure A22: CelebA BEGAN mini-batch 5000 FID 105.

Figure A23: CelebA BEGAN mini-batch 20000 FID 48.

Figure A24: CelebA BEGAN mini-batch 100000 FID 39.

Figure A25: CelebA BEGAN mini-batch 200000 FID 33.

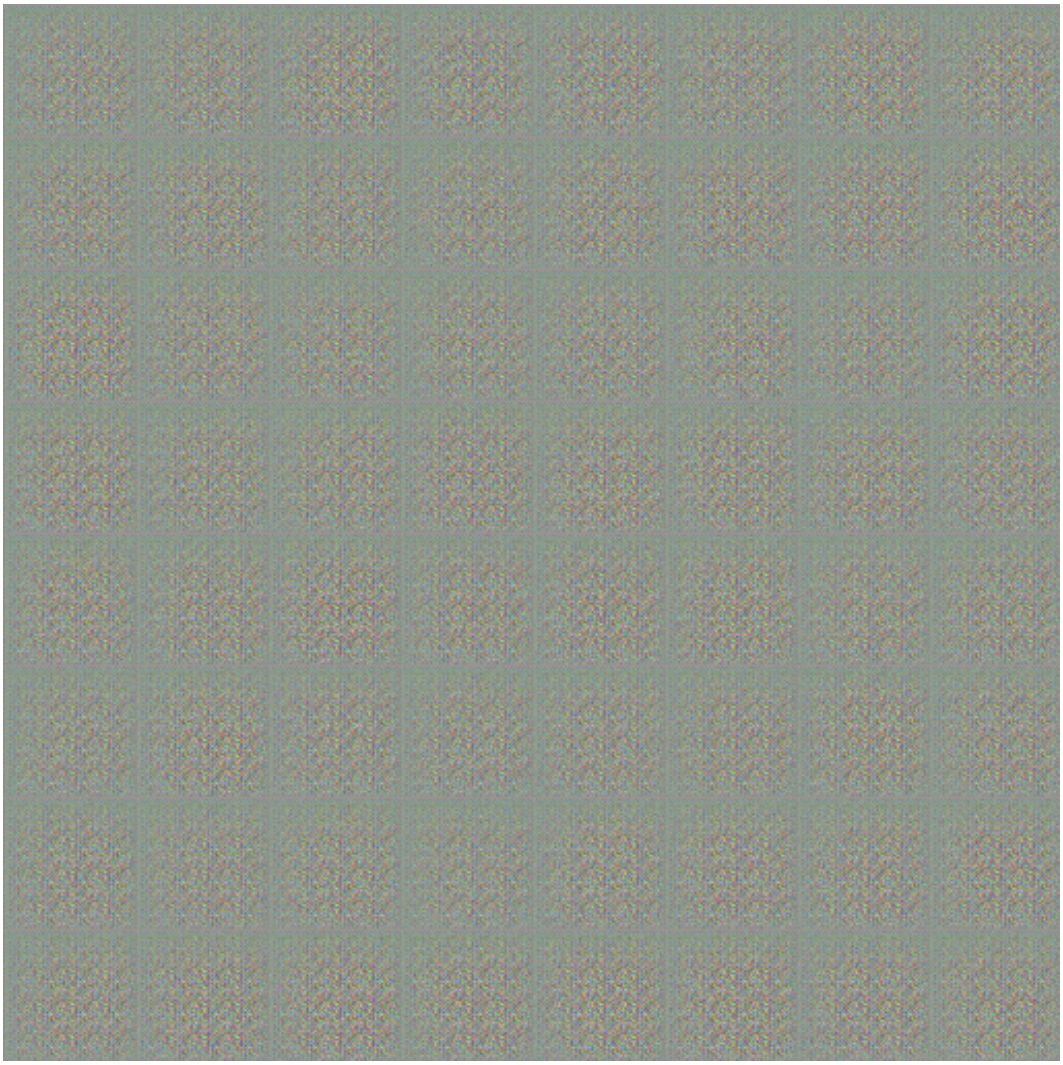

Figure A26: CelebA DCGAN mini-batch 0, FID 453.

Figure A27: CelebA DCGAN mini-batch 5000, FID 111.

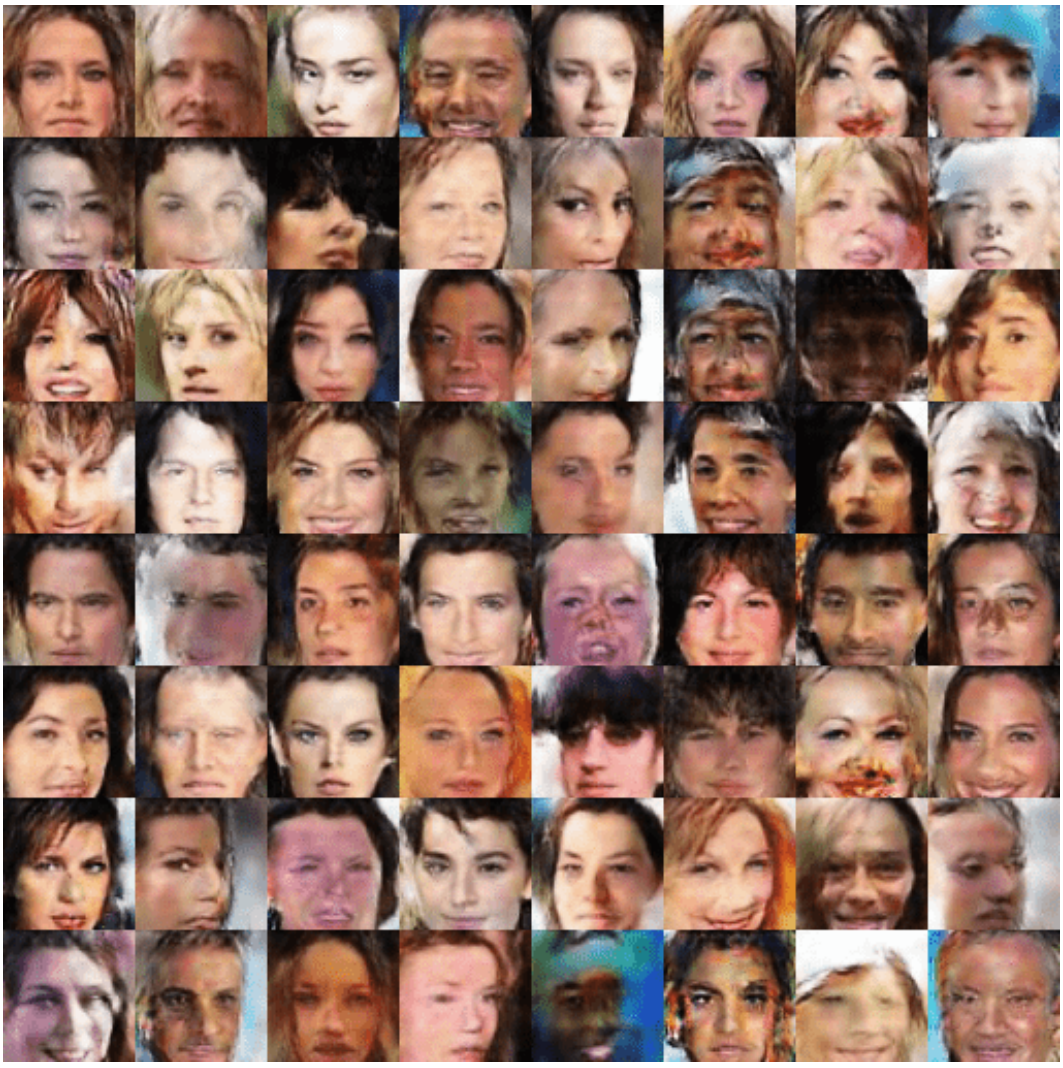

Figure A28: CelebA DCGAN mini-batch 15000, FID 29.

Figure A29: CelebA DCGAN mini-batch 45000, FID 18.

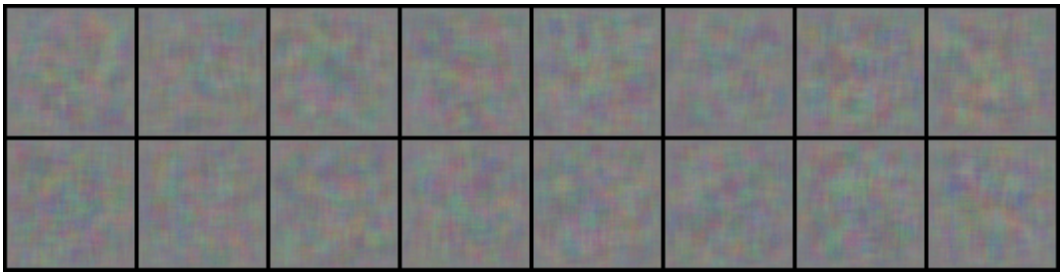

Figure A30: LSUN BEGAN mini-batch 0 FID 445.

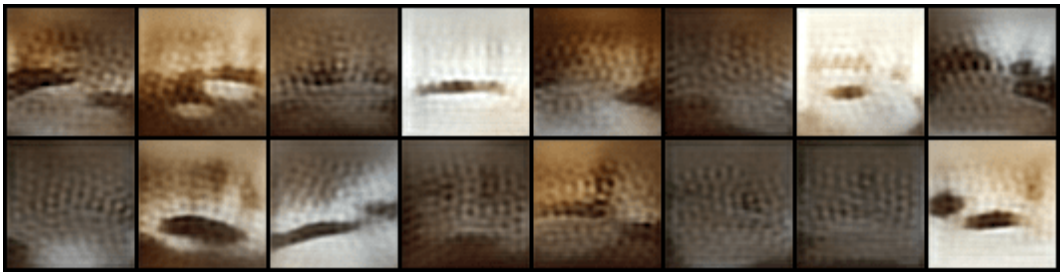

Figure A31: LSUN BEGAN mini-batch 25000 FID 233.

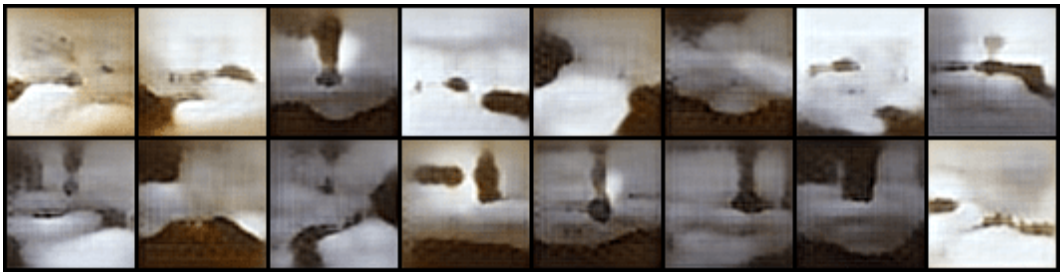

Figure A32: LSUN BEGAN mini-batch 50000 FID 174.

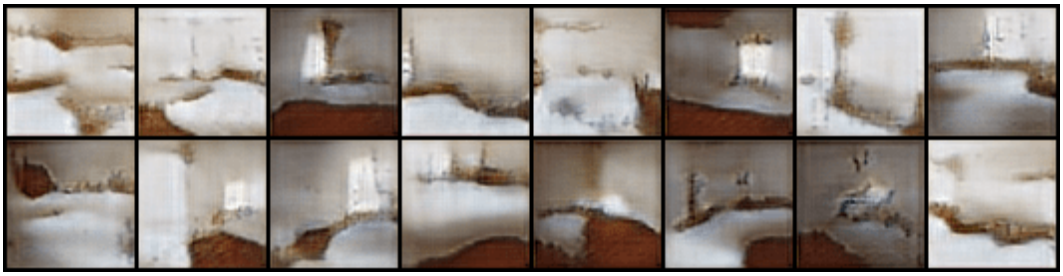

Figure A33: LSUN BEGAN mini-batch 100000 FID 129.

Figure A34: LSUN BEGAN mini-batch 150000 FID 123.

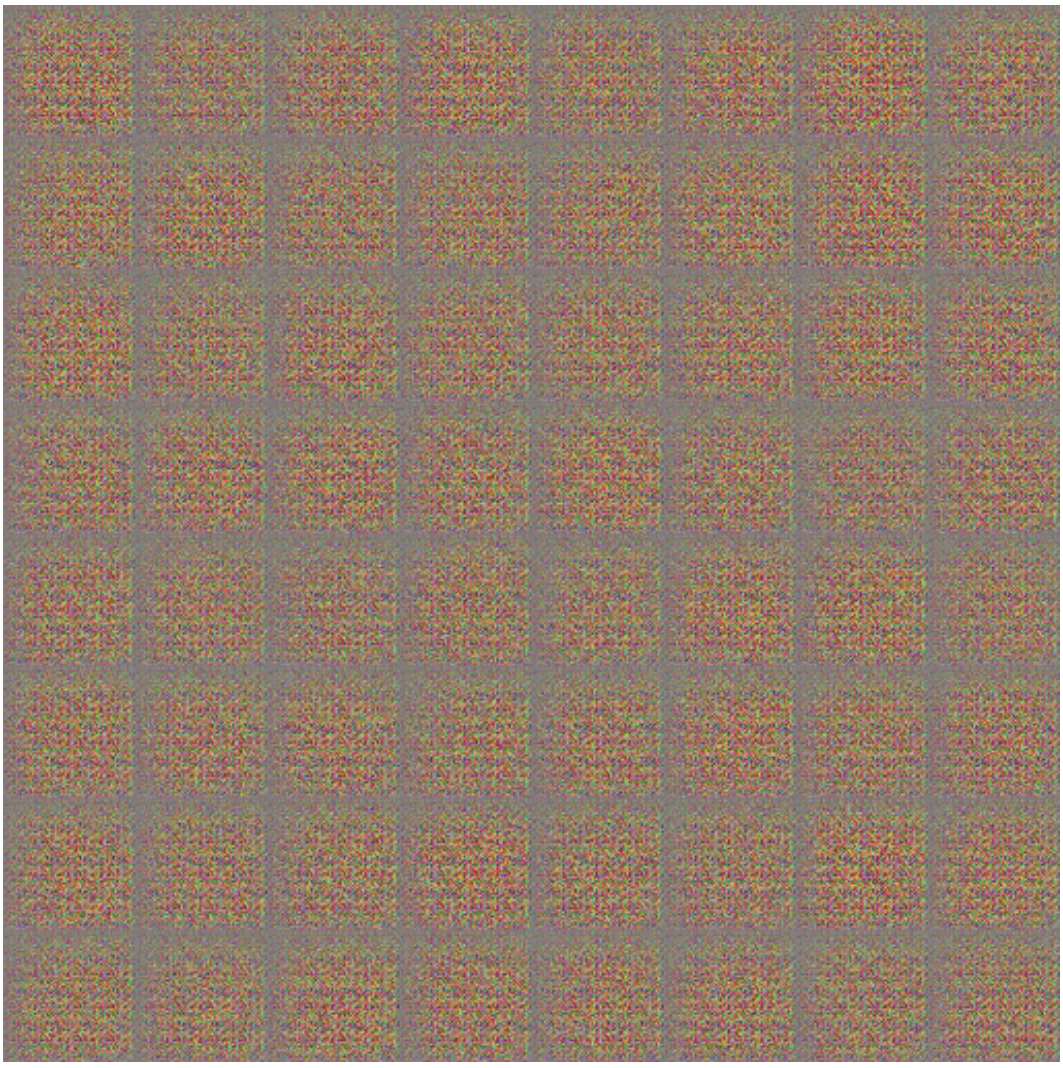

Figure A35: LSUN Bedroom DCGAN mini-batch 0 FID 360.

Figure A36: LSUN Bedroom DCGAN mini-batch 10000 FID 200.

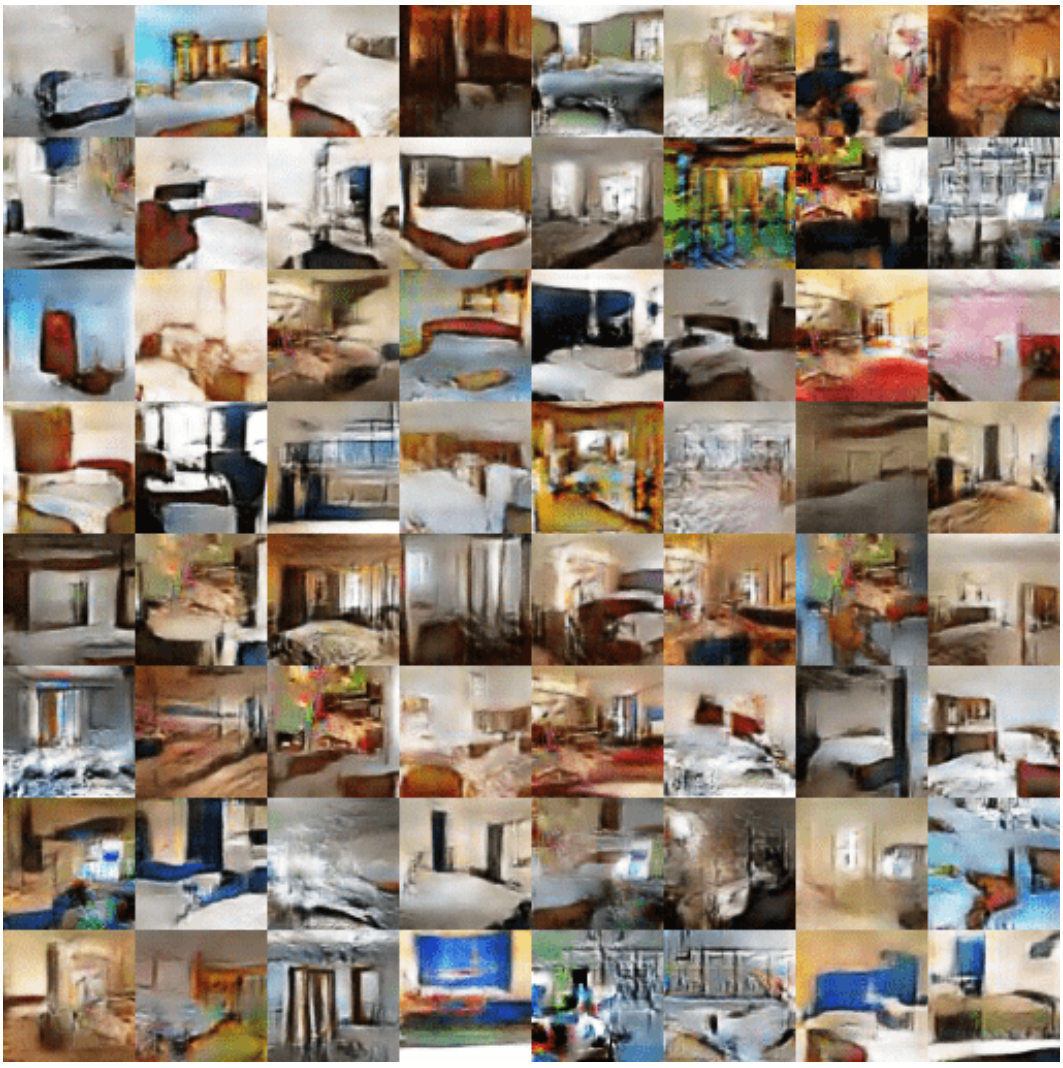

Figure A37: LSUN Bedroom DCGAN mini-batch 20000 FID 110.

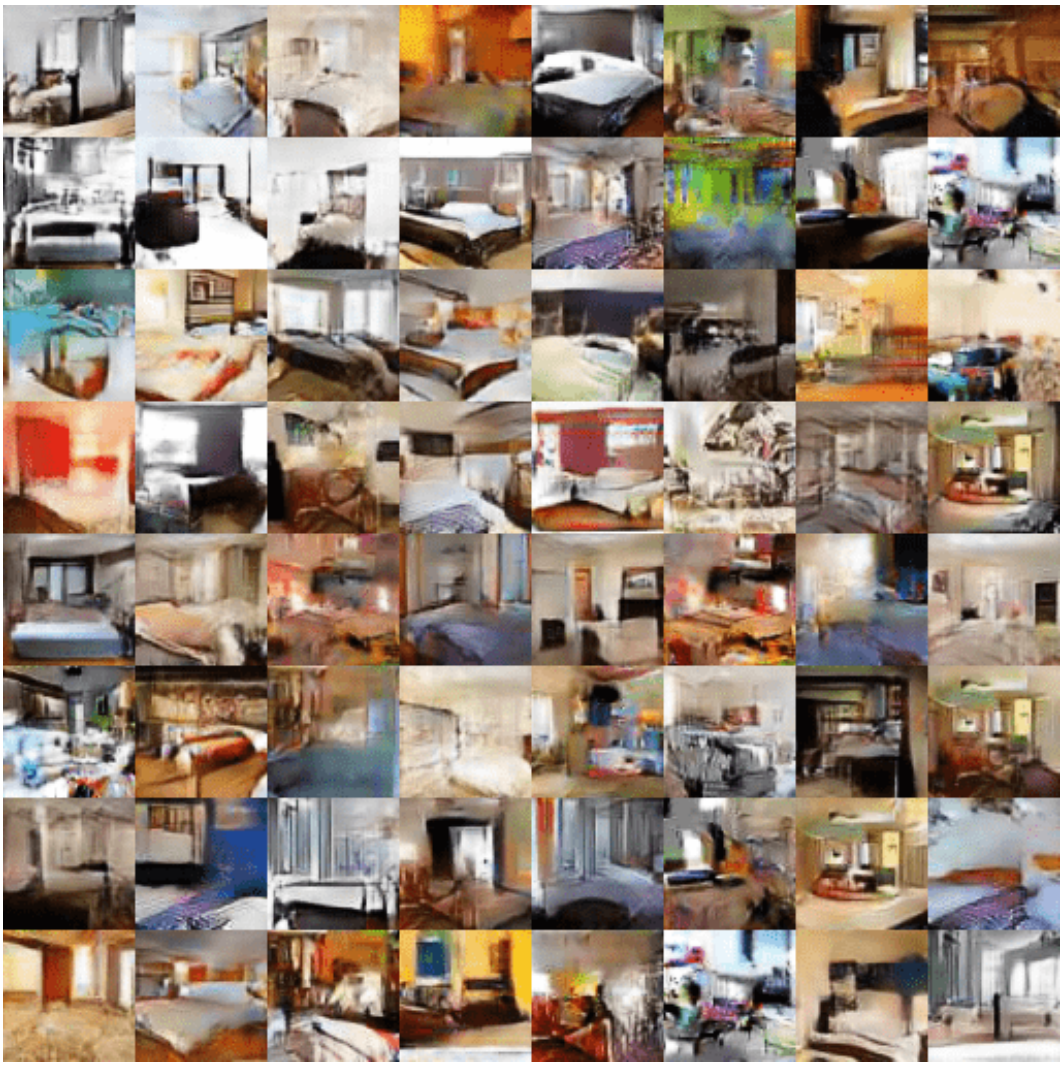

Figure A38: LSUN Bedroom DCGAN mini-batch 110000 FID 69.

A2 Two-Time Scale Stochastic Approximation Algorithms

Stochastic approximation algorithms are iterative procedures to find a root or a stationary point (minimum, maximum, saddle point) of a function when only noisy observations of its values or its derivatives are provided. Two-time scale stochastic approximation algorithms are two coupled iterations with different step sizes. For proving convergence of these interwoven iterates, it is assumed that one step size leads to considerably smaller updates than the other. The slower iterate (typically the one with smaller step size) is assumed to be slow enough to allow the fast iterate converge while being perturbed by the the slower. The perturbations of the slow should be small enough to ensure convergence of the faster.

The iterates map at time step $n \geqslant 0$ the fast variable $\boldsymbol{w}_n \in \mathbb{R}^k$ and the slow variable $\boldsymbol{\theta}_n \in \mathbb{R}^m$ to their new values:

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\Big(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n, \boldsymbol{Z}_n^{(\theta)}\big) \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\;, \tag{15}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\,\Big(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n, \boldsymbol{Z}_n^{(w)}\big) \;+\; \boldsymbol{M}_n^{(w)}\Big)\;. \tag{16}$$

The iterates use

- $\boldsymbol{h}(.) \in \mathbb{R}^m$: mapping for the slow iterate Eq. (15),
- $\boldsymbol{g}(.) \in \mathbb{R}^k$: mapping for the fast iterate Eq. (16),
- $a(n)$: step size for the slow iterate Eq. (15),
- $b(n)$: step size for the fast iterate Eq. (16),
- $\boldsymbol{M}_n^{(\theta)}$: additive random Markov process for the slow iterate Eq. (15),
- $\boldsymbol{M}_n^{(w)}$: additive random Markov process for the fast iterate Eq. (16),
- $\boldsymbol{Z}_n^{(\theta)}$: random Markov process for the slow iterate Eq. (15),
- $\boldsymbol{Z}_n^{(w)}$: random Markov process for the fast iterate Eq. (16).

A2.1 Convergence of Two-Time Scale Stochastic Approximation Algorithms

A2.1.1 Additive Noise

The first result is from Borkar 1997 [7] which was generalized in Konda and Borkar 1999 [32]. Borkar considered the iterates:

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\Big(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\;, \tag{17}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\,\Big(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{M}_n^{(w)}\Big)\;. \tag{18}$$

Assumptions. We make the following assumptions:

(A1) Assumptions on the update functions: The functions $\boldsymbol{h} : \mathbb{R}^{k+m} \mapsto \mathbb{R}^m$ and $\boldsymbol{g} : \mathbb{R}^{k+m} \mapsto \mathbb{R}^k$ are Lipschitz.

(A2) Assumptions on the learning rates:

$$\sum_n a(n) \;=\; \infty \quad , \quad \sum_n a^2(n) \;<\; \infty\,, \tag{19}$$

$$\sum_n b(n) \;=\; \infty \quad , \quad \sum_n b^2(n) \;<\; \infty\,, \tag{20}$$

$$a(n) \;=\; \mathrm{o}(b(n))\,, \tag{21}$$

(A3) Assumptions on the noise: For the increasing σ-field

$$\mathcal{F}_n \;=\; \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, l \leqslant n), n \geqslant 0\,,$$

the sequences of random variables $(\boldsymbol{M}_n^{(\theta)}, \mathcal{F}_n)$ and $(\boldsymbol{M}_n^{(w)}, \mathcal{F}_n)$ satisfy

$$\sum_n a(n)\; \boldsymbol{M}_n^{(\theta)} \;<\; \infty \text{ a.s.} \tag{22}$$

$$\sum_n b(n)\; \boldsymbol{M}_n^{(w)} \;<\; \infty \text{ a.s.}\,. \tag{23}$$

(A4) Assumption on the existence of a solution of the fast iterate: For each $\boldsymbol{\theta} \in \mathbb{R}^m$, the ODE

$$\dot{\boldsymbol{w}}(t) \;=\; \boldsymbol{g}\big(\boldsymbol{\theta}, \boldsymbol{w}(t)\big) \tag{24}$$

has a unique global asymptotically stable equilibrium $\boldsymbol{\lambda}(\boldsymbol{\theta})$ such that $\boldsymbol{\lambda} : \mathbb{R}^m \mapsto \mathbb{R}^k$ is Lipschitz.

(A5) Assumption on the existence of a solution of the slow iterate: The ODE

$$\dot{\boldsymbol{\theta}}(t) \;=\; \boldsymbol{h}\big(\boldsymbol{\theta}(t), \boldsymbol{\lambda}(\boldsymbol{\theta}(t))\big) \tag{25}$$

has a unique global asymptotically stable equilibrium $\boldsymbol{\theta}^*$.

(A6) Assumption of bounded iterates:

$$\sup_n \|\boldsymbol{\theta}_n\| \;<\; \infty \;, \tag{26}$$

$$\sup_n \|\boldsymbol{w}_n\| \;<\; \infty \;, \tag{27}$$

which can be ensured by regularization like weight decay or by proper objectives that saturate for large weights.

Convergence Theorem The next theorem is from Borkar 1997 [7].

Theorem 3 (Borkar)**.** *If the assumptions are satisfied, then the iterates Eq.* (17) *and Eq.* (18) *converge to* $(\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*))$ *a.s.*

Comments

(C1) According to Lemma 2 in [5] Assumption (A3) is fulfilled if $\{\boldsymbol{M}_n^{(\theta)}\}$ is a martingale difference sequence w.r.t $\mathcal{F}_n$ with

$$\mathrm{E}\left[\|\boldsymbol{M}_n^{(\theta)}\|^2 \mid \mathcal{F}_n^{(\theta)}\right] \;\leqslant\; B_1$$

and $\{\boldsymbol{M}_n^{(w)}\}$ is a martingale difference sequence w.r.t $\mathcal{F}_n$ with

$$\mathrm{E}\left[\|\boldsymbol{M}_n^{(w)}\|^2 \mid \mathcal{F}_n^{(w)}\right] \;\leqslant\; B_2 \;,$$

where B_1 and B_2 are positive deterministic constants.

(C2) Assumption (A3) holds for mini-batch learning which is the most frequent case of stochastic gradient. The batch gradient is $\boldsymbol{G}_n := \nabla_\theta(\frac{1}{N}\sum_{i=1}^N f(\boldsymbol{x}_i, \theta)), 1 \leqslant i \leqslant N$ and the mini-batch gradient for batch size s is $\boldsymbol{h}_n := \nabla_\theta(\frac{1}{s}\sum_{i=1}^s f(\boldsymbol{x}_{u_i}, \theta)), 1 \leqslant u_i \leqslant N$, where the indexes u_i are randomly and uniformly chosen. For the noise $\boldsymbol{M}_n^{(\theta)} := \boldsymbol{h}_n - \boldsymbol{G}_n$ we have $\mathrm{E}[\boldsymbol{M}_n^{(\theta)}] = \mathrm{E}[\boldsymbol{h}_n] - \boldsymbol{G}_n = \boldsymbol{G}_n - \boldsymbol{G}_n = 0$. Since the indexes are chosen without knowing past events, we have a martingale difference sequence. For bounded gradients we have bounded $\|\boldsymbol{M}_n^{(\theta)}\|^2$.

(C3) The assumptions (A4) and (A5) of global attractors was relaxed to local attractors via Assumption (A6)' and Theorem 2.7 in Karmakar & Bhatnagar [28]. For local attractors see also Karmakar, Bhatnagar & Ramaswamy 2016 [29].

(C4) The main result used in the proof of the theorem relies on work on perturbations of ODEs according to Hirsch 1989 [23] (see also Appendix C of Bhatnagar, Prasad, & Prashanth 2013 [6]).

(C5) Konda and Borkar 1999 [32] generalized the convergence proof to distributed asynchronous update rules.

(C6) Tadić relaxed the assumptions for showing convergence [49]. In particular the noise assumptions (Assumptions A2 in [49]) do not have to be martingale difference sequences and are more general than in [7]. In another result the assumption of bounded iterates is not necessary if other assumptions are ensured [49]. Finally, Tadić considers the case of non-additive noise [49]. **Tadić does not provide proofs for his results.** We were not able to find such proofs even in other publications of Tadić.

(C7) Typically, Assumption (A6) of bounded iterates is hard to show, however the parameters can be projected to a box which leads to a projected stochastic approximation. Theorem 5.3.1 on page 191 of Kushner & Clark [35] states convergence for projected stochastic approximations for a single iterate. See also Appendix E of Bhatnagar, Prasad, & Prashanth 2013 [6].

A2.1.2 Linear Update, Additive Noise, and Markov Chain

In contrast to previous subsection, we assume that an additional Markov chain influences the iterates [31, 33]. The Markov chain allows applications in reinforcement learning, in particular in actor-critic setting where the Markov chain is used to model the environment. The slow iterate is the actor update while the fast iterate is the critic update. For reinforcement learning both the actor and the critic observe the environment which is driven by the actor actions. The environment observations are assumed to be a Markov chain. The Markov chain can include eligibility traces which are modeled as explicit states in order to keep the Markov assumption.

The Markov chain is the sequence of observation of the environment which progresses via transition probabilities. The transitions are not affected by the critic but by the actor.

Konda et al. considered the iterates [31, 33]:

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\, \boldsymbol{H}_n \;, \tag{28}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\, \Big(\boldsymbol{g}\big(\boldsymbol{Z}_n^{(w)}; \boldsymbol{\theta}_n\big) \;+\; \boldsymbol{G}\big(\boldsymbol{Z}_n^{(w)}; \boldsymbol{\theta}_n\big)\; \boldsymbol{w}_n + \; \boldsymbol{M}_n^{(w)}\, \boldsymbol{w}_n \Big) \;. \tag{29}$$

$\boldsymbol{H}_n$ is a random process that drives the changes of $\boldsymbol{\theta}_n$. We assume that $\boldsymbol{H}_n$ is a slow enough process. We have a linear update rule for the fast iterate using the vector function $\boldsymbol{g}(.) \in \mathbb{R}^k$ and the matrix function $\boldsymbol{G}(.) \in \mathbb{R}^{k \times k}$.

Assumptions. We make the following assumptions:

(A1) Assumptions on the Markov process, that is, the transition kernel: The stochastic process $\boldsymbol{Z}_n^{(w)}$ takes values in a Polish (complete, separable, metric) space $\mathbb{Z}$ with the Borel σ-field

$$\mathcal{F}_n \;=\; \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{Z}_l^{(w)}, \boldsymbol{H}_l, l \leqslant n), n \geqslant 0 \;.$$

For every measurable set $A \subset \mathbb{Z}$ and the parametrized transition kernel $\mathrm{P}(.; \boldsymbol{\theta}_n)$ we have:

$$\mathrm{P}(\boldsymbol{Z}_{n+1}^{(w)} \in A \mid \mathcal{F}_n) \;=\; \mathrm{P}(\boldsymbol{Z}_{n+1}^{(w)} \in A \mid \boldsymbol{Z}_n^{(w)}; \boldsymbol{\theta}_n) \;=\; \mathrm{P}(\boldsymbol{Z}_n^{(w)}, A; \boldsymbol{\theta}_n) \;. \tag{30}$$

We define for every measurable function f

$$\mathrm{P}_{\boldsymbol{\theta}} f(\boldsymbol{z}) \;:=\; \int \mathrm{P}(\boldsymbol{z}, \mathrm{d}\bar{\boldsymbol{z}}; \boldsymbol{\theta}_n)\; f(\bar{\boldsymbol{z}}) \;.$$

(A2) Assumptions on the learning rates:

$$\sum_n b(n) \;=\; \infty \quad , \quad \sum_n b^2(n) \;<\; \infty \;, \tag{31}$$

$$\sum_n \left(\frac{a(n)}{b(n)} \right)^d \;<\; \infty \;, \tag{32}$$

for some $d > 0$.

(A3) Assumptions on the noise: The sequence $\boldsymbol{M}_n^{(w)}$ is a $k \times k$-matrix valued $\mathcal{F}_n$-martingale difference with bounded moments:

$$\mathrm{E}\left[\boldsymbol{M}_n^{(w)} \mid \mathcal{F}_n \right] \;=\; 0 \;, \tag{33}$$

$$\sup_n \mathrm{E}\left[\left\| \boldsymbol{M}_n^{(w)} \right\|^d \right] \;<\; \infty \;,\; \forall d > 0 \;. \tag{34}$$

We assume slowly changing $\boldsymbol{\theta}$, therefore the random process $\boldsymbol{H}_n$ satisfies

$$\sup_n \mathrm{E}\left[\|\boldsymbol{H}_n\|^d \right] \;<\; \infty \;,\; \forall d > 0 \;. \tag{35}$$

(A4) Assumption on the existence of a solution of the fast iterate: We assume the existence of a solution to the Poisson equation for the fast iterate. For each $\boldsymbol{\theta} \in \mathbb{R}^m$, there exist functions $\bar{\boldsymbol{g}}(\boldsymbol{\theta}) \in \mathbb{R}^k$, $\bar{\boldsymbol{G}}(\boldsymbol{\theta}) \in \mathbb{R}^{k \times k}$, $\hat{\boldsymbol{g}}(\boldsymbol{z}; \boldsymbol{\theta}) : \mathbb{Z} \to \mathbb{R}^k$, and $\hat{\boldsymbol{G}}(\boldsymbol{z}; \boldsymbol{\theta}) : \mathbb{Z} \to \mathbb{R}^{k \times k}$ that satisfy the Poisson equations:

$$\hat{\boldsymbol{g}}(\boldsymbol{z}; \boldsymbol{\theta}) \;=\; \boldsymbol{g}(\boldsymbol{z}; \boldsymbol{\theta}) \;-\; \bar{\boldsymbol{g}}(\boldsymbol{\theta}) \;+\; (\mathrm{P}_{\boldsymbol{\theta}} \hat{\boldsymbol{g}}(.; \boldsymbol{\theta}))(\boldsymbol{z}) \;, \tag{36}$$

$$\hat{\boldsymbol{G}}(\boldsymbol{z}; \boldsymbol{\theta}) \;=\; \boldsymbol{G}(\boldsymbol{z}; \boldsymbol{\theta}) \;-\; \bar{\boldsymbol{G}}(\boldsymbol{\theta}) \;+\; (\mathrm{P}_{\boldsymbol{\theta}} \hat{\boldsymbol{G}}(.; \boldsymbol{\theta}))(\boldsymbol{z}) \;. \tag{37}$$

(A5) Assumptions on the update functions and solutions to the Poisson equation:

(a) Boundedness of solutions: For some constant C and for all $\boldsymbol{\theta}$:

$$\max\{\|\bar{g}(\boldsymbol{\theta})\|\} \leqslant C\,, \tag{38}$$

$$\max\{\|\bar{G}(\boldsymbol{\theta})\|\} \leqslant C\,. \tag{39}$$

(b) Boundedness in expectation: All moments are bounded. For any $d > 0$, there exists $C_d > 0$ such that

$$\sup_n \mathrm{E}\left[\left\|\hat{\boldsymbol{g}}(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta})\right\|^d\right] \leqslant C_d\,, \tag{40}$$

$$\sup_n \mathrm{E}\left[\left\|\boldsymbol{g}(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta})\right\|^d\right] \leqslant C_d\,, \tag{41}$$

$$\sup_n \mathrm{E}\left[\left\|\hat{\boldsymbol{G}}(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta})\right\|^d\right] \leqslant C_d\,, \tag{42}$$

$$\sup_n \mathrm{E}\left[\left\|\boldsymbol{G}(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta})\right\|^d\right] \leqslant C_d\,. \tag{43}$$

(c) Lipschitz continuity of solutions: For some constant $C > 0$ and for all $\boldsymbol{\theta}$,$\bar{\boldsymbol{\theta}} \in \mathbb{R}^m$:

$$\left\|\bar{\boldsymbol{g}}(\boldsymbol{\theta}) - \bar{\boldsymbol{g}}(\bar{\boldsymbol{\theta}})\right\| \leqslant C\,\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|\,, \tag{44}$$

$$\left\|\bar{\boldsymbol{G}}(\boldsymbol{\theta}) - \bar{\boldsymbol{G}}(\bar{\boldsymbol{\theta}})\right\| \leqslant C\,\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|\,. \tag{45}$$

(d) Lipschitz continuity in expectation: There exists a positive measurable function $C(.)$ on $\mathbb{Z}$ such that

$$\sup_n \mathrm{E}\left[C(\boldsymbol{Z}_n^{(w)})^d\right] < \infty\,,\ \forall d > 0\,. \tag{46}$$

Function $C(.)$ gives the Lipschitz constant for every $\boldsymbol{z}$:

$$\left\|(\mathrm{P}_{\boldsymbol{\theta}}\hat{\boldsymbol{g}}(.;\boldsymbol{\theta}))(\boldsymbol{z}) - (\mathrm{P}_{\bar{\boldsymbol{\theta}}}\hat{\boldsymbol{g}}(.;\bar{\boldsymbol{\theta}}))(\boldsymbol{z})\right\| \leqslant C(\boldsymbol{z})\,\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|\,, \tag{47}$$

$$\left\|(\mathrm{P}_{\boldsymbol{\theta}}\hat{\boldsymbol{G}}(.;\boldsymbol{\theta}))(\boldsymbol{z}) - (\mathrm{P}_{\bar{\boldsymbol{\theta}}}\hat{\boldsymbol{G}}(.;\bar{\boldsymbol{\theta}}))(\boldsymbol{z})\right\| \leqslant C(\boldsymbol{z})\,\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}\|\,. \tag{48}$$

(e) Uniform positive definiteness: There exists some $\alpha > 0$ such that for all $\boldsymbol{w} \in \mathbb{R}^k$ and $\boldsymbol{\theta} \in \mathbb{R}^m$:

$$\boldsymbol{w}^T\,\bar{G}(\boldsymbol{\theta})\,\boldsymbol{w} \geqslant \alpha\,\|\boldsymbol{w}\|^2\,. \tag{49}$$

Convergence Theorem. We report Theorem 3.2 (see also Theorem 7 in [33]) and Theorem 3.13 from [31]:

Theorem 4 (Konda & Tsitsiklis)**.** *If the assumptions are satisfied, then for the iterates Eq.* (28) *and Eq.* (29) *holds:*

$$\lim_{n\to\infty}\left\|\bar{G}(\boldsymbol{\theta}_n)\,\boldsymbol{w}_n - \bar{g}(\boldsymbol{\theta}_n)\right\| = 0 \quad \textit{a.s.}\,, \tag{50}$$

$$\lim_{n\to\infty}\left\|\boldsymbol{w}_n - \bar{G}^{-1}(\boldsymbol{\theta}_n)\,\bar{g}(\boldsymbol{\theta}_n)\right\| = 0\,. \tag{51}$$

Comments.

(C1) The proofs only use the boundedness of the moments of $\boldsymbol{H}_n$ [31, 33], therefore $\boldsymbol{H}_n$ may depend on $\boldsymbol{w}_n$. In his PhD thesis [31], Vijaymohan Konda used this framework for the actor-critic learning, where $\boldsymbol{H}_n$ drives the updates of the actor parameters $\boldsymbol{\theta}_n$. However the actor updates are based on the current parameters $\boldsymbol{w}_n$ of the critic.

(C2) The random process $\boldsymbol{Z}_n^{(w)}$ can affect $\boldsymbol{H}_n$ as long as boundedness is ensured.

(C3) Nonlinear update rule. $\boldsymbol{g}\big(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta}_n\big) + \boldsymbol{G}\big(\boldsymbol{Z}_n^{(w)};\boldsymbol{\theta}_n\big)\boldsymbol{w}_n$ can be viewed as a linear approximation of a nonlinear update rule. The nonlinear case has been considered in [31] where additional approximation errors due to linearization were addressed. These errors are treated in the given framework [31].

A2.1.3 Additive Noise and Controlled Markov Processes

The most general iterates use nonlinear update functions $\boldsymbol{g}$ and $\boldsymbol{h}$, have additive noise, and have controlled Markov processes [28]. A similar analysis has been performed without controlled Markov processes [47].

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\;\Big(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n, \boldsymbol{Z}_n^{(\theta)}\big) \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\;, \tag{52}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\;\Big(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n, \boldsymbol{Z}_n^{(w)}\big) \;+\; \boldsymbol{M}_n^{(w)}\Big)\;. \tag{53}$$

Required Definitions. *Marchaud Map*: A set-valued map $\boldsymbol{h} : \mathbb{R}^l \to \{\text{subsets of } \mathbb{R}^k\}$ is called a *Marchaud map* if it satisfies the following properties:

(i) For each $\boldsymbol{\theta} \in \mathbb{R}^l$, $\boldsymbol{h}(\boldsymbol{\theta})$ is convex and compact.

(ii) *(point-wise boundedness)* For each $\boldsymbol{\theta} \in \mathbb{R}^l$, $\sup_{\boldsymbol{w} \in \boldsymbol{h}(\boldsymbol{\theta})} \|\boldsymbol{w}\| < K\,(1 + \|\boldsymbol{\theta}\|)$ for some $K > 0$.

(iii) $\boldsymbol{h}$ is an *upper-semicontinuous* map.
We say that $\boldsymbol{h}$ is upper-semicontinuous, if given sequences $\{\boldsymbol{\theta}_n\}_{n\geq 1}$ (in $\mathbb{R}^l$) and $\{\boldsymbol{y}_n\}_{n\geq 1}$ (in $\mathbb{R}^k$) with $\boldsymbol{\theta}_n \to \boldsymbol{\theta}$, $\boldsymbol{y}_n \to \boldsymbol{y}$ and $\boldsymbol{y}_n \in \boldsymbol{h}(\boldsymbol{\theta}_n), n \geq 1, \boldsymbol{y} \in \boldsymbol{h}(\boldsymbol{\theta})$. In other words, the graph of $\boldsymbol{h}$, $\big\{(\boldsymbol{x}, \boldsymbol{y}) \;:\; \boldsymbol{y} \in \boldsymbol{h}(\boldsymbol{x}),\; \boldsymbol{x} \in \mathbb{R}^l\big\}$, is closed in $\mathbb{R}^l \times \mathbb{R}^k$.

If the set-valued map $H : \mathbb{R}^m \to \{\text{subsets of } \mathbb{R}^m\}$ is Marchaud, then the differential inclusion (DI) given by

$$\dot{\boldsymbol{\theta}}(t) \;\in\; H(\boldsymbol{\theta}(t)) \tag{54}$$

is guaranteed to have at least one solution that is absolutely continuous. $\boldsymbol{\Sigma}$ is defined as the set of all absolutely continuous maps $\boldsymbol{\Theta}$ that satisfy Eq. (54).

Invariant Set: $M \subseteq \mathbb{R}^m$ is *invariant* if for every $\boldsymbol{\theta} \in M$ there exists a complete trajectory, $\boldsymbol{\Theta}$, entirely in M with $\boldsymbol{\Theta}(0) = \boldsymbol{\theta}$. In other words, $\boldsymbol{\Theta} \in \boldsymbol{\Sigma}$ (complete trajectory) with $\boldsymbol{\Theta}(t) \in M$, for all $t \geq 0$.
Internally Chain Transitive Set: $M \subset \mathbb{R}^m$ is said to be internally chain transitive if M is compact and for every $\boldsymbol{\theta}, \boldsymbol{y} \in M$, $\epsilon > 0$ and $T > 0$ we have the following: There exist $\Phi^1, \ldots, \Phi^n$ that are n solutions to the differential inclusion $\dot{\boldsymbol{\theta}}(t) \in h(\boldsymbol{\theta}(t))$, a sequence $\boldsymbol{\theta}_1(=\boldsymbol{\theta}), \ldots, \boldsymbol{\theta}_{n+1}(=\boldsymbol{y}) \subset M$ and n real numbers $t_1, t_2, \ldots, t_n$ greater than T such that: $\Phi^i_{t_i}(\boldsymbol{\theta}_i) \in N^\epsilon(\boldsymbol{\theta}_{i+1})$ ($N^\epsilon(\boldsymbol{\theta})$ is the open ϵ-neighborhood of $\boldsymbol{\theta}$) and $\Phi^i_{[0,t_i]}(\boldsymbol{\theta}_i) \subset M$ for $1 \leq i \leq n$. The sequence $(\boldsymbol{\theta}_1(=x), \ldots, \boldsymbol{\theta}_{n+1}(=\boldsymbol{y}))$ is called an (ϵ, T) chain in M from $\boldsymbol{\theta}$ to $\boldsymbol{y}$.

Assumptions. We make the following assumptions [28]:

(A1) Assumptions on the controlled Markov processes: The controlled Markov process $\{\boldsymbol{Z}_n^{(w)}\}$ takes values in a compact metric space $S^{(w)}$. The controlled Markov process $\{\boldsymbol{Z}_n^{(\theta)}\}$ takes values in a compact metric space $S^{(\theta)}$. Both processes are controlled by the iterate sequences $\{\boldsymbol{\theta}_n\}$ and $\{\boldsymbol{w}_n\}$. Furthermore $\{\boldsymbol{Z}_n^{(w)}\}$ is additionally controlled by a random process $\{\boldsymbol{A}_n^{(w)}\}$ taking values in a compact metric space $U^{(w)}$ and $\{\boldsymbol{Z}_n^{(\theta)}\}$ is additionally controlled by a random process $\{\boldsymbol{A}_n^{(\theta)}\}$ taking values in a compact metric space $U^{(\theta)}$. The $\{\boldsymbol{Z}_n^{(\theta)}\}$ dynamics is

$$\mathrm{P}(\boldsymbol{Z}_{n+1}^{(\theta)} \in B^{(\theta)} | \boldsymbol{Z}_l^{(\theta)}, \boldsymbol{A}_l^{(\theta)}, \boldsymbol{\theta}_l, \boldsymbol{w}_l, l \leqslant n) \;=\; \int_{B^{(\theta)}} p^{(\theta)}(\mathrm{d}z | \boldsymbol{Z}_n^{(\theta)}, \boldsymbol{A}_n^{(\theta)}, \boldsymbol{\theta}_n, \boldsymbol{w}_n), n \geqslant 0\;, \tag{55}$$

for $B^{(\theta)}$ Borel in $S^{(\theta)}$. The $\{\boldsymbol{Z}_n^{(w)}\}$ dynamics is

$$\mathrm{P}(\boldsymbol{Z}_{n+1}^{(w)} \in B^{(w)} | \boldsymbol{Z}_l^{(w)}, \boldsymbol{A}_l^{(w)}, \boldsymbol{\theta}_l, \boldsymbol{w}_l, l \leqslant n) \;=\; \int_{B^{(w)}} p^{(w)}(\mathrm{d}z | \boldsymbol{Z}_n^{(w)}, \boldsymbol{A}_n^{(w)}, \boldsymbol{\theta}_n, \boldsymbol{w}_n), n \geqslant 0\;, \tag{56}$$

for $B^{(w)}$ Borel in $S^{(w)}$.

(A2) Assumptions on the update functions: $\boldsymbol{h} : \mathbb{R}^{m+k} \times S^{(\theta)} \to \mathbb{R}^m$ is jointly continuous as well as Lipschitz in its first two arguments uniformly w.r.t. the third. The latter condition means that

$$\forall \boldsymbol{z}^{(\theta)} \in S^{(\theta)} : \ \|\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{z}^{(\theta)}) \ - \ \boldsymbol{h}(\boldsymbol{\theta}', \boldsymbol{w}', \boldsymbol{z}^{(\theta)})\| \ \leqslant \ L^{(\theta)} \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + \|\boldsymbol{w} - \boldsymbol{w}'\|\right) . \tag{57}$$

Note that the Lipschitz constant $L^{(\theta)}$ does not depend on $\boldsymbol{z}^{(\theta)}$.

$\boldsymbol{g} : \mathbb{R}^{k+m} \times S^{(w)} \to \mathbb{R}^k$ is jointly continuous as well as Lipschitz in its first two arguments uniformly w.r.t. the third. The latter condition means that

$$\forall \boldsymbol{z}^{(w)} \in S^{(w)} : \ \|\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{z}^{(w)}) \ - \ \boldsymbol{g}(\boldsymbol{\theta}', \boldsymbol{w}', \boldsymbol{z}^{(w)})\| \ \leqslant \ L^{(w)} \left(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\| + \|\boldsymbol{w} - \boldsymbol{w}'\|\right) . \tag{58}$$

Note that the Lipschitz constant $L^{(w)}$ does not depend on $\boldsymbol{z}^{(w)}$.

(A3) Assumptions on the additive noise: $\{\boldsymbol{M}_n^{(\theta)}\}$ and $\{\boldsymbol{M}_n^{(w)}\}$ are martingale difference sequence with second moments bounded by $K(1 + \|\boldsymbol{\theta}_n\|^2 + \|\boldsymbol{w}_n\|^2)$. More precisely, $\{\boldsymbol{M}_n^{(\theta)}\}$ is a martingale difference sequence w.r.t. increasing σ-fields

$$\mathcal{F}_n \ = \ \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, \boldsymbol{Z}_l^{(\theta)}, \boldsymbol{Z}_l^{(w)}, l \leqslant n), \ n \geqslant 0 \, , \tag{59}$$

satisfying

$$\mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(\theta)}\|^2 \mid \mathcal{F}_n\right] \ \leqslant \ K \left(1 \ + \ \|\boldsymbol{\theta}_n\|^2 \ + \ \|\boldsymbol{w}_n\|^2\right) , \tag{60}$$

for $n \geqslant 0$ and a given constant $K > 0$.

$\{\boldsymbol{M}_n^{(w)}\}$ is a martingale difference sequence w.r.t. increasing σ-fields

$$\mathcal{F}_n \ = \ \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, \boldsymbol{Z}_l^{(\theta)}, \boldsymbol{Z}_l^{(w)}, l \leqslant n), \ n \geqslant 0 \, , \tag{61}$$

satisfying

$$\mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(w)}\|^2 \mid \mathcal{F}_n\right] \ \leqslant \ K \left(1 \ + \ \|\boldsymbol{\theta}_n\|^2 \ + \ \|\boldsymbol{w}_n\|^2\right) , \tag{62}$$

for $n \geqslant 0$ and a given constant $K > 0$.

(A4) Assumptions on the learning rates:

$$\sum_n a(n) \ = \ \infty \quad , \quad \sum_n a^2(n) \ < \ \infty \, , \tag{63}$$

$$\sum_n b(n) \ = \ \infty \quad , \quad \sum_n b^2(n) \ < \ \infty \, , \tag{64}$$

$$a(n) \ = \ \mathrm{o}(b(n)) \, , \tag{65}$$

Furthermore, $a(n), b(n), n \geqslant 0$ are non-increasing.

(A5) Assumptions on the controlled Markov processes, that is, the transition kernels: The state-action map

$$S^{(\theta)} \times U^{(\theta)} \times \mathbb{R}^{m+k} \ni (\boldsymbol{z}^{(\theta)}, \boldsymbol{a}^{(\theta)}, \boldsymbol{\theta}, \boldsymbol{w}) \ \to \ \mathrm{p}^{(\theta)}(\mathrm{d}\boldsymbol{y} \mid \boldsymbol{z}^{(\theta)}, \boldsymbol{a}^{(\theta)}, \boldsymbol{\theta}, \boldsymbol{w}) \tag{66}$$

and the state-action map

$$S^{(w)} \times U^{(w)} \times \mathbb{R}^{m+k} \ni (\boldsymbol{z}^{(w)}, \boldsymbol{a}^{(w)}, \boldsymbol{\theta}, \boldsymbol{w}) \ \to \ \mathrm{p}^{(w)}(\mathrm{d}\boldsymbol{y} \mid \boldsymbol{z}^{(w)}, \boldsymbol{a}^{(w)}, \boldsymbol{\theta}, \boldsymbol{w}) \tag{67}$$

are continuous.

(A6) Assumptions on the existence of a solution:

We consider *occupation measures* which give for the controlled Markov process the probability or density to observe a particular state-action pair from $S \times U$ for given $\boldsymbol{\theta}$ and a given control policy π. We denote by $D^{(w)}(\boldsymbol{\theta}, \boldsymbol{w})$ the set of all ergodic occupation measures for the prescribed $\boldsymbol{\theta}$ and $\boldsymbol{w}$ on state-action space $S^{(w)} \times U^{(\theta)}$ for the controlled Markov

process $\boldsymbol{Z}^{(w)}$ with policy $\pi^{(w)}$. Analogously we denote, by $D^{(\theta)}(\boldsymbol{\theta}, \boldsymbol{w})$ the set of all ergodic occupation measures for the prescribed $\boldsymbol{\theta}$ and $\boldsymbol{w}$ on state-action space $S^{(\theta)} \times U^{(\theta)}$ for the controlled Markov process $\boldsymbol{Z}^{(\theta)}$ with policy $\pi^{(\theta)}$. Define

$$\tilde{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\nu}) \;=\; \int \boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{z})\, \boldsymbol{\nu}(\mathrm{d}\boldsymbol{z}, U^{(w)}) \tag{68}$$

for $\boldsymbol{\nu}$ a measure on $S^{(w)} \times U^{(w)}$ and the Marchaud map

$$\hat{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}) \;=\; \{\tilde{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{\nu}) :\; \boldsymbol{\nu} \in D^{(w)}(\boldsymbol{\theta}, \boldsymbol{w})\}\,. \tag{69}$$

We assume that the set $D^{(w)}(\boldsymbol{\theta}, \boldsymbol{w})$ is singleton, that is, $\hat{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w})$ contains a single function and we use the same notation for the set and its single element. If the set is not a singleton, the assumption of a solution can be expressed by the differential inclusion $\dot{\boldsymbol{w}}(t) \in \hat{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}(t))$ [28].

$\forall \boldsymbol{\theta} \in \mathbb{R}^m$, the ODE

$$\dot{\boldsymbol{w}}(t) \;=\; \hat{\boldsymbol{g}}(\boldsymbol{\theta}, \boldsymbol{w}(t)) \tag{70}$$

has an asymptotically stable equilibrium $\boldsymbol{\lambda}(\boldsymbol{\theta})$ with domain of attraction G_θ where $\boldsymbol{\lambda} : \mathbb{R}^m \to \mathbb{R}^k$ is a Lipschitz map with constant K. Moreover, the function $V : G \to [0, \infty)$ is continuously differentiable where $V(\boldsymbol{\theta}, .)$ is the Lyapunov function for $\boldsymbol{\lambda}(\boldsymbol{\theta})$ and $G = \{(\boldsymbol{\theta}, \boldsymbol{w}) :\; \boldsymbol{w} \in G_\theta, \boldsymbol{\theta} \in \mathbb{R}^m\}$. This extra condition is needed so that the set $\{(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta})) : \boldsymbol{\theta} \in \mathbb{R}^m\}$ becomes an asymptotically stable set of the coupled ODE

$$\dot{\boldsymbol{w}}(t) \;=\; \hat{\boldsymbol{g}}(\boldsymbol{\theta}(t), \boldsymbol{w}(t)) \tag{71}$$

$$\dot{\boldsymbol{\theta}}(t) \;=\; 0\,. \tag{72}$$

(A7) Assumption of bounded iterates:

$$\sup_n \|\boldsymbol{\theta}_n\| \;<\; \infty \text{ a.s.}\,, \tag{73}$$

$$\sup_n \|\boldsymbol{w}_n\| \;<\; \infty \text{ a.s.} \tag{74}$$

Convergence Theorem. The following theorem is from Karmakar & Bhatnagar [28]:

Theorem 5 (Karmakar & Bhatnagar)**.** *Under above assumptions if for all $\boldsymbol{\theta} \in \mathbb{R}^m$, with probability 1, $\{\boldsymbol{w}_n\}$ belongs to a compact subset Q_θ (depending on the sample point) of G_θ "eventually", then*

$$(\boldsymbol{\theta}_n, \boldsymbol{w}_n) \;\to\; \cup_{\boldsymbol{\theta}^* \in A_0} (\boldsymbol{\theta}^*, \boldsymbol{\lambda}(\boldsymbol{\theta}^*)) \;\; a.s. \quad as\; n \;\to\; \infty\,, \tag{75}$$

where $A_0 = \cap_{t \geqslant 0} \overline{\{\bar{\boldsymbol{\theta}}(s) : s \geqslant t\}}$ which is almost everywhere an internally chain transitive set of the differential inclusion

$$\dot{\boldsymbol{\theta}}(t) \;\in\; \hat{\boldsymbol{h}}(\boldsymbol{\theta}(t)), \tag{76}$$

where $\hat{\boldsymbol{h}}(\boldsymbol{\theta}) = \{\tilde{\boldsymbol{h}}(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\nu}) : \boldsymbol{\nu} \in D^{(w)}(\boldsymbol{\theta}, \boldsymbol{\lambda}(\boldsymbol{\theta}))\}$.

Comments.

- **(C1)** This framework allows to show convergence for gradient descent methods beyond stochastic gradient like for the ADAM procedure where current learning parameters are memorized and updated. The random processes $\boldsymbol{Z}^{(w)}$ and $\boldsymbol{Z}^{(\theta)}$ may track the current learning status for the fast and slow iterate, respectively.
- **(C2)** Stochastic regularization like dropout is covered via the random processes $A^{(w)}$ and $A^{(\theta)}$.
- **(C3)** Similar results have been derived without controlled Markov processes [47].

A2.2 Rate of Convergence of Two-Time Scale Stochastic Approximation Algorithms

A2.2.1 Linear Update Rules

First we consider linear iterates according to the PhD thesis of Konda [31] and Konda & Tsitsiklis [34].

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\Big(\boldsymbol{a}_1 \;-\; \boldsymbol{A}_{11}\,\boldsymbol{\theta}_n \;-\; \boldsymbol{A}_{12}\,\boldsymbol{w}_n \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\;, \tag{77}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\,\Big(\boldsymbol{a}_2 \;-\; \boldsymbol{A}_{21}\,\boldsymbol{\theta}_n \;-\; \boldsymbol{A}_{22}\,\boldsymbol{w}_n \;+\; \boldsymbol{M}_n^{(w)}\Big)\;. \tag{78}$$

Assumptions. We make the following assumptions:

(A1) The random variables $(\boldsymbol{M}_n^{(\theta)}, \boldsymbol{M}_n^{(w)}), n = 0, 1, \ldots$, are independent of $\boldsymbol{w}_0, \boldsymbol{\theta}_0$ and of each other. The have zero mean: $\mathrm{E}[\boldsymbol{M}_n^{(\theta)}] = 0$ and $\mathrm{E}[\boldsymbol{M}_n^{(w)}] = 0$. The covariance is

$$\mathrm{E}\left[\boldsymbol{M}_n^{(\theta)}\,(\boldsymbol{M}_n^{(\theta)})^T\right] \;=\; \boldsymbol{\Gamma}_{11}\;, \tag{79}$$

$$\mathrm{E}\left[\boldsymbol{M}_n^{(\theta)}\,(\boldsymbol{M}_n^{(w)})^T\right] \;=\; \boldsymbol{\Gamma}_{12} \;=\; \boldsymbol{\Gamma}_{21}^T\;, \tag{80}$$

$$\mathrm{E}\left[\boldsymbol{M}_n^{(w)}\,(\boldsymbol{M}_n^{(w)})^T\right] \;=\; \boldsymbol{\Gamma}_{22}\;. \tag{81}$$

(A2) The learning rates are deterministic, positive, nonincreasing and satisfy with $\epsilon \geqslant 0$:

$$\sum_n a(n) \;=\; \infty \quad , \quad \lim_{n\to\infty} a(n) \;=\; 0\;, \tag{82}$$

$$\sum_n b(n) \;=\; \infty \quad , \quad \lim_{n\to\infty} b(n) \;=\; 0\;, \tag{83}$$

$$\frac{a(n)}{b(n)} \;\to\; \epsilon\;. \tag{84}$$

We often consider the case $\epsilon = 0$.

(A3) Convergence of the iterates: We define

$$\boldsymbol{\Delta} \;:=\; \boldsymbol{A}_{11} \;-\; \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\;. \tag{85}$$

A matrix is *Hurwitz* if the real part of each eigenvalue is strictly negative. We assume that the matrices $-\boldsymbol{A}_{22}$ and $-\boldsymbol{\Delta}$ are Hurwitz.

(A4) Convergence rate remains simple:

(a) There exists a constant $\bar{a} \geqslant 0$ such that

$$\lim_n (a(n+1)^{-1} \;-\; a(n)^{-1}) \;=\; \bar{a}\;. \tag{86}$$

(b) If $\epsilon = 0$, then

$$\lim_n (b(n+1)^{-1} \;-\; b(n)^{-1}) \;=\; 0\;. \tag{87}$$

(c) The matrix

$$-\left(\boldsymbol{\Delta} \;-\; \frac{\bar{a}}{2}\,\boldsymbol{I}\right) \tag{88}$$

is Hurwitz.

Rate of Convergence Theorem. The next theorem is taken from Konda [31] and Konda & Tsitsiklis [34].

Let $\boldsymbol{\theta}^* \in \mathbb{R}^m$ and $\boldsymbol{w}^* \in \mathbb{R}^k$ be the unique solution to the system of linear equations

$$\boldsymbol{A}_{11}\,\boldsymbol{\theta}_n \;+\; \boldsymbol{A}_{12}\,\boldsymbol{w}_n \;=\; \boldsymbol{a}_1\;, \tag{89}$$

$$\boldsymbol{A}_{21}\,\boldsymbol{\theta}_n \;+\; \boldsymbol{A}_{22}\,\boldsymbol{w}_n \;=\; \boldsymbol{a}_2\;. \tag{90}$$

For each n, let

$$\hat{\boldsymbol{\theta}}_n \;=\; \boldsymbol{\theta}_n \;-\; \boldsymbol{\theta}^*\;, \tag{91}$$

$$\hat{\boldsymbol{w}}_n \;=\; \boldsymbol{w}_n \;-\; \boldsymbol{A}_{22}^{-1}\;(\boldsymbol{a}_2 \;-\; \boldsymbol{A}_{21}\,\boldsymbol{\theta}_n)\;, \tag{92}$$

$$\boldsymbol{\Sigma}_{11}^n \;=\; \boldsymbol{\theta}_n^{-1}\,\mathrm{E}\left[\hat{\boldsymbol{\theta}}_n\hat{\boldsymbol{\theta}}_n^T\right]\;, \tag{93}$$

$$\boldsymbol{\Sigma}_{12}^n \;=\; \left(\boldsymbol{\Sigma}_{21}^n\right)^T \;=\; \boldsymbol{\theta}_n^{-1}\,\mathrm{E}\left[\hat{\boldsymbol{\theta}}_n\hat{\boldsymbol{w}}_n^T\right]\;, \tag{94}$$

$$\boldsymbol{\Sigma}_{22}^n \;=\; \boldsymbol{w}_n^{-1}\,\mathrm{E}\left[\hat{\boldsymbol{w}}_n\hat{\boldsymbol{w}}_n^T\right]\;, \tag{95}$$

$$\boldsymbol{\Sigma}^n \;=\; \begin{pmatrix}\boldsymbol{\Sigma}_{11}^n & \boldsymbol{\Sigma}_{12}^n\\ \boldsymbol{\Sigma}_{21}^n & \boldsymbol{\Sigma}_{22}^n\end{pmatrix}\;. \tag{96}$$

Theorem 6 (Konda & Tsitsiklis)**.** *Under above assumptions and when the constant ϵ is sufficiently small, the limit matrices*

$$\boldsymbol{\Sigma}_{11}^{(\epsilon)} \;=\; \lim_n \boldsymbol{\Sigma}_{11}^n\;,\quad \boldsymbol{\Sigma}_{12}^{(\epsilon)} \;=\; \lim_n \boldsymbol{\Sigma}_{12}^n\;,\quad \boldsymbol{\Sigma}_{22}^{(\epsilon)} \;=\; \lim_n \boldsymbol{\Sigma}_{22}^n\;. \tag{97}$$

exist. Furthermore, the matrix

$$\boldsymbol{\Sigma}^{(0)} \;=\; \begin{pmatrix}\boldsymbol{\Sigma}_{11}^{(0)} & \boldsymbol{\Sigma}_{12}^{(0)}\\ \boldsymbol{\Sigma}_{21}^{(0)} & \boldsymbol{\Sigma}_{22}^{(0)}\end{pmatrix} \tag{98}$$

is the unique solution to the following system of equations

$$\boldsymbol{\Delta}\,\boldsymbol{\Sigma}_{11}^{(0)} \;+\; \boldsymbol{\Sigma}_{11}^{(0)}\,\boldsymbol{\Delta}^T \;-\; \bar{a}\,\boldsymbol{\Sigma}_{11}^{(0)} \;+\; \boldsymbol{A}_{12}\,\boldsymbol{\Sigma}_{21}^{(0)} \;+\; \boldsymbol{\Sigma}_{12}^{(0)}\,\boldsymbol{A}_{12}^T \;=\; \boldsymbol{\Gamma}_{11}\;, \tag{99}$$

$$\boldsymbol{A}_{12}\,\boldsymbol{\Sigma}_{22}^{(0)} \;+\; \boldsymbol{\Sigma}_{12}^{(0)}\,\boldsymbol{A}_{22}^T \;=\; \boldsymbol{\Gamma}_{12}\;, \tag{100}$$

$$\boldsymbol{A}_{22}\,\boldsymbol{\Sigma}_{22}^{(0)} \;+\; \boldsymbol{\Sigma}_{22}^{(0)}\,\boldsymbol{A}_{22}^T \;=\; \boldsymbol{\Gamma}_{22}\;. \tag{101}$$

Finally,

$$\lim_{\epsilon\downarrow 0} \boldsymbol{\Sigma}_{11}^{(\epsilon)} \;=\; \boldsymbol{\Sigma}_{11}^{(0)}\;,\quad \lim_{\epsilon\downarrow 0} \boldsymbol{\Sigma}_{12}^{(\epsilon)} \;=\; \boldsymbol{\Sigma}_{12}^{(0)}\;,\quad \lim_{\epsilon\downarrow 0} \boldsymbol{\Sigma}_{22}^{(\epsilon)} \;=\; \boldsymbol{\Sigma}_{22}^{(0)}\;. \tag{102}$$

The next theorems shows that the asymptotic covariance matrix of $a(n)^{-1/2}\boldsymbol{\theta}_n$ is the same as that of $a(n)^{-1/2}\bar{\boldsymbol{\theta}}_n$, where $\bar{\boldsymbol{\theta}}_n$ evolves according to the single-time-scale stochastic iteration:

$$\bar{\boldsymbol{\theta}}_{n+1} \;=\; \bar{\boldsymbol{\theta}}_n \;+\; a(n)\,\left(\boldsymbol{a}_1 \;-\; \boldsymbol{A}_{11}\,\bar{\boldsymbol{\theta}}_n \;-\; \boldsymbol{A}_{12}\,\bar{\boldsymbol{w}}_n \;+\; \boldsymbol{M}_n^{(\theta)}\right)\;, \tag{103}$$

$$\boldsymbol{0} \;=\; \boldsymbol{a}_2 \;-\; \boldsymbol{A}_{21}\,\bar{\boldsymbol{\theta}}_n \;-\; \boldsymbol{A}_{22}\,\bar{\boldsymbol{w}}_n \;+\; \boldsymbol{M}_n^{(w)}\;. \tag{104}$$

The next theorem combines Theorem 2.8 of Konda & Tsitsiklis and Theorem 4.1 of Konda & Tsitsiklis:

Theorem 7 (Konda & Tsitsiklis 2nd)**.** *Under above assumptions*

$$\boldsymbol{\Sigma}_{11}^{(0)} \;=\; \lim_n a(n)^{-1}\,\mathrm{E}\left[\bar{\boldsymbol{\theta}}_n\bar{\boldsymbol{\theta}}_n^T\right]\;. \tag{105}$$

If the assumptions hold with $\epsilon = 0$, then $a(n)^{-1/2}\hat{\boldsymbol{\theta}}_n$ converges in distribution to $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{11}^{(0)})$.

Comments.

(C1) In his PhD thesis [31] Konda extended the analysis to the nonlinear case. Konda makes a linearization of the nonlinear function $\boldsymbol{h}$ and $\boldsymbol{g}$ with

$$\boldsymbol{A}_{11} \;=\; -\,\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}}\,,\;\; \boldsymbol{A}_{12} \;=\; -\,\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}}\,,\;\; \boldsymbol{A}_{21} \;=\; -\,\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\theta}}\,,\;\; \boldsymbol{A}_{22} \;=\; -\,\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{w}}\,. \tag{106}$$

There are additional errors due to linearization which have to be considered. However only a sketch of a proof is provided but not a complete proof.

(C2) Theorem 4.1 of Konda & Tsitsiklis is important to generalize to the nonlinear case.

(C3) The convergence rate is governed by $\boldsymbol{A}_{22}$ for the fast and $\boldsymbol{\Delta}$ for the slow iterate. $\boldsymbol{\Delta}$ in turn is affected by the interaction effects captured by $\boldsymbol{A}_{21}$ and $\boldsymbol{A}_{12}$ together with the inverse of $\boldsymbol{A}_{22}$.

A2.2.2 Nonlinear Update Rules

The rate of convergence for nonlinear update rules according to Mokkadem & Pelletier is considered [40].

The iterates are

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\Big(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{Z}_n^{(\theta)} \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\,, \tag{107}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; b(n)\,\Big(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{Z}_n^{(w)} \;+\; \boldsymbol{M}_n^{(w)}\Big)\,. \tag{108}$$

with the increasing σ-fields

$$\mathcal{F}_n \;=\; \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, \boldsymbol{Z}_l^{(\theta)}, \boldsymbol{Z}_l^{(w)}, l \leqslant n),\; n \geqslant 0\,. \tag{109}$$

The terms $\boldsymbol{Z}_n^{(\theta)}$ and $\boldsymbol{Z}_n^{(w)}$ can be used to address the error through linearization, that is, the difference of the nonlinear functions to their linear approximation.

Assumptions. We make the following assumptions:

(A1) Convergence is ensured:

$$\lim_{n\to\infty} \boldsymbol{\theta}_n \;=\; \boldsymbol{\theta}^* \;\text{ a.s.}\,, \tag{110}$$

$$\lim_{n\to\infty} \boldsymbol{w}_n \;=\; \boldsymbol{w}^* \;\text{ a.s.}\,. \tag{111}$$

(A2) Linear approximation and Hurwitz:
There exists a neighborhood $\mathcal{U}$ of $(\boldsymbol{\theta}^*, \boldsymbol{w}^*)$ such that, for all $(\boldsymbol{\theta}, \boldsymbol{w}) \in \mathcal{U}$

$$\begin{pmatrix} \boldsymbol{h}\big(\boldsymbol{\theta}, \boldsymbol{w}\big) \\ \boldsymbol{g}\big(\boldsymbol{\theta}, \boldsymbol{w}\big) \end{pmatrix} \;=\; \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \;-\; \boldsymbol{\theta}^* \\ \boldsymbol{w} \;-\; \boldsymbol{w}^* \end{pmatrix} \;+\; \mathrm{O}\left(\left\| \begin{matrix} \boldsymbol{\theta} \;-\; \boldsymbol{\theta}^* \\ \boldsymbol{w} \;-\; \boldsymbol{w}^* \end{matrix} \right\|^2 \right)\,. \tag{112}$$

We define

$$\boldsymbol{\Delta} \;:=\; \boldsymbol{A}_{11} \;-\; \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}\,. \tag{113}$$

A matrix is *Hurwitz* if the real part of each eigenvalue is strictly negative. We assume that the matrices $\boldsymbol{A}_{22}$ and $\boldsymbol{\Delta}$ are Hurwitz.

(A3) Assumptions on the learning rates:

$$a(n) \;=\; a_0\, n^{-\alpha} \tag{114}$$

$$b(n) \;=\; b_0\, n^{-\beta}\,, \tag{115}$$

where $a_0 > 0$ and $b_0 > 0$ and $1/2 < \beta < \alpha \leqslant 1$. If $\alpha = 1$, then $a_0 > 1/(2e_{\min})$ with $e_{\min}$ as the absolute value of the largest eigenvalue of $\boldsymbol{\Delta}$ (the eigenvalue closest to 0).

(A4) Assumptions on the noise and error:

(a) martingale difference sequences:

$$\mathrm{E}\left[\boldsymbol{M}_{n+1}^{(\theta)} \mid \mathcal{F}_n\right] \;=\; 0 \;\text{ a.s.}\,, \tag{116}$$

$$\mathrm{E}\left[\boldsymbol{M}_{n+1}^{(w)} \mid \mathcal{F}_n\right] \;=\; 0 \;\text{ a.s.}\,. \tag{117}$$

(b) existing second moments:

$$\lim_{n\to\infty} \mathrm{E}\left[\begin{pmatrix}\boldsymbol{M}_{n+1}^{(\theta)}\\ \boldsymbol{M}_{n+1}^{(w)}\end{pmatrix}\left((\boldsymbol{M}_{n+1}^{(\theta)})^T \quad (\boldsymbol{M}_{n+1}^{(w)})^T\right) \mid \mathcal{F}_n\right] \;=\; \boldsymbol{\Gamma} \;=\; \begin{pmatrix}\boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12}\\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22}\end{pmatrix} \;\text{ a.s.} \tag{118}$$

(c) bounded moments:
There exist $l > 2/\beta$ such that

$$\sup_n \mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(\theta)}\|^l \mid \mathcal{F}_n\right] \;<\; \infty \;\text{ a.s.}\,, \tag{119}$$

$$\sup_n \mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(w)}\|^l \mid \mathcal{F}_n\right] \;<\; \infty \;\text{ a.s.} \tag{120}$$

(d) bounded error:

$$\boldsymbol{Z}_n^{(\theta)} \;=\; r_n^{(\theta)} \;+\; \mathrm{O}\big(\|\boldsymbol{\theta} \;-\; \boldsymbol{\theta}^*\|^2 \;+\; \|\boldsymbol{w} \;-\; \boldsymbol{w}^*\|^2\big)\,, \tag{121}$$

$$\boldsymbol{Z}_n^{(w)} \;=\; r_n^{(w)} \;+\; \mathrm{O}\big(\|\boldsymbol{\theta} \;-\; \boldsymbol{\theta}^*\|^2 \;+\; \|\boldsymbol{w} \;-\; \boldsymbol{w}^*\|^2\big)\,, \tag{122}$$

with

$$\|r_n^{(\theta)}\| \;+\; \|r_n^{(w)}\| \;=\; \mathrm{o}(\sqrt{a(n)}) \;\text{ a.s.} \tag{123}$$

Rate of Convergence Theorem. We report a theorem and a proposition from Mokkadem & Pelletier [40]. However first we have to define the covariance matrices $\boldsymbol{\Sigma}_\theta$ and $\boldsymbol{\Sigma}_w$ which govern the rate of convergence.

First we define

$$\boldsymbol{\Gamma}_\theta \;:=\; \lim_{n\to\infty} \mathrm{E}\left[\left(\boldsymbol{M}_{n+1}^{(\theta)} \;-\; \boldsymbol{A}_{12}\,\boldsymbol{A}_{22}^{-1}\,\boldsymbol{M}_{n+1}^{(w)}\right)\left(\boldsymbol{M}_{n+1}^{(\theta)} \;-\; \boldsymbol{A}_{12}\,\boldsymbol{A}_{22}^{-1}\,\boldsymbol{M}_{n+1}^{(w)}\right)^T \mid \mathcal{F}_n\right] \;= \tag{124}$$

$$\boldsymbol{\Gamma}_{11} \;+\; \boldsymbol{A}_{12}\,\boldsymbol{A}_{22}^{-1}\,\boldsymbol{\Gamma}_{22}\,(\boldsymbol{A}_{22}^{-1})^T\,\boldsymbol{A}_{12}^T \;-\; \boldsymbol{\Gamma}_{12}(\boldsymbol{A}_{22}^{-1})^T\,\boldsymbol{A}_{12}^T \;-\; \boldsymbol{A}_{12}\,\boldsymbol{A}_{22}^{-1}\,\boldsymbol{\Gamma}_{21}\,.$$

We now define the asymptotic covariance matrices $\boldsymbol{\Sigma}_\theta$ and $\boldsymbol{\Sigma}_w$:

$$\boldsymbol{\Sigma}_\theta \;=\; \int_0^\infty \exp\left(\left(\boldsymbol{\Delta} \;+\; \frac{\mathbb{1}_{a=1}}{2\,a_0}\,\boldsymbol{I}\right)\,t\right)\,\boldsymbol{\Gamma}_\theta\,\exp\left(\left(\boldsymbol{\Delta}^T \;+\; \frac{\mathbb{1}_{a=1}}{2\,a_0}\,\boldsymbol{I}\right)\,t\right)\,\mathrm{d}t\,, \tag{125}$$

$$\boldsymbol{\Sigma}_w \;=\; \int_0^\infty \exp\left(\boldsymbol{A}_{22}\,t\right)\,\boldsymbol{\Gamma}_{22}\,\exp\left(\boldsymbol{A}_{22}\,t\right)\,\mathrm{d}t\,. \tag{126}$$

$\boldsymbol{\Sigma}_\theta$ and $\boldsymbol{\Sigma}_w$ are solutions of the Lyapunov equations:

$$\left(\boldsymbol{\Delta} \;+\; \frac{\mathbb{1}_{a=1}}{2\,a_0}\,\boldsymbol{I}\right)\,\boldsymbol{\Sigma}_\theta \;+\; \boldsymbol{\Sigma}_\theta\,\left(\boldsymbol{\Delta}^T \;+\; \frac{\mathbb{1}_{a=1}}{2\,a_0}\,\boldsymbol{I}\right) \;=\; -\,\boldsymbol{\Gamma}_\theta\,, \tag{127}$$

$$\boldsymbol{A}_{22}\,\boldsymbol{\Sigma}_w \;+\; \boldsymbol{\Sigma}_w\,\boldsymbol{A}_{22}^T \;=\; -\,\boldsymbol{\Gamma}_{22}\,. \tag{128}$$

Theorem 8 (Mokkadem & Pelletier: Joint weak convergence)**.** *Under above assumptions:*

$$\begin{pmatrix}\sqrt{a(n)^{-1}}\,(\boldsymbol{\theta} \;-\; \boldsymbol{\theta}^*)\\ \sqrt{b(n)^{-1}}\,(\boldsymbol{w} \;-\; \boldsymbol{w}^*)\end{pmatrix} \;\xrightarrow{\mathcal{D}}\; \mathcal{N}\left(\boldsymbol{0}\,,\,\begin{pmatrix}\boldsymbol{\Sigma}_\theta & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}_w\end{pmatrix}\right)\,. \tag{129}$$

Theorem 9 (Mokkadem & Pelletier: Strong convergence)**.** *Under above assumptions:*

$$\|\boldsymbol{\theta} \;-\; \boldsymbol{\theta}^*\| \;=\; \mathrm{O}\left(\sqrt{a(n)\;\log\left(\sum_{l=1}^n a(l)\right)}\right) \;\; \textit{a.s.}\,, \tag{130}$$

$$\|\boldsymbol{w} \;-\; \boldsymbol{w}^*\| \;=\; \mathrm{O}\left(\sqrt{b(n)\;\log\left(\sum_{l=1}^n b(l)\right)}\right) \;\; \textit{a.s.} \tag{131}$$

Comments.

(C1) Besides the learning steps $a(n)$ and $b(n)$, the convergence rate is governed by $\boldsymbol{A}_{22}$ for the fast and $\boldsymbol{\Delta}$ for the slow iterate. $\boldsymbol{\Delta}$ in turn is affected by interaction effects which are captured by $\boldsymbol{A}_{21}$ and $\boldsymbol{A}_{12}$ together with the inverse of $\boldsymbol{A}_{22}$.

A2.3 Equal Time Scale Stochastic Approximation Algorithms

In this subsection we consider the case when the learning rates have equal time scale.

A2.3.1 Equal Time Scale for Saddle Point Iterates

If equal time scales assumed then the iterates revisit infinite often an environment of the solution [54]. In Zhang 2007, the functions of the iterates are the derivatives of a Lagrangian with respect to the dual and primal variables [54]. The iterates are

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\Big(\boldsymbol{h}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{Z}_n^{(\theta)} \;+\; \boldsymbol{M}_n^{(\theta)}\Big)\;, \tag{132}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; a(n)\,\Big(\boldsymbol{g}\big(\boldsymbol{\theta}_n, \boldsymbol{w}_n\big) \;+\; \boldsymbol{Z}_n^{(w)} \;+\; \boldsymbol{M}_n^{(w)}\Big)\;. \tag{133}$$

with the increasing σ-fields

$$\mathcal{F}_n \;=\; \sigma(\boldsymbol{\theta}_l, \boldsymbol{w}_l, \boldsymbol{M}_l^{(\theta)}, \boldsymbol{M}_l^{(w)}, \boldsymbol{Z}_l^{(\theta)}, \boldsymbol{Z}_l^{(w)}, l \leqslant n),\; n \geqslant 0\;. \tag{134}$$

The terms $\boldsymbol{Z}_n^{(\theta)}$ and $\boldsymbol{Z}_n^{(w)}$ subsum biased estimation errors.

Assumptions. We make the following assumptions:

(A1) Assumptions on update function: $\boldsymbol{h}$ and $\boldsymbol{g}$ are continuous, differentiable, and bounded. The Jacobians

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{w}} \quad \text{and} \quad \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\theta}} \tag{135}$$

are Hurwitz. A matrix is *Hurwitz* if the real part of each eigenvalue is strictly negative. This assumptions corresponds to the assumption in [54] that the Lagrangian is concave in $\boldsymbol{w}$ and convex in $\boldsymbol{\theta}$.

(A2) Assumptions on noise:

$\{\boldsymbol{M}_n^{(\theta)}\}$ and $\{\boldsymbol{M}_n^{(w)}\}$ are a martingale difference sequences w.r.t. the increasing σ-fields $\mathcal{F}_n$. Furthermore they are mutually independent.

Bounded second moment:

$$\mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(\theta)}\|^2 \mid \mathcal{F}_n\right] \;<\; \infty \;\text{ a.s.}\,, \tag{136}$$

$$\mathrm{E}\left[\|\boldsymbol{M}_{n+1}^{(w)}\|^2 \mid \mathcal{F}_n\right] \;<\; \infty \;\text{ a.s.}\,. \tag{137}$$

(A3) Assumptions on the learning rate:

$$a(n) \;>\; 0 \quad , \quad a(n) \;\to\; 0 \quad , \quad \sum_n a(n) \;=\; \infty \quad , \quad \sum_n a^2(n) \;<\; \infty\,. \tag{138}$$

(A4) Assumption on the biased error:

Boundedness:

$$\limsup_n \|\boldsymbol{Z}_n^{(\theta)}\| \;\leqslant\; \alpha^{(\theta)} \text{ a.s.} \tag{139}$$

$$\limsup_n \|\boldsymbol{Z}_n^{(w)}\| \;\leqslant\; \alpha^{(w)} \text{ a.s.} \tag{140}$$

Theorem. Define the "contraction region" A_η as follows:

$$A_\eta \;=\; \{(\boldsymbol{\theta}, \boldsymbol{w}) : \alpha^{(\theta)} \geqslant \eta \, \|\boldsymbol{h}(\boldsymbol{\theta}, \boldsymbol{w})\| \quad \text{or} \quad \alpha^{(w)} \geqslant \eta \, \|\boldsymbol{g}(\boldsymbol{\theta}, \boldsymbol{w})\|, \; 0 \leqslant \eta < 1\} \; . \tag{141}$$

Theorem 10 (Zhang)**.** *Under above assumptions the iterates return to A_η infinitely often with probability one (a.s.).*

Comments.

- **(C1)** The proof of the theorem in [54] does not use the saddle point condition and not the fact that the functions of the iterates are derivatives of the same function.
- **(C2)** For the unbiased case, Zhang showed in Theorem 3.1 of [54] that the iterates converge. However he used the saddle point condition of the Lagrangian. He considered iterates with functions that are the derivatives of a Lagrangian with respect to the dual and primal variables [54].

A2.3.2 Equal Time Step for Actor-Critic Method

If equal time scales assumed then the iterates revisit infinite often an environment of the solution of DiCastro & Meir [13]. The iterates of DiCastro & Meir are derived for actor-critic learning.

To present the actor-critic update iterates, we have to define some functions and terms. $\boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x}, \boldsymbol{\theta})$ is the policy function parametrized by $\boldsymbol{\theta} \in \mathbb{R}^m$ with observations $\boldsymbol{x} \in \mathcal{X}$ and actions $\boldsymbol{u} \in \mathcal{U}$. A Markov chain given by $\mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{u})$ gives the next observation $\boldsymbol{y}$ using the observation $\boldsymbol{x}$ and the action $\boldsymbol{u}$. In each state $\boldsymbol{x}$ the agent receives a reward $r(\boldsymbol{x})$.

The average reward per stage is for the recurrent state $\boldsymbol{x}^*$:

$$\tilde{\eta}(\boldsymbol{\theta}) \;=\; \lim_{T \to \infty} \mathrm{E}\left[\frac{1}{T} \sum_{n=0}^{T-1} r(\boldsymbol{x}_n) \mid \boldsymbol{x}_0 = \boldsymbol{x}^*, \boldsymbol{\theta} \right] \; . \tag{142}$$

The estimate of $\tilde{\eta}$ is denoted by η.

The differential value function is

$$\tilde{h}(\boldsymbol{x}, \boldsymbol{\theta}) \;=\; \mathrm{E}\left[\sum_{n=0}^{T-1} (r(\boldsymbol{x}_n) \;-\; \tilde{\eta}(\boldsymbol{\theta})) \mid \boldsymbol{x}_0 = \boldsymbol{x}, \boldsymbol{\theta} \right] \; . \tag{143}$$

The temporal difference is

$$\tilde{d}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) \;=\; r(\boldsymbol{x}) \;-\; \tilde{\eta}(\boldsymbol{\theta}) \;+\; \tilde{h}(\boldsymbol{y}, \boldsymbol{\theta}) \;-\; \tilde{h}(\boldsymbol{x}, \boldsymbol{\theta}) \; . \tag{144}$$

The estimate of $\tilde{d}$ is denoted by d.

The likelihood ratio derivative $\boldsymbol{\Psi} \in \mathbb{R}^m$ is

$$\boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \;=\; \frac{\nabla_\theta \boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x}, \boldsymbol{\theta})}{\boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x}, \boldsymbol{\theta})} \; . \tag{145}$$

The value function $\tilde{h}$ is approximated by

$$h(\boldsymbol{x}, \boldsymbol{w}) \;=\; \boldsymbol{\phi}(\boldsymbol{x})^T \, \boldsymbol{w} \; , \tag{146}$$

where $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^k$. We define $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{X}| \times k}$

$$\boldsymbol{\Phi} \;=\; \begin{pmatrix} \boldsymbol{\phi}_1(\boldsymbol{x}_1) & \boldsymbol{\phi}_2(\boldsymbol{x}_1) & \dots & \boldsymbol{\phi}_k(\boldsymbol{x}_1) \\ \boldsymbol{\phi}_1(\boldsymbol{x}_2) & \boldsymbol{\phi}_2(\boldsymbol{x}_2) & \dots & \boldsymbol{\phi}_k(\boldsymbol{x}_2) \\ \vdots & \vdots & & \vdots \\ \boldsymbol{\phi}_1(\boldsymbol{x}_{|\mathcal{X}|}) & \boldsymbol{\phi}_2(\boldsymbol{x}_{|\mathcal{X}|}) & \dots & \boldsymbol{\phi}_k(\boldsymbol{x}_{|\mathcal{X}|}) \end{pmatrix} \tag{147}$$

and

$$h(\boldsymbol{w}) \;=\; \boldsymbol{\Phi}\,\boldsymbol{w}\;. \tag{148}$$

For TD(λ) we have an eligibility trace:

$$e_n \;=\; \lambda\, e_{n-1} \;+\; \boldsymbol{\phi}(\boldsymbol{x}_n)\;. \tag{149}$$

We define the approximation error with optimal parameter $\boldsymbol{w}^*(\boldsymbol{\theta})$:

$$\epsilon_{\text{app}}(\boldsymbol{\theta}) \;=\; \inf_{\boldsymbol{w}\in\mathbb{R}^k} \|\tilde{h}(\boldsymbol{\theta}) \;-\; \boldsymbol{\Phi}\,\boldsymbol{w}\|_{\pi(\boldsymbol{\theta})} \;=\; \|\tilde{h}(\boldsymbol{\theta}) \;-\; \boldsymbol{\Phi}\,\boldsymbol{w}^*(\boldsymbol{\theta})\|_{\pi(\boldsymbol{\theta})}\;, \tag{150}$$

where $\pi(\boldsymbol{\theta})$ is an projection operator into the span of $\boldsymbol{\Phi}\boldsymbol{w}$. We bound this error by

$$\epsilon_{\text{app}} \;=\; \sup_{\boldsymbol{\theta}\in\mathbb{R}^k} \epsilon_{\text{app}}(\boldsymbol{\theta})\;. \tag{151}$$

We denoted by $\tilde{\eta}$, $\tilde{d}$, and $\tilde{h}$ the exact functions and used for their approximation η, d, and h, respectively. We have learning rate adjustments Γ_η and Γ_w for the critic.

The update rules are:

Critic:

$$\eta_{n+1} \;=\; \eta_n \;+\; a(n)\,\Gamma_\eta\,(r(\boldsymbol{x}_n) \;-\; \eta_n)\;, \tag{152}$$

$$h(\boldsymbol{x},\boldsymbol{w}_n) \;=\; \boldsymbol{\phi}(\boldsymbol{x})^T\,\boldsymbol{w}_n\;, \tag{153}$$

$$d(\boldsymbol{x}_n,\boldsymbol{x}_{n+1},\boldsymbol{w}_n) \;=\; r(\boldsymbol{x}_n) \;-\; \eta_n \;+\; h(\boldsymbol{x}_{n+1},\boldsymbol{w}_n) \;-\; h(\boldsymbol{x}_n,\boldsymbol{w}_n)\;, \tag{154}$$

$$e_n \;=\; \lambda\, e_{n-1} \;+\; \boldsymbol{\phi}(\boldsymbol{x}_n)\;, \tag{155}$$

$$\boldsymbol{w}_{n+1} \;=\; \boldsymbol{w}_n \;+\; a(n)\,\Gamma_w\,d(\boldsymbol{x}_n,\boldsymbol{x}_{n+1},\boldsymbol{w}_n)\,e_n\;. \tag{156}$$

Actor:

$$\boldsymbol{\theta}_{n+1} \;=\; \boldsymbol{\theta}_n \;+\; a(n)\,\boldsymbol{\Psi}(\boldsymbol{x}_n,\boldsymbol{u}_n,\boldsymbol{\theta}_n)\,d(\boldsymbol{x}_n,\boldsymbol{x}_{n+1},\boldsymbol{w}_n)\;. \tag{157}$$

Assumptions. We make the following assumptions:

- **(A1)** Assumption on rewards:
  The rewards $\{r(\boldsymbol{x})\}_{\boldsymbol{x}\in\mathcal{X}}$ are uniformly bounded by a finite constant B_r.
- **(A2)** Assumption on the Markov chain:
  Each Markov chain for each $\boldsymbol{\theta}$ is aperiodic, recurrent, and irreducible.
- **(A3)** Assumptions on the policy function:
  The conditional probability function $\boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x},\boldsymbol{\theta})$ is twice differentiable. Moreover, there exist positive constants, B_{μ_1} and B_{μ_2}, such that for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{u} \in \mathcal{U}$, $\boldsymbol{\theta} \in \mathbb{R}^m$ and $1 \leqslant l_1, l_2 \leqslant m$ we have
  $$\left\|\frac{\partial\boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x},\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_l}\right\| \;\leqslant\; B_{\mu_1}\;, \quad \left\|\frac{\partial^2\boldsymbol{\mu}(\boldsymbol{u} \mid \boldsymbol{x},\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_{l_1}\,\partial\boldsymbol{\theta}_{l_2}}\right\| \;\leqslant\; B_{\mu_2}\;. \tag{158}$$
- **(A4)** Assumption on the likelihood ratio derivative:
  For all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{u} \in \mathcal{U}$, and $\boldsymbol{\theta} \in \mathbb{R}^m$, there exists a positive constant B_Ψ, such that
  $$\|\boldsymbol{\Psi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\theta})\|_2 \;\leqslant\; B_\Psi \;<\; \infty\;, \tag{159}$$
  where $\|.\|_2$ is the Euclidean L_2 norm.
- **(A5)** Assumptions on the approximation space given by $\boldsymbol{\Phi}$:
  The columns of the matrix $\boldsymbol{\Phi}$ are independent, that is, the form a basis of dimension k. The norms of the columns vectors of the matrix $\boldsymbol{\Phi}$ are bounded above by 1, that is, $\|\boldsymbol{\phi}_l\|_2 \leqslant 1$ for $1 \leqslant l \leqslant k$.
- **(A6)** Assumptions on the learning rate:
  $$\sum_n a(n) \;=\; \infty \quad , \quad \sum_n a^2(n) \;<\; \infty\;. \tag{160}$$

Theorem. The algorithm converged if $\nabla_\theta \tilde{\eta}(\boldsymbol{\theta}) = \mathbf{0}$, since the actor reached a stationary point where the updates are zero. We assume that $\|\nabla_\theta \tilde{\eta}(\boldsymbol{\theta})\|$ hints at how close we are to the convergence point.

The next theorem from DiCastro & Meir [13] implies that the trajectory visits a neighborhood of a local maximum infinitely often. Although it may leave the local vicinity of the maximum, it is guaranteed to return to it infinitely often.

Theorem 11 (DiCastro & Meir)**.** *Define*

$$B_{\nabla\tilde{\eta}} \;=\; \frac{B_{\Delta td1}}{\Gamma_w} \;+\; \frac{B_{\Delta td2}}{\Gamma_\eta} \;+\; B_{\Delta td3}\,\epsilon_{\text{app}}\;, \tag{161}$$

where $B_{\Delta td1}$, $B_{\Delta td2}$, and $B_{\Delta td3}$ are finite constants depending on the Markov decision process and the agent parameters.

Under above assumptions

$$\lim_{t\to\infty} \inf \;\|\nabla_\theta \tilde{\eta}(\boldsymbol{\theta}_t)\| \;\leqslant\; B_{\nabla\tilde{\eta}}\;. \tag{162}$$

The trajectory visits a neighborhood of a local maximum infinitely often.

Comments.

- **(C1)** The larger the critic learning rates Γ_w and Γ_η are, the smaller is the region around the local maximum.
- **(C2)** The results are in agreement with those of Zhang 2007 [54].
- **(C3)** Even if the results are derived for a special actor-critic setting, they carry over to a more general setting of the iterates.

A3 ADAM Optimization as Stochastic Heavy Ball with Friction

The Nesterov Accelerated Gradient Descent (NAGD) [43] has raised considerable interest due to its numerical simplicity and its low complexity. Previous to NAGD there was Polyak's Heavy Ball method [44]. The idea of the Heavy Ball is a ball that evolves over the graph of a function f with damping (due to friction) and acceleration. Therefore this is a second-order dynamical system that can be described by the ODE for a Heavy Ball with Friction (HBF) [16]. HBF can overshoot a local minimum and find a neighboring minium [3]. Therefore, HBF has some exploratory properties via the ball's motion and is a step towards global optimization [20]. See Figure A39

GANs suffer from "mode collapsing" where large masses of probability are mapped onto a few modes that cover only small regions. While these regions represent meaningful samples, the variety of the real world data is lost and only few prototype samples are generated. Different methods have been proposed to avoid mode collapsing [10, 39]. We obviate model collapsing by using Adam stochastic approximation [30]. Adam can be described as Heavy Ball with Friction (HBF) (see below), since it averages over past gradients. This averaging corresponds to a velocity that makes a generator trained with Adam more robust to discriminator signals that tend to push its probability mass into small regions. Adam as an HBF method can overshoot local minima that correspond to model collapse and find broader minima. Next, we analyze whether GANs trained with TTUR converge when using Adam.

We recapitulate the Adam update rule at step n, with learning rate a, exponential averaging factors β_1 for the first and β_2 for the second moment of the gradient $\nabla f(\boldsymbol{\theta}_{n-1})$:

$$\begin{aligned}
\boldsymbol{g}_n \;&\longleftarrow\; \nabla f(\boldsymbol{\theta}_{n-1}) \\
\boldsymbol{m}_n \;&\longleftarrow\; (\beta_1/(1-\beta_1^n))\,\boldsymbol{m}_{n-1} \;+\; ((1-\beta_1)/(1-\beta_1^n))\,\boldsymbol{g}_n \\
\boldsymbol{v}_n \;&\longleftarrow\; (\beta_2/(1-\beta_2^n))\,\boldsymbol{v}_{n-1} \;+\; ((1-\beta_2)/(1-\beta_2^n))\,\boldsymbol{g}_n \odot \boldsymbol{g}_n \\
\boldsymbol{\theta}_n \;&\longleftarrow\; \boldsymbol{\theta}_{n-1} \;-\; a\,\boldsymbol{m}_n/(\sqrt{\boldsymbol{v}_n}+\epsilon)\;,
\end{aligned} \tag{163}$$

where following operations are meant componentwise: the product $\odot$, the square root $\sqrt{.}$, and the division $/$ in the last line.

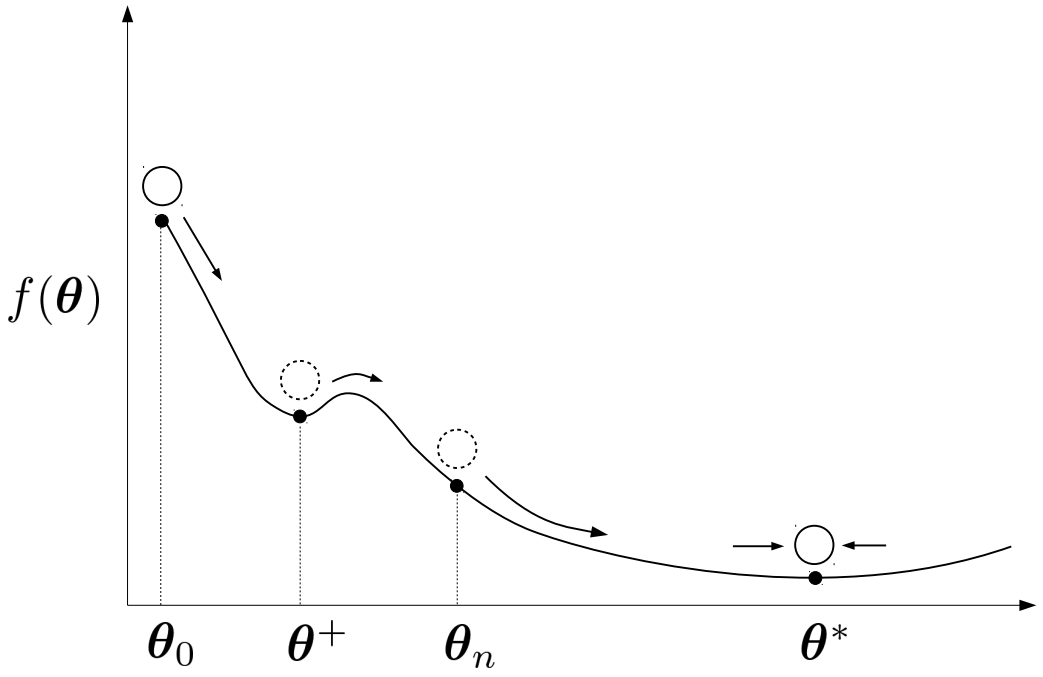

Figure A39: Heavy Ball with Friction, where the ball with mass overshoots the local minimum $\boldsymbol{\theta}^+$ and settles at the flat minimum $\boldsymbol{\theta}^*$.

Instead of learning rate a, we introduce the damping coefficient $a(n)$ with $a(n) = an^{-\tau}$ for $\tau \in (0, 1]$. Adam has parameters β_1 for averaging the gradient and β_2 for averaging the squared gradient. These parameters can be considered as defining a memory for Adam. To characterize β_1 and β_2 in the following, we define the exponential memory $r(n) = r$ and the polynomial memory $r(n) = r/\sum_{l=1}^{n} a(l)$ for some positive constant r. The next theorem describes Adam by a differential equation, which in turn allows to apply the idea of (T, δ) perturbed ODEs to TTUR. Consequently, learning GANs with TTUR and Adam converges.

Theorem 12. *If Adam is used with $\beta_1 = 1 - a(n+1)r(n)$, $\beta_2 = 1 - \alpha a(n+1)r(n)$ and with ∇f as the full gradient of the lower bounded, continuously differentiable objective f, then for stationary second moments of the gradient, Adam follows the differential equation for Heavy Ball with Friction (HBF):*

$$\ddot{\boldsymbol{\theta}}_t \ + \ a(t)\, \dot{\boldsymbol{\theta}}_t \ + \ \nabla f(\boldsymbol{\theta}_t) \ = \ \mathbf{0} \, . \tag{164}$$

Adam converges for gradients ∇f that are L-Lipschitz.

Proof. Gadat et al. derived a discrete and stochastic version of Polyak's Heavy Ball method [44], the Heavy Ball with Friction (HBF) [16]:

$$\begin{aligned} \boldsymbol{\theta}_{n+1} \ &= \ \boldsymbol{\theta}_n \ - \ a(n+1)\, \boldsymbol{m}_n \, , \\ \boldsymbol{m}_{n+1} \ &= \ \big(1 \ - \ a(n+1)\, r(n)\big)\, \boldsymbol{m}_n \ + \ a(n+1)\, r(n)\, \big(\nabla f(\boldsymbol{\theta}_n) \ + \ \boldsymbol{M}_{n+1}\big) \, . \end{aligned} \tag{165}$$

These update rules are the first moment update rules of Adam [30]. The HBF can be formulated as the differential equation Eq. (164) [16]. Gadat et al. showed that the update rules Eq. (165) converge for loss functions f with at most quadratic grow and stated that convergence can be proofed for ∇f that are L-Lipschitz [16]. Convergence has been proved for continuously differentiable f that is quasiconvex (Theorem 3 in Goudou & Munier [20]). Convergence has been proved for ∇f that is L-Lipschitz and bounded from below (Theorem 3.1 in Attouch et al. [3]).

Adam normalizes the average $\boldsymbol{m}_n$ by the second moments $\boldsymbol{v}_n$ of of the gradient $\boldsymbol{g}_n$: $\boldsymbol{v}_n = \mathrm{E}\left[\boldsymbol{g}_n \odot \boldsymbol{g}_n\right]$. $\boldsymbol{m}_n$ is componentwise divided by the square root of the components of $\boldsymbol{v}_n$. We assume that the second moments of $\boldsymbol{g}_n$ are stationary, i.e., $\boldsymbol{v} = \mathrm{E}\left[\boldsymbol{g}_n \odot \boldsymbol{g}_n\right]$. In this case the normalization can be considered as additional noise since the normalization factor randomly deviates from its mean. In the HBF interpretation the normalization by $\sqrt{\boldsymbol{v}}$ corresponds to introducing gravitation. We obtain

$$\boldsymbol{v}_n \;=\; \frac{1 \,-\, \beta_2}{1 \,-\, \beta_2^n} \sum_{l=1}^{n} \beta_2^{n-l} \; \boldsymbol{g}_l \odot \boldsymbol{g}_l \;, \quad \Delta \boldsymbol{v}_n \;=\; \boldsymbol{v}_n \,-\, \boldsymbol{v} \;=\; \frac{1 \,-\, \beta_2}{1 \,-\, \beta_2^n} \sum_{l=1}^{n} \beta_2^{n-l} \; \left(\boldsymbol{g}_l \odot \boldsymbol{g}_l \,-\, \boldsymbol{v}\right) \;. \tag{166}$$

For a stationary second moments $\boldsymbol{v}$ and $\beta_2 = 1 - \alpha a(n+1)r(n)$, we have $\Delta \boldsymbol{v}_n \propto a(n+1)r(n)$. We use a componentwise linear approximation to Adam's second moment normalization $1/\sqrt{\boldsymbol{v} + \Delta \boldsymbol{v}_n} \approx 1/\sqrt{\boldsymbol{v}} - (1/(2\boldsymbol{v} \odot \sqrt{\boldsymbol{v}})) \odot \Delta \boldsymbol{v}_n + \mathrm{O}(\Delta^2 \boldsymbol{v}_n)$, where all operations are meant componentwise. If we set $\boldsymbol{M}_{n+1}^{(v)} = -(\boldsymbol{m}_n \odot \Delta \boldsymbol{v}_n)/(2\boldsymbol{v} \odot \sqrt{\boldsymbol{v}} a(n+1)r(n))$, then $\boldsymbol{m}_n/\sqrt{\boldsymbol{v}_n} \approx \boldsymbol{m}_n/\sqrt{\boldsymbol{v}} + a(n+1)r(n)\boldsymbol{M}_{n+1}^{(v)}$ and $\mathrm{E}\left[\boldsymbol{M}_{n+1}^{(v)}\right] = \boldsymbol{0}$, since $\mathrm{E}\left[\boldsymbol{g}_l \odot \boldsymbol{g}_l - \boldsymbol{v}\right] = \boldsymbol{0}$. For a stationary second moment $\boldsymbol{v}$, the random variable $\{\boldsymbol{M}_n^{(v)}\}$ is a martingale difference sequence with a bounded second moment. Therefore $\{\boldsymbol{M}_{n+1}^{(v)}\}$ can be subsumed into $\{\boldsymbol{M}_{n+1}\}$ in update rules Eq. (165). The factor $1/\sqrt{\boldsymbol{v}}$ can be componentwise incorporated into the gradient $\boldsymbol{g}$ which corresponds to rescaling the parameters without changing the minimum. □

The differential equation for the Heavy Ball with Friction (HBF) is:

$$\ddot{\boldsymbol{\theta}}_t \;+\; a\, \dot{\boldsymbol{\theta}}_t \;+\; \nabla f(\boldsymbol{\theta}_t) \;=\; \boldsymbol{0} \;, \tag{167}$$

where $a > 0$ is the dampening coefficient. According to Attouch et al. [3] the energy is

$$E(t) \;=\; \frac{1}{2} \left|\dot{\boldsymbol{\theta}}(t)\right|^2 \;+\; f(\boldsymbol{\theta}(t)) \;, \tag{168}$$

which corresponds to the sum of kinetic and potential energy. This energy formulation leads to

$$\dot{E}(t) \;=\; -a \left|\dot{\boldsymbol{\theta}}(t)\right|^2 \;. \tag{169}$$

Thus, the energy is dissipated with increasing t. Therefore the trajectories asymptotically converge to equilibria which are local minima of f.

Since Adam can be expressed as differential equation and has a Lyapunov function, the idea of (T, δ) perturbed ODEs [7, 23, 6] carries over to Adam. Therefore the convergence of Adam with TTUR can be proved via two time-scale stochastic approximation analysis like in Borkar [7] for stationary second moments of the gradient.

A4 Experiments: Additional Details and Results

A4.1 CelebA

We compare TTUR to the original GAN training methods at the Large-scale CelebFaces Attributes (CelebA) dataset [37]. CelebA is a well established benchmark to evaluate GANs [24]. The CelebA images were center cropped to 64×64 pixels.

A4.1.1 BEGAN

We start to test TTUR with the Boundary Equilibrium GAN (BEGAN) [4]. The BEGAN discriminator is a 3×3 convolutional autoencoder for 64×64 input/output images with exponential linear units (ELUs) [12] as activation function. The autoencoder architecture is schematically depicted in Figure A40. The encoder consists of a convolutional layer followed by a variable number of blocks of convolutional layers. Each block consists of three convolutional layers, where the last layer in each block except the last one is a down-sampling layer implemented as sub-sampling with stride 2. The

number of blocks is calculated by $\log_2(image\_height) - 2$. The last layer is a fully-connected layer. The decoder starts with a fully-connected layer followed by the same number of convolutional blocks as the encoder, however the down-sampling layers are replaced by up-sampling layers implemented by nearest neighbor. The last layer is a single convolutional layer. The generator architecture is a clone of the discriminator decoder architecture. Mini-batch size is 16 and $n = 128$. The 128 hidden units, the first layer of the generator, are uniform distributed between -1 und 1. BEGAN implements a learning rate scheduling by halving the learning rates every 100,000 mini-batches. For optimization we use Adam with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ (see Section A6 for the original implementation that we used).

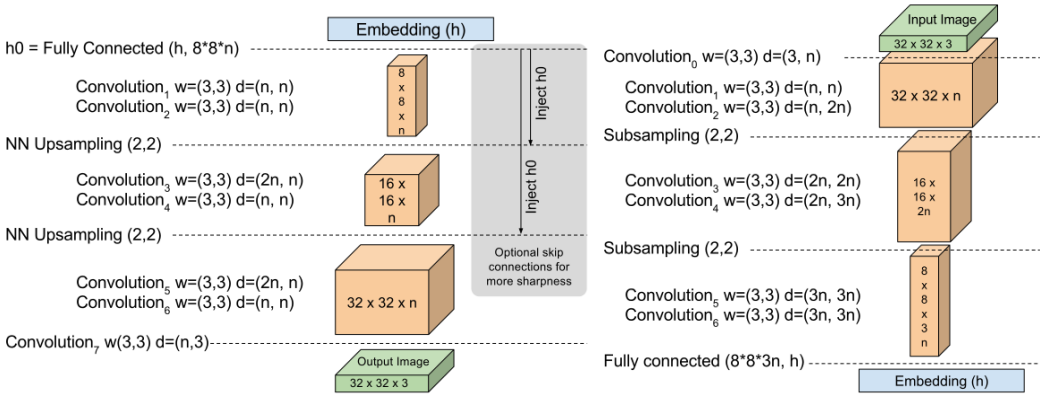

Figure A40: BEGAN network architecture for the generator and discriminator. Left: Generator/Decoder. Right: Encoder. Figure is taken from Berthelot et al. 2017 [4].

Figure A41 shows the FID averaged over 8 runs during learning BEGAN models with the original learning method and with TTUR. TTUR learning rates are given as pairs (b, a) of discriminator learning rate b and generator learning rate a. We report the average FID and standard deviation for 8 runs for TTUR and the training procedure every 5,000 mini-batches. Figure A42 shows the FID at the end of the training. TTUR outperformed the original training starting from mini-batch around 100k. The best FIDs that could be obtained with original BEGANs and TTUR trained BEGANs over all runs are 28.55 and 26.19, respectively (see Table 1).

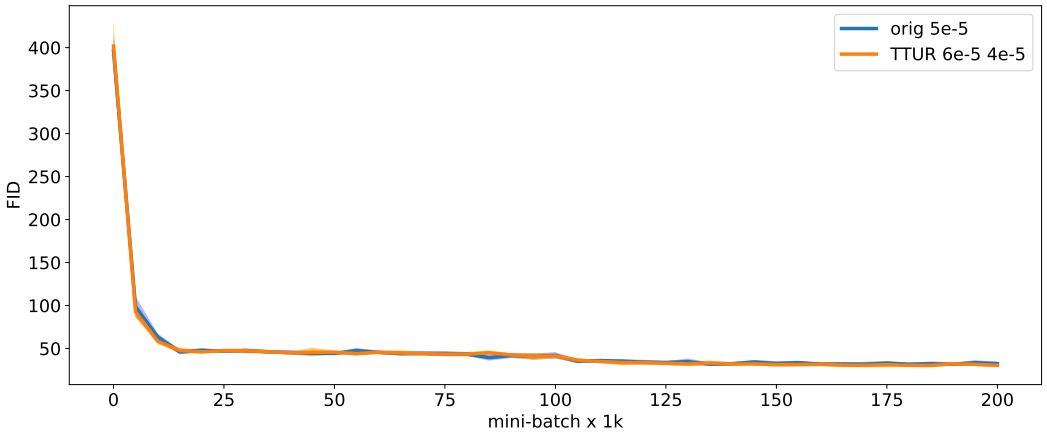

Figure A41: FID for BEGAN trained on CelebA.

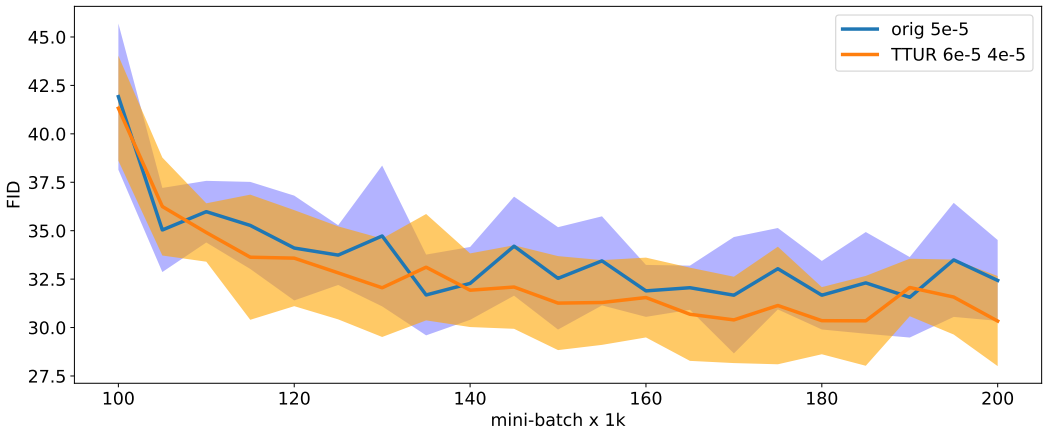

Figure A42: FID for BEGAN trained on CelebA zoomed in starting from 100k mini-batches.

A4.1.2 DCGAN

The next experiment is to test TTUR for the deep convolutional GAN (DCGAN) [46] at CelebA. The DCGAN discriminator consists of 4 convolution layers with batch normalization [26] and leaky ReLUs [38]. The last layer is a fully-connected layer connected to a single sigmoid output unit. The generator starts with a fully-connected layer followed by four transposed convolution layers with ReLU activations [42], except for the last convolutional layer, which has $\tanh$ activations. The 100 hidden variables, which drive the generator, are uniformly distributed. DCGAN is trained with mini-batches of size 64 and Adam. The Adam optimizer is used with its default parameters, except $\beta_1 = 0.5$ (for the implementation see Section A6).

Figure A43 shows the FID during learning DCGAN with the original learning method and with TTUR. The original training method is faster at the beginning but TTUR achieves better performance. Figure A44 zooms in at the region of 10k to 50k mini-batch updates of Figure A44 to show the difference between TTUR and original learning of DCGANs. DCGAN achieves a lower FID than BEGAN, therefore gives better results which we attribute to higher variety of the generated images. TTUR reaches a lower FID than the original method.

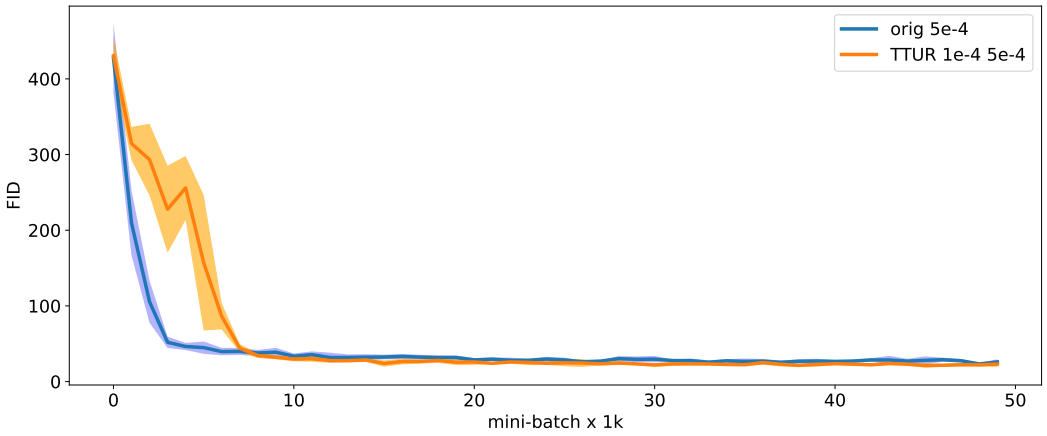

Figure A43: FID for DCGAN trained on CelebA. The original training is faster at the beginning, however TTUR reaches a lower FID.

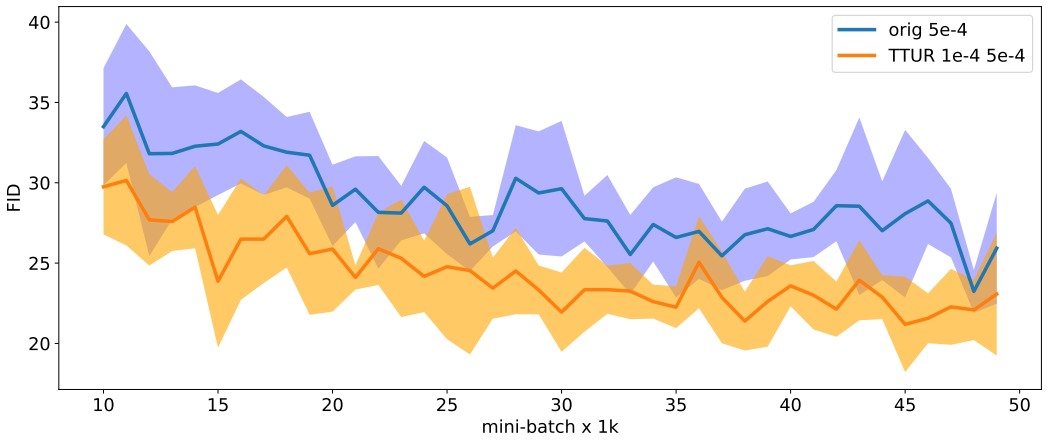

Figure A44: FID for DCGAN trained on CelebA zoomed in at the region of 10k to 50k mini-batch updates of Figure A43 to show the difference between TTUR and original learning of DCGANs. That TTUR reaches a lower FID than the original method becomes more visible compared to Figure A43.

A4.2 One Billion Word

The One Billion Word Benchmark [11] serves to compare TTUR to the original training for the Improved Wasserstein GAN (WGAN) [22]. The character-level generative language modelbatch is a 1D convolutional neural network (CNN) which maps a latent vector into a sequence of one-hot character vectors of dimension 32 given by the maximum of a softmax output. The discriminator is also a 1D CNN applied to sequences of one-hot vectors of 32 characters. Both the discriminator and the generator consist of 5 ResNet blocks with two 1d-convolutional layers each. The discriminator has as last layer a fully-connected layer and one output unit. The generator has as first layer a fully-connected layer and as last layer a softmax layer. The 128 hidden units in the first layer of the generator are normally distributed. Mini-batch size is 64. The Improved WGAN for the language model is trained by Adam with default parameters, except $\beta_1 = 0.5$ and $\beta_2 = 0.9$. For the used implementation see Section A6. In contrast to the original code, where the critic is updated 10 times for each generator update, TTUR updates the discriminator only once, therefore we align the training progress with wall-clock time. TTUR can use a higher learning rate for the discriminator since TTUR stabilizes learning. As the FID criterium only works for images, we measured the performance by the Jensen-Shannon-divergence (JSD) between the model and the real world distribution of 4-gram and 6-gram statistics as has been done previously [22]. For calculating the JSD, we precomputed before learning the 4-gram and 6-gram statistics on the whole dataset. During learning we computed every 100 iterations the same statistics for samples from the model, where the number of samples is 10 times the batch size. The JSD is calculated from the statistics of the whole dataset and the statistics of model samples.

For original and TTUR training, we report in Figure A45 the normalized mean JSD averaged over 10 runs for 4-gram and 6-gram word evaluations. TTUR outperforms the standard training for both evaluation measures. The 6-gram statistics shows TTUR enables to learn to generate more subtle pseudo-words which better resembles real words than the original training. In Table A8 we show randomly chosen samples from models trained with original method and TTUR.

Table A8: Samples of One Billion Word benchmark generated by Improved WGAN trained with TTUR (left) the original method (right).

Dry Hall Sitning tven the concer	No say that tent Franstal at Bra
There are court phinchs hasffort	Caulh Paphionars tven got corfle
He scores a supponied foutver il	Resumaly , braaky facting he at
Bartfol reportings ane the depor	On toipe also houd , aid of sole
Seu hid , it 's watter 's remold	When Barrysels commono toprel to
Later fasted the store the inste	The Moster suprr tent Elay diccu
Indiwezal deducated belenseous K	The new vebators are demases to
Starfers on Rbama 's all is lead	Many 's lore wockerssaow 2 2) A
Inverdick oper , caldawho 's non	Andly , has le wordd Uold steali
She said , five by theically rec	But be the firmoters is no 200 s
RichI , Learly said remain .''''	Jermueciored a noval wan 't mar
Reforded live for they were like	Onles that his boud-park , the g
The plane was git finally fuels	ISLUN , The crather wilh a them
The skip lifely will neek by the	Fow 22o2 surgeedeto , theirestra
SEW McHardy Berfect was luadingu	Make Sebages of intarmamates , a
But I pol rated Franclezt is the	Gullla " has cautaria Thoug ly t

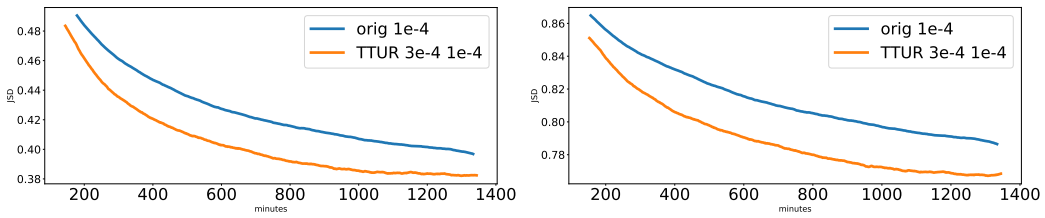

Figure A45: Performance of Improved WGAN models trained with the original (orig) and our TTUR method on the One Billion Word benchmark. The performance is measured by the normalized Jensen-Shannon-divergence based on 4-grams (left) and 6-grams (right) and averaged over 10 runs. TTUR learning clearly improved the original learning which is more prominent at 6-gram than at 4-gram.

A4.3 LSUN Bedrooms

A4.3.1 BEGAN

TTUR is compared to original GAN training for BEGANs at the bedrooms category of the large scale image database LSUN [53]. At the start of learning, the original method is faster than TTUR as depicted in Figure A46. However, after 40k mini-batches, TTUR reaches a lower FID, which is more obvious at the zoom-in given by Figure A47. TTUR leads to lower FID and improves the original training procedure. Figure 12 shows examples of samples of BEGAN trained with the original method and with TTUR.

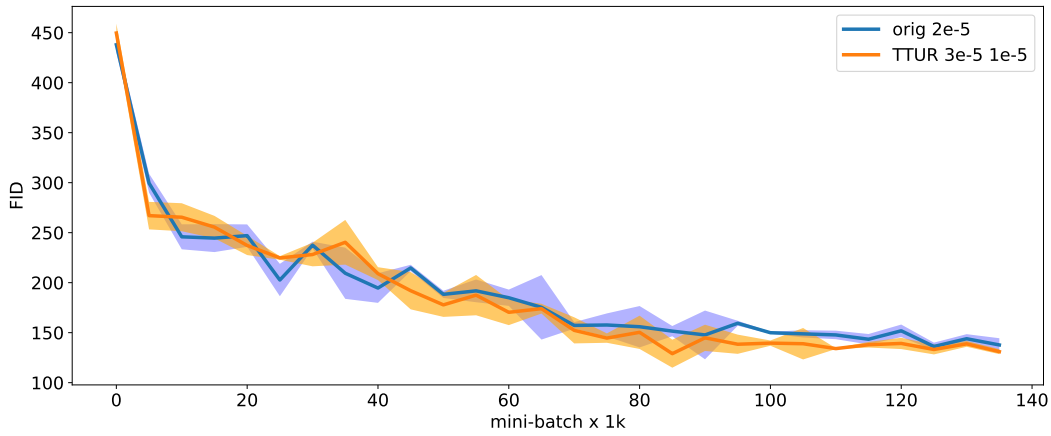

Figure A46: FID for BEGAN trained on LSUN bedroom. TTUR leads to lower FID and improves the original training procedure.

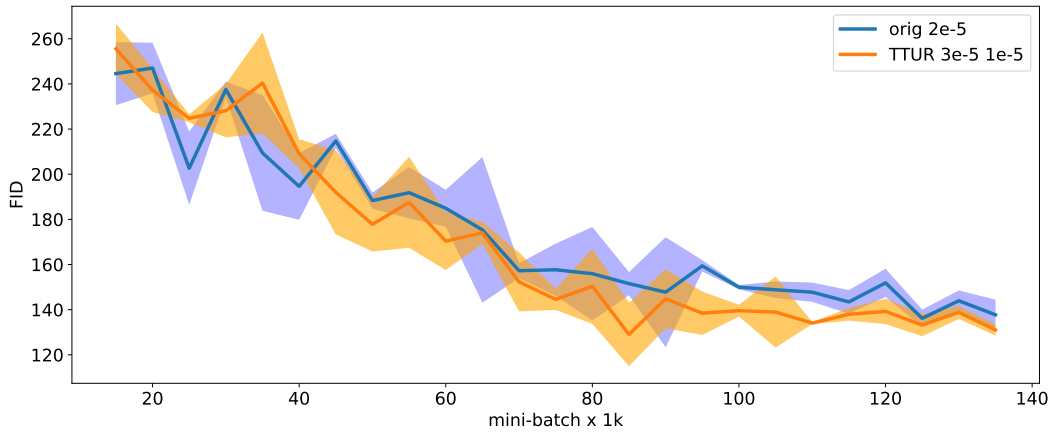

Figure A47: FID for BEGAN trained on LSUN bedroom zoomed in at the end of training of Figure A46 to show the difference between TTUR and original learning of BEGANs. After 40k mini-batches TTUR reaches a lower FID.

A4.3.2 DCGAN

TTUR is compared to original GAN training for DCGANs at the bedrooms category of the large scale image database LSUN [53]. At the start of learning, the original method is faster than TTUR as depicted in Figure A48. However, after 50k mini-batches, TTUR reaches a lower FID, which is more obvious at the zoom-in given by Figure A49. TTUR leads to lower FID and improves the original training procedure. Figure 14 shows examples of samples of DCGAN trained with the original method and with TTUR.

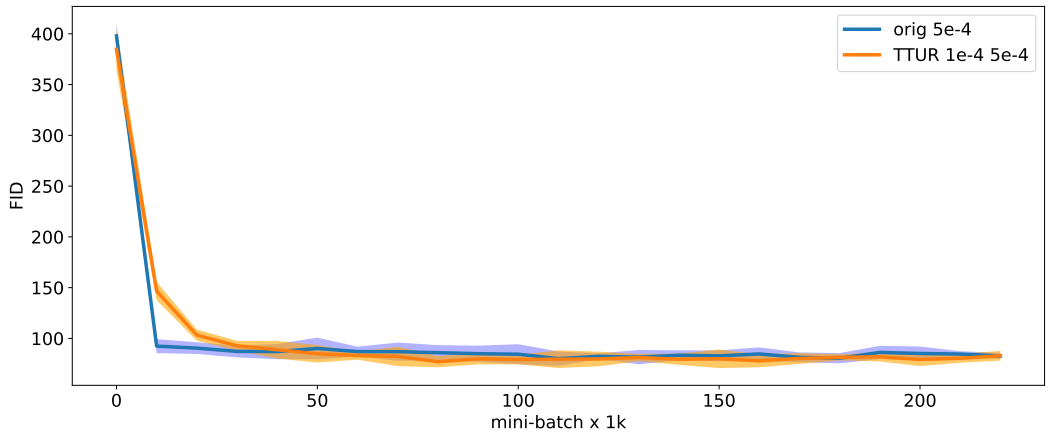

Figure A48: FID for DCGAN trained on LSUN bedroom. TTUR clearly leads to lower FID and improves the original training procedure.

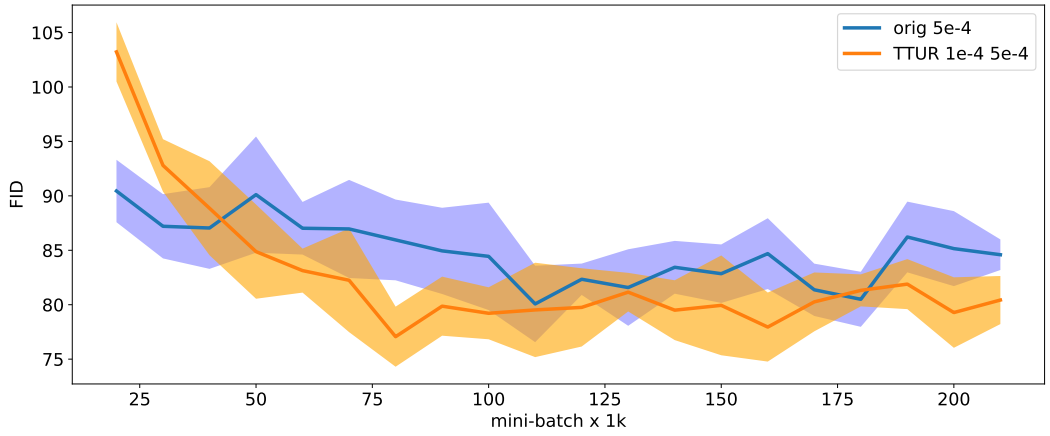

Figure A49: FID for DCGAN trained on LSUN bedroom zoomed in at the end of training of Figure A48 to show the difference between TTUR and original learning of DCGANs. After 50k mini-batches TTUR reaches a lower FID.

A4.4 Fixed k BEGAN at CelebA

Here we introduce a BEGAN variant with fixed k. k trades-off real-world examples and generated examples when training the discriminator. BEGAN maintains the equilibrium $\mathrm{E}[\mathcal{L}(G(\boldsymbol{z}))] = \gamma \mathrm{E}[\mathcal{L}(\boldsymbol{x})]$ with a variable k_t by adjusting it with $k_{t+1} = k_t + \lambda_k(\gamma \mathcal{L}(\boldsymbol{x}) - \mathcal{L}(G(\boldsymbol{z}_G)))$. Figure A50 shows that fixing k to 0.08 and using a small learning rate leads to a smaller FID than in the original BEGAN. However, TTUR does not show an improvement over the standard training procedure.

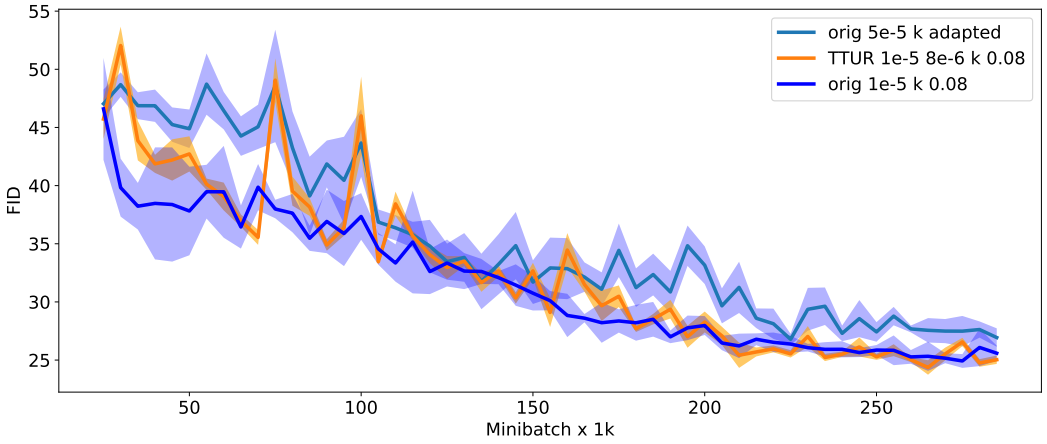

Figure A50: FID for BEGAN with fixed k trained on CelebA. The shaded area is the standard deviation for 10 runs for each experiment.

A5 Discriminator vs. Generator Learning Rate

The convergence proof for learning GANs with TTUR assumes that the generator learning rate will eventually become small enough to ensure convergence of the discriminator learning. At some time point, the perturbations of the discriminator updates by updates of the generator parameters are sufficient small to assure that the discriminator converges. Crucial for discriminator convergence is the magnitude of the perturbations which the generator induces into the discriminator updates. These perturbations are not only determined by the generator learning rate but also by its loss function, current value of the loss function, optimization method, size of the error signals that reach the generator (vanishing or exploding gradient), complexity of generator's learning task, architecture of the generator, regularization, and others. Consequently, the size of generator learning rate does not solely determine how large the perturbations of the discriminator updates are but serve to modulate them. Thus, the generator learning rate may be much larger than the discriminator learning rate without inducing large perturbation into the discriminator learning.

Even the learning dynamics of the generator is different from the learning dynamics of the discriminator, though they both have the same learning rate. Figure A51 shows the loss of the generator and the discriminator for an experiment with DCGAN on CelebA, where the learning rate was 0.0005 for both the discriminator and the generator. However, the discriminator loss is decreasing while the generator loss is increasing. This example shows that the learning rate neither determines the perturbations nor the progress in learning for two coupled update rules. The choice of the learning rate for the generator should be independent from choice for the discriminator. Also the search ranges of discriminator and generator learning rates should be independent from each other, but adjusted to the corresponding architecture, task, etc.

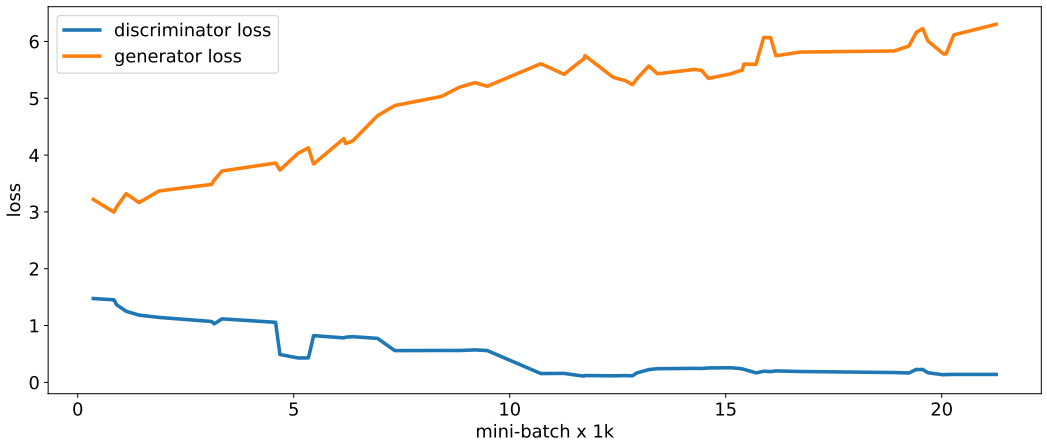

Figure A51: The respective losses of the discriminator and the generator show the different learning dynamics of the two networks.

A6 Used Software, Datasets, and Pretrained Models

All experiments rely on the respective reference implementations for corresponding GAN model. The software framework for our experiments was Tensorflow 1.1 [1] and Python 3.6. We used following software, datasets and pretrained models:

- BEGAN in Tensorflow, `https://github.com/carpedm20/BEGAN-tensorflow`, Fixed random seeds removed. Accessed: 2017-05-30
- DCGAN in Tensorflow, `https://github.com/carpedm20/DCGAN-tensorflow`, Fixed random seeds removed. Accessed: 2017-04-03
- Improved Training of Wasserstein GANs, language model, `https://github.com/igul222/improved_wgan_training/blob/master/gan_language.py`, Accessed: 2017-06-12
- Inception-v3 pretrained, `http://download.tensorflow.org/models/image/imagenet/inception-2015-12-05.tgz`, Accessed: 2017-05-02

A7 References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv e-prints*, abs/1701.07875, 2017.

[3] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method, I. the continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Communications in Contemporary Mathematics*, 2(1):1–34, 2000.

[4] D. Berthelot, T. Schumm, and L. Metz. BEGAN: boundary equilibrium generative adversarial networks. *ArXiv e-prints*, abs/1703.10717, 2017.

[5] D. P. Bertsekas and J. N. Tsitsiklis. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

[6] S. Bhatnagar, H. L. Prasad, and L. A. Prashanth. *Stochastic Recursive Algorithms for Optimization*. Lecture Notes in Control and Information Sciences. Springer-Verlag London, 2013.

[7] V. S. Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[8] V. S. Borkar and S. P. Meyn. The O.D.E. method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

[9] J. H. Lim J. C. and Ye. Geometric GAN. *ArXiv e-prints*, abs/1705.02894, 2017.

[10] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *ArXiv e-prints*, abs/1612.02136, 2016. The International Conference on Learning Representations (ICLR), 2017.

[11] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. *ArXiv e-prints*, abs/1312.3005, 2013.

[12] D.-A., T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *ArXiv e-prints*, abs/1511.07289, 2015. The International Conference on Learning Representations (ICLR), 2016.

[13] D. DiCastro and R. Meir. A convergent online single time scale actor critic algorithm. *J. Mach. Learn. Res.*, 11:367–410, 2010.

[14] D. C. Dowson and B. V. Landau. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12:450–455, 1982.

[15] M. Fréchet. Sur la distance de deux lois de probabilité. *C. R. Acad. Sci. Paris*, 244:689–692, 1957.

[16] S. Gadat, F. Panloup, and S. Saadane. Stochastic heavy ball. *ArXiv e-prints*, abs/1609.04228, 2016.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.

[18] I. J. Goodfellow. On distinguishability criteria for estimating generative models. *ArXiv e-prints*, abs/1412.6515, 2014.

[19] I. J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *ArXiv e-prints*, abs/1701.00160, 2017.

[20] X. Goudou and J. Munier. The gradient and heavy ball with friction dynamical systems: the quasiconvex case. *Mathematical Programming*, 116(1):173–191, 2009.

[21] P. Grnarova, K. Y. Levy, A. Lucchi, T. Hofmann, and A. Krause. An online learning approach to generative adversarial networks. *ArXiv e-prints*, abs/1706.03269, 2017.

[22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of Wasserstein GANs. *ArXiv e-prints*, abs/1704.00028, 2017.

[23] M. W. Hirsch. Convergent activation dynamics in continuous time networks. *Neural Networks*, 2(5):331–349, 1989.

[24] R. D. Hjelm, A. P. Jacob, T. Che, K. Cho, and Y. Bengio. Boundary-seeking generative adversarial networks. *ArXiv e-prints*, abs/1702.08431, 2017.

[25] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[26] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Journal of Machine Learning Research*, 37:448–456, 2015. Proceedings of the 32nd International Conference on Machine Learning (ICML15).

[27] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *ArXiv e-prints*, abs/1611.07004, 2016.

[28] P. Karmakar and S. Bhatnagar. Two timescale stochastic approximation with controlled Markov noise and off-policy temporal difference learning. *ArXiv e-prints*, abs/1503.09105, 2017.

[29] P. Karmakar, S. Bhatnagar, and A. Ramaswamy. Dynamics of stochastic approximation with Markov iterate-dependent noise with the stability of the iterates not ensured. *ArXiv e-prints*, abs/1601.02217, 2016.

[30] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *ArXiv e-prints*, abs/1412.6980, 2014. The International Conference on Learning Representations (ICLR), San Diego, 2015.

[31] V. R. Konda. *Actor-Critic Algorithms*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2002.

[32] V. R. Konda and V. S. Borkar. Actor-critic-type learning algorithms for Markov decision processes. *SIAM J. Control Optim.*, 38(1):94–123, 1999.

[33] V. R. Konda and J. N. Tsitsiklis. Linear stochastic approximation driven by slowly varying Markov chains. *Systems & Control Letters*, 50(2):95–102, 2003.

[34] V. R. Konda and J. N. Tsitsiklis. Convergence rate of linear two-time-scale stochastic approximation. *The Annals of Applied Probability*, 14(2):796–819, 2004.

[35] H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Applied Mathematical Sciences. Springer-Verlag New York, 1978.

[36] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *ArXiv e-prints*, abs/1609.04802, 2016.

[37] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[38] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning (ICML13)*, 2013.

[39] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *ArXiv e-prints*, abs/1611.02163, 2016.

[40] A. Mokkadem and M. Pelletier. Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms. *The Annals of Applied Probability*, 16(3):1671–1702, 2006.

[41] V. Nagarajan and J. Z. Kolter. Gradient descent GAN optimization is locally stable. *ArXiv e-prints*, abs/1706.04156, 2017.

[42] V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML10)*, pages 807–814, 2010.

[43] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[44] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[45] H. L. Prasad, L. A. Prashanth, and S. Bhatnagar. Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*, pages 1371–1379, 2015.

[46] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[47] A. Ramaswamy and S. Bhatnagar. Stochastic recursive inclusion in two timescales with an application to the lagrangian dual problem. *Stochastics*, 88(8):1173–1187, 2016.

[48] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. *ArXiv e-prints*, abs/1606.03498, 2016.

[49] V. B. Tadić. Almost sure convergence of two time-scale stochastic approximation algorithms. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3802–3807, 2004.

[50] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. *ArXiv e-prints*, abs/1511.01844, 2015. The International Conference on Learning Representations (ICLR), 2016.

[51] L. N. Wasserstein. Markov processes over denumerable products of spaces describing large systems of automata. *Probl. Inform. Transmission*, 5:47–52, 1969.

[52] Y. Wu, Y. Burda, R. Salakhutdinov, and R. B. Grosse. On the quantitative analysis of decoder-based generative models. *ArXiv e-prints*, abs/1611.04273, 2016. The International Conference on Learning Representations (ICLR), 2017.

[53] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv e-prints*, abs/1506.03365, 2015.

[54] J. Zhang, D. Zheng, and M. Chiang. The impact of stochastic noisy feedback on distributed network utility maximization. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 222–230, 2007.

List of Figures

List of Tables