On the Anatomy of MCMC-based Maximum Likelihood Learning of Energy-Based Models

Erik Nijkamp[*], Mitch Hill[*], Tian Han, Song-Chun Zhu, Ying Nian Wu
Department of Statistics
University of California, Los Angeles
{enijkamp,mkhill,hantian}@ucla.edu {sczhu,ywu}@stat.ucla.edu

Abstract

This study investigates the effects of Markov chain Monte Carlo (MCMC) sampling in unsupervised Maximum Likelihood (ML) learning. Our attention is restricted to the family of unnormalized probability densities for which the negative log density (or energy function) is a ConvNet. We find that many of the techniques used to stabilize training in previous studies are not necessary. ML learning with a ConvNet potential requires only a few hyper-parameters and no regularization. Using this minimal framework, we identify a variety of ML learning outcomes that depend solely on the implementation of MCMC sampling.

On one hand, we show that it is easy to train an energy-based model which can sample realistic images with short-run Langevin. ML can be effective and stable even when MCMC samples have much higher energy than true steady-state samples throughout training. Based on this insight, we introduce an ML method with purely noise-initialized MCMC, high-quality short-run synthesis, and the same budget as ML with informative MCMC initialization such as CD or PCD. Unlike previous models, our energy model can obtain realistic high-diversity samples from a noise signal after training.

On the other hand, ConvNet potentials learned with non-convergent MCMC do not have a valid steady-state and cannot be considered approximate unnormalized densities of the training data because long-run MCMC samples differ greatly from observed images. We show that it is much harder to train a ConvNet potential to learn a steady-state over realistic images. To our knowledge, long-run MCMC samples of all previous models lose the realism of short-run samples. With correct tuning of Langevin noise, we train the first ConvNet potentials for which long-run and steady-state MCMC samples are realistic images.

1. Introduction

1.1. Diagnosing Energy-Based Models

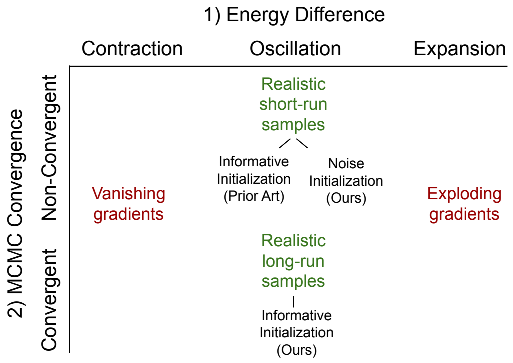
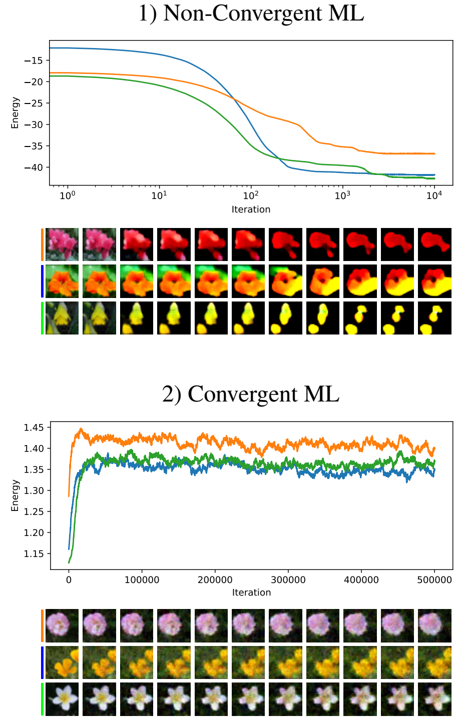

1) Energy Difference
Contraction
Oscillation
Expansion
2) MCMC Convergence
Non-Convergent
Convergent
Vanishing gradients
Realistic short-run samples
Informative Initialization (Prior Art)
Noise Initialization (Ours)
Exploding gradients
Realistic long-run samples
Informative Initialization (Ours)

Figure 1: Two axes characterize ML learning of ConvNet potential energy functions: 1) energy difference between data samples and synthesized samples, and 2) MCMC convergence towards steady-state. Learning a sampler with realistic short-run MCMC synthesis is surprisingly simple whereas learning an energy with realistic long-run samples requires proper MCMC implementation. We propose: a) ML with short-run MCMC and noise initialization of the chains, and b) an explanation and implementation of correct tuning for training models with realistic long-run samples.

Statistical modeling of high-dimensional signals is a challenging task encountered in many academic disciplines and practical applications. We study image signals in this work. When images come without annotations or labels, the effective tools of deep supervised learning cannot be applied and unsupervised techniques must be used. This work focuses on the unsupervised paradigm of the energy-based model (1) with a ConvNet potential function (2).

Previous works studying Maximum Likelihood (ML) training of ConvNet potentials, such as [33, 32, 7], use Langevin MCMC samples to approximate the gradient of

[*] Equal contributions.


1) Non-Convergent ML

2) Convergent ML

Figure 2: Long-run MH-adjusted Langevin paths from data samples to metastable samples for the Oxford Flowers 102 dataset. Models were trained with two variations of Algorithm 1: non-convergent ML trained with $L = 100$ MCMC steps from noise initialization (*top*), and convergent ML trained with $L = 500$ MCMC steps from persistent initialization (*bottom*).

the unknown and intractable log partition function during learning. The authors universally find that after enough model updates, MCMC samples generated by short-run Langevin from *informative initialization* (see Section 2.3) are realistic images that resemble the data.

However, we find that energy functions learned by prior works have a major defect regardless of MCMC initialization, network structure, and auxiliary training parameters. The long-run and steady-state MCMC samples of energy functions from all previous implementations are oversaturated images with significantly lower energy than the observed data (see Figure 2 top, and Figure 3). In this case it is not appropriate to describe the learned model as an approximate density for the training set because the model assigns disproportionately high probability mass to images which differ dramatically from observed data. The systematic difference between high-quality short-run samples and low-quality long-run samples is a crucial phenomenon that appears to have gone unnoticed in previous studies.

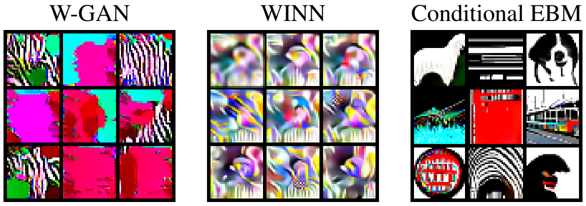


Figure 3: Long-run Langevin samples of recent energy-based models. Probability mass is concentrated on images that have unrealistic appearance. From left to right: Wasserstein-GAN critic on Oxford flowers [1], WINN on Oxford flowers [20], conditional EBM on ImageNet [6]. The W-GAN critic is not trained to be an unnormalized density but we include samples for reference.

1.2. Our Contributions

In this work, we present a fundamental understanding of learning ConvNet potentials by MCMC-based ML. We diagnose previously unrecognized complications that arise during learning and distill our insights to train models with new capabilities. Our main contributions are:

- Identification of two distinct axes which characterize each parameter update in MCMC-based ML learning: 1) energy difference of positive and negative samples, and 2) MCMC convergence or non-convergence. Contrary to common expectations, convergence is *not* needed for high-quality synthesis. See Figure 1 and Section 3.
- The first ConvNet potentials trained using ML with purely noise-initialized MCMC. Unlike prior models, our model can efficiently generate realistic and diverse samples after training from noise alone. See Figure 7. This method is further explored in our companion work [24].
- The first ConvNet potentials with realistic steady-state samples. To our knowledge, ConvNet potentials with realistic MCMC sampling in the image space are unobtainable by all previous training implementations. We refer to [18] for a discussion. See Figure 2 (bottom) and Figure 8 (middle and right column).
- Mapping the macroscopic structure of image space energy functions using diffusion in a magnetized energy landscape for unsupervised cluster discovery. See Figure 9.

1.3. Related Work

1.3.1 Energy-Based Image Models

Energy-based models define an unnormalized probability density over a state space to represent the distribution of states in a given system. The Hopfield network [15] adapted the Ising energy model into a model capable of representing arbitrary observed data. The RBM [14] and FRAME (Filters, Random field, And Maximum Entropy) [36, 30] models introduce energy functions with greater representational capacity. The RBM uses hidden units which have a joint density with the observable image pixels. The FRAME model uses convolutional filters and histogram matching to learn data features.

The pioneering work [13] studies the hierarchical energy-based model. [23] is an important early work proposing feedforward neural networks to model energy functions. The energy-based model in the form of (2) is introduced in [4]. Deep variants of the FRAME model [33, 21] are the first to achieve realistic synthesis with a ConvNet potential and Langevin sampling. [6] applies similar methods.

The Multi-grid model [7] learns an ensemble of ConvNet potentials for images of different scales with finite-budget Langevin sampling. Synthesized images from smaller scales are used as the informative initialization for MCMC sampling at larger scales.

Learning a ConvNet potential with the help of a generator network as approximative direct sampler is explored in [17, 5, 31, 32, 10, 18]. [35, 34] explore an adversarial interpretation of ML learning. These works show connections to W-GAN and herding [29].

The INN model [26] learns unnormalized densities in a discriminative framework. [16, 19] investigate a ConvNet parameterization of this model from the perspective of image classification and synthesis respectively. The W-GAN [1] framework is adapted to the INN method in the WINN model [20].

Two common threads between these learning algorithms are the ML parameter update (8) and the Langevin image update (9). We emphasize that some of the above works do not use both.

Although many of these works claim to train the energy (2) to be an approximate unnormalized density for the observed images, the resulting energy functions do not have a steady-state that reflects the data (see Figure 3). Short-run Langevin samples from informative initialization are presented as approximate steady-state samples, but further investigation shows long-run Langevin consistently disrupts the realism of short-run images. Our work is the first to address and remedy the systematic non-convergence of all prior implementations.

We emphasize that unrealistic image space steady-states are a central concern specifically when training ConvNet potentials (2). Earlier energy-based models such as RBM do not exhibit a dramatic difference in realism between short-run samples from informative initialization and steady-state images. Variational Walkback [9] can learn an energy-free MCMC transition with a realistic steady-state in the image space.

1.3.2 Energy Landscape Mapping

The full potential of the energy-based model lies in the structure of the energy landscape. Hopfield observed that the energy landscape is a model of associative memory [15]. Diffusion along the potential energy manifold is analogous to memory recall because the diffusion process will gradually refine a high-energy image (an incomplete or corrupted memory) until it reaches a low-energy metastable state, which corresponds to the revised memory. Techniques for mapping and visualizing the energy landscape of non-convex functions in the physical chemistry literature [2, 27] have been applied to map the latent space of Cooperative Networks [11]. Defects in the energy function (2) from previous ML implementations prevent these techniques from being applied in the image space. Our convergent ML models enable image space mapping.

2. Learning Energy-Based Models

In this section, we review the established principles of the MCMC-based ML learning from prior works such as [12, 36, 33].

2.1. Maximum Likelihood Estimation

An energy-based model is a Gibbs-Boltzmann density

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp\{-U(x;\theta)\} \tag{1}$$

over signals $x \in \mathcal{X} \subset \mathbb{R}^N$. The energy potential $U(x;\theta)$ belongs to a parametric family $\mathcal{U} = \{U(\cdot\ ;\theta) : \theta \in \Theta\}$. The intractable constant $Z(\theta) = \int_\mathcal{X} \exp\{-U(x;\theta)\}dx$ is never used explicitly because the potential $U(x;\theta)$ provides sufficient information for MCMC sampling. In this paper we focus our attention on energy potentials with the form

$$U(x;\theta) = F(x;\theta) \tag{2}$$

where $F(x;\theta)$ is a convolutional neural network with scalar output and weights $\theta \in \mathbb{R}^D$.

In ML learning, we seek to find $\theta \in \Theta$ such that the parametric model $p_\theta(x)$ is a close approximation of the data distribution $q(x)$. One measure of closeness is the Kullback-Leibler (KL) divergence. Learning proceeds by solving

$$\begin{aligned} \arg\min_\theta \mathcal{L}(\theta) &= \arg\min_\theta D_{KL}(q\|p_\theta) && (3) \\ &= \arg\min_\theta \{\log Z(\theta) + E_q[U(X;\theta)]\}\,. && (4) \end{aligned}$$

We can minimize $\mathcal{L}(\theta)$ by finding the roots of the derivative

$$\frac{d}{d\theta}\mathcal{L}(\theta) = \frac{d}{d\theta}\log Z(\theta) + \frac{d}{d\theta}E_q[U(X;\theta)]. \tag{5}$$

The term $\frac{d}{d\theta} \log Z(\theta)$ is intractable, but it can be expressed

$$\frac{d}{d\theta} \log Z(\theta) = -E_{p_\theta}\left[\frac{\partial}{\partial\theta} U(X;\theta)\right]. \tag{6}$$

The gradient used to learn θ then becomes

$$\frac{d}{d\theta}\mathcal{L}(\theta) = \frac{d}{d\theta} E_q[U(X;\theta)] - E_{p_\theta}\left[\frac{\partial}{\partial\theta} U(X;\theta)\right] \tag{7}$$

$$\approx \frac{\partial}{\partial\theta}\left(\frac{1}{n}\sum_{i=1}^{n} U(X_i^+;\theta) - \frac{1}{m}\sum_{i=1}^{m} U(X_i^-;\theta)\right) \tag{8}$$

where $\{X_i^+\}_{i=1}^n$ are i.i.d. samples from the data distribution q (called *positive* samples since probability is increased), and $\{X_i^-\}_{i=1}^m$ are i.i.d. samples from current learned distribution p_θ (called *negative* samples since probability is decreased). In practice, the positive samples $\{X_i^+\}_{i=1}^n$ are a batch of training images and the negative samples $\{X_i^-\}_{i=1}^m$ are obtained after L iterations of MCMC sampling.

2.2. MCMC Sampling with Langevin Dynamics

Obtaining the negative samples $\{X_i^-\}_{i=1}^m$ from the current distribution p_θ is a computationally intensive task which must be performed for each update of θ. ML learning does not impose a specific MCMC algorithm. Early energy-based models such as the RBM and FRAME model use Gibbs sampling as the MCMC method. Gibbs sampling updates each dimension (one pixel of the image) sequentially. This is computationally infeasible when training an energy with the form (2) for standard image sizes.

Several works studying the energy (2) recruit Langevin Dynamics to obtain sample from p_θ [33, 21, 32, 7, 20]. The Langevin Equation

$$X_{\ell+1} = X_\ell - \frac{\varepsilon^2}{2}\frac{\partial}{\partial x} U(X_\ell;\theta) + \varepsilon Z_\ell, \tag{9}$$

where $Z_\ell \sim \mathrm{N}(0, I_N)$ and $\varepsilon > 0$, has stationary distribution p_θ [8, 22]. A complete implementation of Langevin Dynamics requires a momentum update and Metropolis-Hastings update in addition to (9), but most authors find that these can be ignored in practice for small enough ε [3].

Like most MCMC methods, Langevin dynamics exhibits high auto-correlation and has difficulty mixing between separate modes. The consistent appearance of long-run MCMC samples can actually be a useful feature of a learned potential because a metastable representation is needed for mapping applications [11]. In general it is not appropriate to describe long-run Langevin samples from a fixed low-energy starting image as steady-state samples because the chains cannot mix between modes in computationally feasible time scales. Even so, long-run Langevin samples with a suitable initialization can still be considered approximate steady-state samples, as discussed in the next section.

2.3. MCMC Initialization

We distinguish two main branches of MCMC initialization: *informative initialization*, where the density of initial states is meant to approximate the model density, and *non-informative initialization*, where initial states are obtained from a distribution that is unrelated to the model density. *Noise initialization* is a specific type of non-informative initialization where initial states come from a noise distribution such as uniform or Gaussian.

In the most extreme case, a Markov chain initialized from its steady-state will follow the steady-state distribution after a single MCMC update. In more general cases, a Markov chain initialized from an image that is likely under the steady-state can converge much more quickly than a Markov chain initialized from noise. For this reason, all prior works studying ConvNet potentials use informative initialization during training and for generation of images after training has concluded.

Data-based initialization uses samples from the training data as the initial MCMC states. Contrastive Divergence (CD) [12] introduces this practice. To our knowledge CD has *not* been used to trained the energy (2). In our diagnosis it appears that CD can be problematic when training ConvNet potentials for reasons discussed in Section 3.2. The Multigrid Model [7] generalizes CD by using multi-scale energy functions to sequentially refine downsampled data.

Persistent initialization uses negative samples from a previous learning iteration as initial MCMC states in the current iteration. The persistent chains can be initialized from noise as in [36, 33] or from data samples as in Persistent Contrastive Divergence (PCD) [25]. The authors of [20, 6] store a large set of persistent images. The Cooperative Learning model [32] generalizes persistent chains by learning a generator network for MCMC initialization in tandem with the energy.

In this paper we consider long-run Langevin chains from both data-based initialization such as CD and persistent initialization such as PCD to be approximate steady-state samples, even when Langevin chains cannot mix between modes. Prior art indicates that both initialization types span the modes of the learned density, and long-run Langevin will obtain fair MCMC samples within each mode.

Informative MCMC initialization during ML training can limit the ability of the final model p_θ to generate new and diverse synthesized images after training. MCMC samples initialized from noise distributions after training tend to result in images with a similar appearance when informative initialization is used in training.

In contrast to common wisdom, we find that informative initialization is not necessary for efficient and realistic synthesis when training ConvNet potentials with ML. In accordance with common wisdom, we find that informative initialization is essential for learning a realistic steady-state.

3. Two Axes of ML Learning

Inspection of the gradient (8) reveals the central role of the difference of the average energy of negative and positive samples. Let

$$d_{s_t}(\theta) = E_q[U(X;\theta)] - E_{s_t}[U(X;\theta)] \qquad (10)$$

where $s_t(x)$ is the distribution of negative samples given the finite-step MCMC sampler and initialization used at training step t. The difference $d_{s_t}(\theta)$ measures whether the positive samples from the data distribution q or the negative samples from s_t are more likely under the model p_θ. The ideal case $p_\theta = q$ (perfect learning) and $s_t = p_\theta$ (exact MCMC convergence) satisfies $d_{s_t}(\theta) = 0$. A large value of $|d_{s_t}|$ indicates that either learning or sampling (or both) have not converged.

Although $d_{s_t}(\theta)$ is not equivalent to the ML objective (4), it bridges the gap between theoretical ML and the behavior encountered when MCMC approximation is used. Two outcomes occur for each update on the parameter path $\{\theta_t\}_{t=1}^{T+1}$:

1. $d_{s_t}(\theta_t) < 0$ (expansion) or $d_{s_t}(\theta_t) > 0$ (contraction)
2. $s_t \approx p_{\theta_t}$ (MCMC convergence) or $s_t \not\approx p_{\theta_t}$ (MCMC non-convergence) .

We find that only the first axis governs the stability and short-run synthesis results of the learning process. Oscillation of expansion and contraction updates is an indicator of stable ML learning, but this can occur in cases where either s_t is always approximately convergent or where s_t never converges.

Behavior along the second axis determines the realism of steady-state samples from the learned energy. Samples from p_{θ_t} will be realistic if and only if s_t has realistic samples and $s_t \approx p_{\theta_t}$. We use *convergent ML* to refer to implementations where $s_t \approx p_{\theta_t}$ for all $t > t_0$, where t_0 represents burn-in learning steps (e.g. early stages of persistent learning). We use *non-convergent ML* to refer to all other implementations. All prior ConvNet potentials are learned with non-convergent ML, although this is not recognized by previous authors.

Without proper tuning of the sampling phase, the learning heavily gravitates towards non-convergent ML. In this section we outline principles to explain this behavior and provide a remedy for the tendency of model non-convergence.

3.1. First Axis: Expansion or Contraction

Following prior art for high-dimensional image models, we use the Langevin Equation (9) to obtain MCMC samples. Let w_t give the joint distribution of a Langevin chain $(Y_t^{(0)}, \ldots, Y_t^{(L)})$ at training step t, where $Y_t^{(0)}$ is obtained from MCMC initialization, $Y_t^{(\ell+1)}$ is obtained by applying (9) to $Y_t^{(\ell)}$, and $Y_t^{(L)} \sim s_t$. Since the gradient $\frac{\partial U}{\partial x}$ appears directly in the Langevin equation, the quantity

$$v_t = E_{w_t}\left[\frac{1}{L+1}\sum_{\ell=0}^{L}\left\|\frac{\partial}{\partial y}U(Y_t^{(\ell)};\theta_t)\right\|_2\right],$$

which gives the average image gradient magnitude of U along an MCMC path at training step t, plays a central role in sampling. Sampling at noise magnitude ε will lead to very different behavior depending on the gradient magnitude. If v_t is very large, gradients will overwhelm the noise and the resulting dynamics are similar to gradient descent. If v_t is very small, sampling becomes an isotropic random walk. A valid image density should appropriately balance energy gradient magnitude and noise strength to enable realistic long-run sampling.

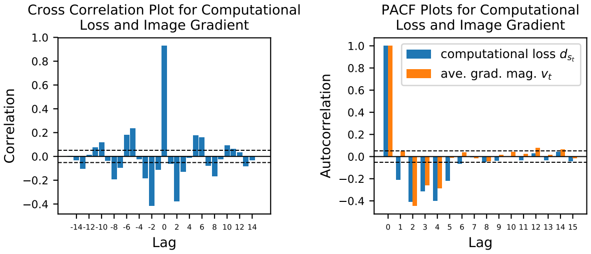


Figure 4: Illustration of expansion/contraction oscillation for a single training implementation. This behavior is typical of convergent *and* non-convergent ML. *Left:* Cross correlation of d_{s_t} (uncentered) and v_t (mean centered). The two are highly correlated at lag 0 and exhibit negative correlation for lag ±3 steps, indicating that expansion updates tend to increase gradient strength in the near future and vice-versa. *Right:* PACF plots of d_{s_t} (uncentered) and v_t (mean centered). Both have a strong negative autocorrelation within the next 4 training batches, showing that expansion updates tend to follow contraction updates and vice-versa.

We empirically observe that expansion and contraction updates tend to have opposite effects on v_t (see Figure 4). Gradient magnitude v_t and computational loss d_{s_t} are highly correlated at the current iteration and exhibit significant negative correlation at a short-range lag. Both have significant negative autocorrelation for short-range lag. This indicates that expansion updates tend to increase v_t and contraction updates tend to decrease v_t, and that expansion updates tend to lead to contraction updates and vice-versa. We believe that the natural oscillation between expansion and contraction updates underlies the stability of ML with (2).

Learning can become unstable when U is updated in the expansion phase for many consecutive iterations if $v_t \to \infty$ as $U(X^+) \to -\infty$ for positive samples and $U(X^-) \to \infty$

for negative samples. This behavior is typical of W-GAN training (informally interpreting the generator as w_t with $L = 0$) and the W-GAN Lipschitz bound is needed to prevent such instability. In ML learning with ConvNet potentials, consecutive updates in the expansion phase will increase v_t so that the gradient can better overcome noise and samples can more quickly reach low-energy regions. In contrast, many consecutive contraction updates can cause v_t to shrink to 0, leading to the solution $U(x) = c$ for some constant c (see Figure 5 right, blue lines). In proper ML learning, the expansion updates that follow contraction updates prevent the model from collapsing to a flat solution and force U to learn meaningful features of the data.

Throughout our experiments, we find that the network can easily learn to balance the energy of the positive and negative samples so that $d_{s_t}(\theta_t) \approx 0$ after only a few model updates. In fact, ML learning can easily adjust v_t so that the gradient is strong enough to balance d_{s_t} and obtain high-quality samples from virtually *any* initial distribution in a small number of MCMC steps. This insight leads to our ML method with noise-initialized MCMC. The natural oscillation of ML learning is the foundation of the robust synthesis capabilities of ConvNet potentials, but realistic short-run MCMC samples can mask the true steady-state behavior of the model, as discussed next.

3.2. Second Axis: MCMC Convergence or Non-Convergence

In the literature, it is expected that the finite-step MCMC distribution s_t must approximately converge to its steady-state p_{θ_t} for learning to be effective. On the contrary, we find that high-quality synthesis is possible, and actually easier to learn, when there is a drastic difference between the finite-step MCMC distribution s_t and true steady-state samples of p_{θ_t}. An examination of ConvNet potentials learned by existing methods shows that in all cases, running the MCMC sampler for significantly longer than the number of training steps results in samples with significantly lower energy and unrealistic appearance. Although synthesis is possible without convergence, it is not appropriate to describe a non-convergent ML model p_{θ_t} as an approximate data density.

Oscillation of expansion and contraction updates occurs for both convergent and non-convergent ML learning, but for very different reasons. In convergent ML, we expect the average gradient magnitude v_t to converge to a constant that is balanced with the noise magnitude ε at a value that reflects the temperature of the data density q. However, ConvNet potentials can circumvent this desired behavior by tuning v_t with respect to the burn-in energy landscape rather than noise ε. Figure 5 shows how average image space displacement $r_t = \frac{\varepsilon^2}{2} v_t$ is affected by noise magnitude ε and number of Langevin steps L for noise, data-based, and persistent MCMC initializations.

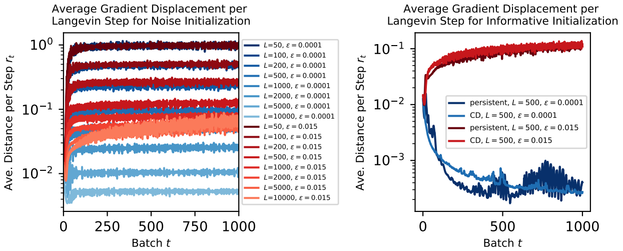


Figure 5: Illustration of gradient strength for convergent and non-convergent ML. With low noise (blue) the energy either learns only the burn-in path (left) or contracts to a constant function (right). With sufficient noise (red), the network gradient learns to balance with noise magnitude and it becomes possible to learn a realistic steady-state.

For noise initialization with low ε, the model adjusts v_t so that $r_t L \approx R$ where R is the average distance between an image from the noise initialization distribution and an image from the data distribution. In other words, the MCMC paths obtained from non-convergent ML with noise initialization are nearly linear from the starting point to the ending point. Mixing does *not* improve when L increases because r_t shrinks in proportion to the increase. Oscillation of expansion and contraction updates occurs because the model tunes v_t to control how far along the burn-in path the negative samples travel. Samples never reach the steady-state energy spectrum and MCMC mixing is not possible.

For data-based initialization and persistent initialization with low ε, we see that $v_t, r_t \to 0$ and that learning tends to the trivial solution $U(x) = c$. This occurs because contraction updates dominate the learning dynamics. At low ε, samples initialized from the data will easily have lower energy than the data since sampling reduces to gradient descent. For persistent learning, the model learns to synthesize meaningful features early in learning and then contracts in gradient strength once it becomes easy to find negative samples with lower energy than the data. Previous authors who trained models with persistent chains use auxiliary techniques such as a Gaussian prior [33] or occasional rejuvenation from noise [6] which prevent unbalanced network contraction, although the role of these techniques is not recognized by the authors. To our knowledge no authors have trained (2) using CD, possibly because the energy can easily collapse to a trivial flat solution.

For all three initialization types, we can see that convergent ML becomes possible when ε is large enough. ML with noise initialization behaves similarly for high and low ε when L is small. For large L with high ε, the model tunes v_t to balance with ε rather than R/L. The MCMC samples complete burn-in and begin to mix for large L, and increas-

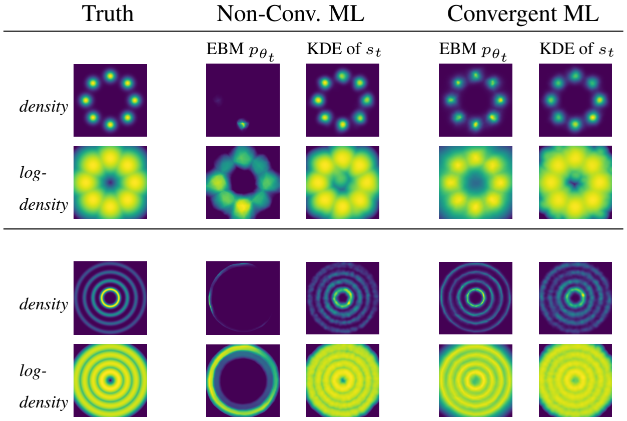


Figure 6: Comparison of convergent and non-convergent ML for 2D toy distributions. Non-convergent ML does not learn a valid density but the kernel density estimate of the negative samples reflects the groundtruth. Convergent ML learns an energy that closely approximates the true density.

ing L will indeed lead to improved MCMC convergence as usual. For data-based and persistent initialization, we see that v_t adjusts to balance with ε instead of contracting to 0 because the noise added during Langevin sampling forces U to learn meaningful features.

3.3. Learning Algorithm

We now present an algorithm for ML learning. The algorithm is essentially the same as earlier work such as [33] that investigates the potential (2). Our intention is not to introduce a novel algorithm but to demonstrate the range of phenomena that can occur with the ML objective based on changes to MCMC sampling. We present guidelines for the effect of tuning on the learning outcome.

- *Noise and Step Size for Non-Convergent ML*: For non-convergent training we find the tuning of noise and step-size have little effect on training stability. We use $\varepsilon = 1$ and $\tau = 0$. Noise is not needed for oscillation because d_{s_t} is controlled by the depth of samples along the burn-in path. Including low noise appears to improve synthesis quality.

- *Noise and Step Size for Convergent ML:* For convergent training, we find that it is essential to include noise with $\tau = 1$ and precisely tune ε so that the network learns true mixing dynamics through the gradient strength. The step size ε should approximately match the local standard deviation of the data along the most constrained direction [22]. An effective ε for 32×32 images with pixel values in [-1, 1] appears to lie around 0.015.

Algorithm 1: ML Learning

```
input : ConvNet potential $U(x;\theta)$, number of training steps $T$, initial weight $\theta_1$, training
        images $\{x_i^+\}_{i=1}^N$, step size $\varepsilon$, noise indicator $\tau \in \{0,1\}$, Langevin steps $L$, learning rate $\gamma$.
output: Weights $\theta_{T+1}$ for energy $U(x;\theta)$.
for $t = 1 : T$ do
    1. Draw batch images $\{X_i^+\}_{i=1}^n$ from training set.
       Draw initial negative samples $\{Y_i^{(0)}\}_{i=1}^m$ from MCMC initialization method (noise or
       informative initialization, see Section 2.3).
    2. Update $\{Y_i^{(0)}\}_{i=1}^m$ with
       $Y_i^{(\ell)} = Y_i^{(\ell-1)} - \frac{\varepsilon^2}{2}\frac{\partial}{\partial y}U(Y_i^{(\ell-1)};\theta_t) + \varepsilon\tau Z_{i,\ell},$
       where $Z_{i,\ell} \sim \mathrm{N}(0, I_N)$, for $L$ steps to obtain negative samples $\{X_i^-\}_{i=1}^m = \{Y_i^{(L)}\}_{i=1}^m$.
    3. Update the weights by $\theta_{t+1} = \theta_t - g(\Delta\theta_t, \gamma)$
       where $\Delta\theta_t$ is the stochastic gradient (8) and $g$ is the SGD or ADAM optimizer.
```

- *Number of Steps*: When $\tau = 0$ or $\tau = 1$ and ε is very small, learning leads to similar non-convergent ML outcomes for any $L \geq 100$. When $\tau = 1$ and ε is correctly tuned, sufficiently high values of L lead to convergent ML and lower values of L lead to non-convergent ML.

- *Informative Initialization:* Informative MCMC initialization is not needed for non-convergent ML even with as few as $L = 100$ Langevin updates. The model can naturally learn fast pathways to realistic negative samples from an arbitrary initial distribution. On the other hand, informative initialization can greatly reduce the magnitude of L needed for convergent ML. We use persistent initialization starting from noise.

- *Network structure*: For the first convolutional layer, we observe that a 3×3 convolution with stride 1 helps to avoid checkerboard patterns or other artifacts. For convergent ML, use of non-local layers [28] appears to improve synthesis realism.

- *Regularization and Normalization*: Previous studies employ a variety of auxiliary training techniques such as prior distributions (e.g. Gaussian), weight regularization, batch normalization, layer normalization, and spectral normalization to stabilize sampling and weight updates. We find that these techniques are not needed.

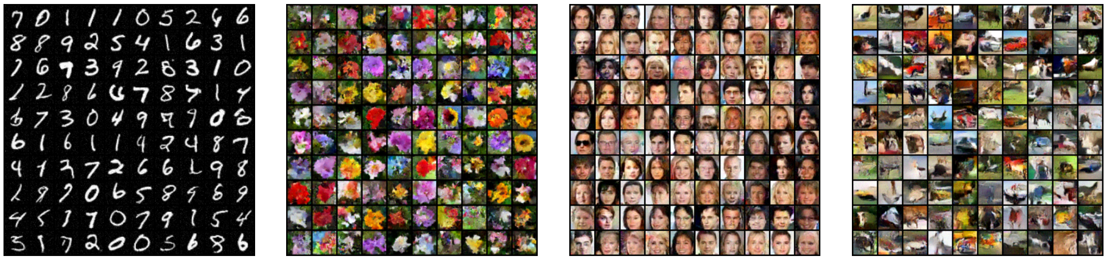


Figure 7: Short-run samples obtained from an energy function trained with non-convergent ML with noise initialization. The images are generated using 100 Langevin updates from uniform noise initialization. Contrary to prior art, informative initialization is not needed for high-quality synthesis. From left to right: MNIST, Oxford Flowers 102, CelebA, CIFAR-10.

- *Optimizer and Learning Rate:* For non-convergent ML, ADAM improves training speed and image quality. Our non-convergent models use ADAM with $\gamma = 0.0001$. For convergent ML, ADAM appears to interfere with learning a realistic steady-state and we use SGD instead. When using SGD with $\tau = 1$ and properly tuned ε and L, higher values of γ lead to non-convergent ML and sufficiently low values of γ lead to convergent ML. See Appendix A for details on tuning the SGD learning rate γ for convergent ML.

4. Experiments

4.1. Low-Dimensional Toy Experiments

We first demonstrate the outcomes of convergent and non-convergent ML for low-dimensional toy distributions (Figure 6). Both toy models have a standard deviation of 0.15 along the most constrained direction, and the ideal step size ε for Langevin dynamics is close to this value [22]. Non-convergent models are trained using noise MCMC initialization with $L = 100$ and $\varepsilon = 0.01$ (too low for the data temperature) and convergent models are trained using persistent MCMC initialization with $L = 500$ and $\varepsilon = 0.125$ (approximately the right magnitude relative to the data temperature). The distributions of the short-run samples from the non-convergent models reflect the ground-truth densities, but the learned densities are sharply concentrated and different from the ground-truths. In higher dimensions this sharp concentration of non-convergent densities manifests as oversaturated long-run images. With sufficient Langevin noise, one can learn an energy function that closely approximates the ground-truth.

4.2. Synthesis from Noise with Non-Convergent ML Learning

In this experiment, we learn an energy function (2) using ML with uniform noise initialization and short-run MCMC. We apply our ML algorithm with $L = 100$ Langevin steps starting from uniform noise images for each update of θ with $\tau = 0$ and $\varepsilon = 1$. We use ADAM with $\gamma = 0.0001$.

Previous authors argued that informative MCMC initialization is a key element for successful synthesis with ML learning, but our learning method can sample from scratch with the same number of Langevin steps. Unlike the models learned by previous authors, our models can generate high-fidelity and diverse images from a noise signal. Our results are shown in Figure 7, Figure 8 (left), and Figure 2 (top). Our recent companion work [24] thoroughly explores the capabilities of non-convergent ML.

4.3. Convergent ML Learning

With the correct Langevin noise, one can ensure that MCMC samples mix in the steady-state energy spectrum throughout training. The model will eventually learn a realistic steady-state as long as MCMC samples approximately converge for each parameter update t beyond a burn-in period t_0. One can implement convergent ML with noise initialization, but we find that this requires $L \approx$ 20,000 steps.

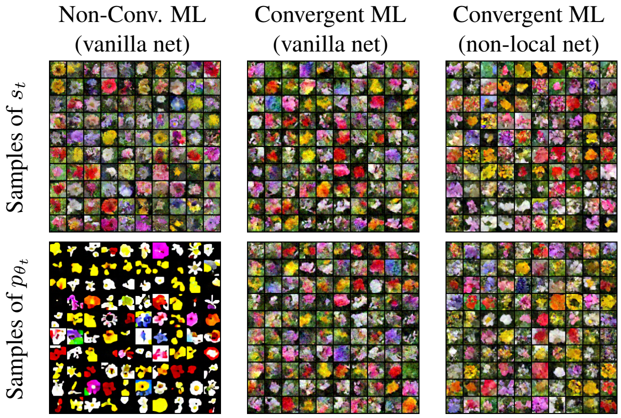


Figure 8: Comparison of short-run negative and steady-state samples. Method: non-convergent ML using noise initialization and 100 Langevin steps (*left*), convergent ML with a vanilla ConvNet, persistent initialization and 500 Langevin steps (*center*), and convergent ML with a non-local net, persistent initialization and 500 Langevin steps (*right*).

Informative initialization can dramatically reduce the number of MCMC steps needed for convergent learning. By using SGD with no momentum and learning rate $\gamma = 0.0005$, noise indicator $\tau = 1$ and step size $\varepsilon = 0.015$, we were able to train convergent models using persistent initialization and $L = 500$ sampling steps. We initialize 10,000 persistent images from noise and update 100 images for each batch. We implement the same training procedure for a vanilla ConvNet and a network with non-local layers [28]. Our results are shown in Figure 8 (middle, right) and Figure 2 (bottom). See Appendix A for additional details on energy initialization for convergent ML.

4.4. Mapping the Image Space

A well-formed energy function partitions the image space into meaningful Hopfield basins of attraction. Following [11], we map the structure of a convergent energy. We first identify many metastable MCMC samples. We then sort the metastable samples from lowest energy to highest energy and sequentially group images if travel between samples is possible in a magnetized energy landscape. This process is continued until all minima have been clustered. Our mappings show that the convergent energy has meaningful metastable structures encoding recognizable concepts (Figure 9).

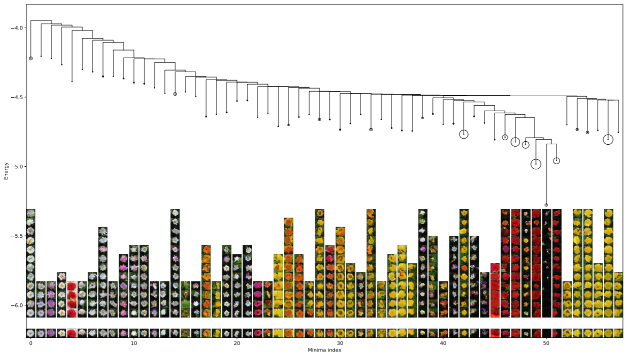


Figure 9: Visualization of basin structure of the learned energy function $U(x)$ for the Oxford Flowers 102 dataset. Columns display randomly selected basins members and circles indicate the total number of basin members. Vertical lines encode basin minimum energy and horizontal lines depict the lowest known barrier at which two basins merge.

5. Conclusion and Future Work

Our experiments on energy-based models with the form (2) reveal two distinct axes of ML learning. We use our insights to train models with sampling capabilities that are unobtainable by previous implementations. The informative MCMC initializations used by previous authors are not necessary for high-quality synthesis. By removing this technique we train the first energy functions capable of high-diversity and realistic synthesis from noise initialization after training. We identify a severe defect in the steady-state distributions of prior implementations and introduce the first ConvNet potentials of the form (2) for which long-run and steady-state samples have realistic appearance. Our observations could be very useful for convergent ML learning with more complex MCMC initialization methods used in [32, 7]. We hope that our work paves the way for future unsupervised and weakly supervised applications with energy-based models.

Acknowledgment

The work is supported by DARPA XAI project N66001-17-2-4029; ARO project W911NF1810296; and ONR MURI project N00014-16-1-2007; and Extreme Science and Engineering Discovery Environment (XSEDE) grant ASC170063. We thank Prafulla Dhariwal and Anirudh Goyal for helpful discussions.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017. 2, 3

[2] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *Journal of Chemical Physics*, 106(4), 1997. 3

[3] T. Chen, E. Fox, and G. C. Stochastic gradient hamiltonian monte carlo. *ICML*, 2014. 4

[4] J. Dai, Y. Lu, and Y.-N. Wu. Generative modeling of convolutional neural networks. *arXiv preprint arXiv:1412.6296*, 2014. 3

[5] Z. Dai, A. Almahairi, P. Bachman, E. Hovy, and A. Courville. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017. 3

[6] Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019. 2, 3, 4, 6

[7] R. Gao, Y. Lu, J. Zhou, S.-C. Zhu, and Y. N. Wu. Learning generative convnets via multi-grid modeling and sampling. *CVPR*, 2018. 1, 3, 4, 9

[8] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6:721–741, 1984. 4

[9] A. G. A. P. Goyal, N. R. Ke, S. Ganguli, and Y. Bengio. Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pages 4392–4402, 2017. 3

[10] T. Han, E. Nijkamp, X. Fang, M. Hill, S.-C. Zhu, and Y. N. Wu. Divergence triangle for joint training of generator model, energy-based model, and inference model. *arXiv preprint arXiv:1812.10907*, 2018. 3

[11] M. Hill, E. Nijkamp, and S.-C. Zhu. Building a telescope to look into high-dimensional image spaces. *QAM*, 77(2):269–321, 2019. 3, 4, 8

[12] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, pages 1771–1800, 2002. 3, 4

[13] G. Hinton, S. Osindero, M. Welling, and Y.-W. Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive science*, 30(4):725–731, 2006. 3

[14] G. E. Hinton. A practical guide to training restricted boltzmann machines. *Tech. Rep. UTML TR 2010-003, Dept. Comp. Sci., Univ. Toronto*, 2010. 3

[15] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982. 3

[16] L. Jin, J. Lazarow, and Z. Tu. Introspective learning for discriminative classification. In *Advances in Neural Information Processing Systems*, 2017. 3

[17] T. Kim and Y. Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016. 3

[18] R. Kumar, A. Goyal, A. Courville, and Y. Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019. 2, 3

[19] J. Lazarow, L. Jin, and Z. Tu. Introspective neural networks for generative modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2783, 2017. 3

[20] K. Lee, W. Xu, F. Fan, and Z. Tu. Wasserstein introspective neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 4

[21] Y. Lu, S. C. Zhu, and Y. N. Wu. Learning frame models using cnn filters. *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 3, 4

[22] R. M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo, Chapter 5*, 2011. 4, 7, 8

[23] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng. Learning deep energy models. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1105–1112, 2011. 3

[24] E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu. On learning non-convergent non-persistent short-run mcmc toward energy-based model. *NeurIPS (to appear)*, 2019. 2, 8

[25] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. *ICML*, pages 1064–1071, 2008. 4

[26] Z. Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 3

[27] D. J. Wales. The energy landscape as a unifying theme in molecular science. *Phil. Trans. R. Soc. A*, 363:357–377, 2005. 3

[28] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *CVPR*, 2018. 7, 8

[29] M. Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128. ACM, 2009. 3

[30] Y. N. Wu, S. C. Zhu, and X. Liu. Equivalence of julesz ensembles and frame models. *International Journal of Computer Vision*, 38(3):247–265, 2000. 3

[31] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, 2018. 3

[32] J. Xie, Y. Lu, and Y. N. Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. *AAAI*, 2018. 1, 3, 4, 9

[33] J. Xie, Y. Lu, S. C. Zhu, and Y. N. Wu. A theory of generative convnet. *International Conference on Machine Learning*, 2016. 1, 3, 4, 6, 7

[34] J. Xie, Z. Zheng, R. Gao, W. Wang, S.-C. Zhu, and Y. Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018. 3

[35] J. Xie, S.-C. Zhu, and Y. N. Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7101, 2017. 3

[36] S.-C. Zhu, Y. N. Wu, and D. Mumford. Filters, random fields and maximum entropy (frame): Toward a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998. 3, 4

A. Energy Initialization and Scale of SGD Learning Rate for Convergent ML

In this section we discuss some details about initializing the energy function and scaling the SGD learning rate. Energy initialization is important for efficient convergent ML but not crucial for non-convergent ML. We find that convergent ML is most effective when r_t (see Section 3.2) has approximately the same order of magnitude throughout training. With noise $\varepsilon = 0.015$, we observe that r_t typically lies in the range [0.08, 0.15] for large t. However, when the initial weights θ_1 come from standard ConvNet initialization, we observe $r_1 \approx 10^{-6}$. To address this we use the scaled energy

$$U(x;\theta) = \frac{F(x;\theta)}{\varepsilon^2/2}, \tag{11}$$

where F is a ConvNet. This is equivalent to using the Langevin update

$$X_{\ell+1} = X_\ell - \frac{\partial}{\partial x} F(X_\ell;\theta) + \varepsilon Z_\ell. \tag{12}$$

When θ_1 is obtained from standard ConvNet initialization and the rescaled energy (11) is used, we observe that

$$r_1 = \left[\frac{1}{L+1} \sum_{\ell=0}^{L} \left\| \frac{\partial}{\partial y} F(Y_t^{(\ell)};\theta_1) \right\|_2 \right] \approx 0.01$$

which is within a reasonable magnitude of the approximate target range [0.08, 0.15]. Additional scaling is required when $r_1 \approx 0.01$ is either too low or high for the ideal noise ε and the target range of r_t but the same principles apply.

We note that the rescaling causes further complications, since the computational loss

$$d_{s_t}(\theta) = \frac{2}{\varepsilon^2} \left(E_q[F(X;\theta_t)] - E_{s_t}[F(X;\theta_t)] \right)$$

now depends on ε. To address this, we find that is helpful to use a scaled learning rate $\gamma = \frac{\varepsilon^2}{2}\gamma_0$ where $\gamma_0 \approx 0.0005$, to obtain the update gradient

$$\gamma \Delta\theta_t = \gamma_0 \left[\frac{\partial}{\partial\theta} \left(\frac{1}{n} \sum_{i=1}^{n} F(X_i^+;\theta_t) - \frac{1}{m} \sum_{i=1}^{m} F(X_i^-;\theta_t) \right) \right] \tag{13}$$

where $\Delta\theta_t$ is given by (8). When using the vanilla SGD update

$$\theta_{t+1} = \theta_t - \gamma\Delta\theta_t, \tag{14}$$

the scale of the parameter change $\|\theta_{t+1} - \theta_t\|_2 = \|\gamma\Delta\theta_t\|_2$ depends only on the scale of $\|\frac{\partial}{\partial\theta} F(x;\theta_t)\|_2$ and the scale of γ_0 and not on the scale of ε. We find that this enables standardized weight initialization and LR tuning that is independent of ε. In practical training of convergent models we implement ML using (11), (12), (13), and (14).