LANDSCAPE LEARNING FOR NEURAL NETWORK INVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many machine learning methods operate by inverting a neural network at inference time, which has become a popular technique for solving inverse problems in computer vision, robotics, and graphics. However, these methods often involve gradient descent through a highly non-convex loss landscape, causing the optimization process to be unstable and slow. We introduce a method that learns a loss landscape where gradient descent is efficient, bringing massive improvement and acceleration to the inversion process. We demonstrate this advantage on a number of methods for both generative and discriminative tasks, including GAN inversion, adversarial defense, and 3D human pose reconstruction.

1 INTRODUCTION

Many inference problems in machine learning are formulated as inverting a forward model $F(x)$ by optimizing an objective over the input space x. This approach, which we term optimization-based inference (OBI), has traditionally been used to solve a range of inverse problems in vision, graphics, robotics, recommendation systems, and security (Hernandez et al., 2008; Lee & Kuo, 1993; Domke, 2012; Brakel et al., 2013; Stoyanov et al., 2011; Cremer et al., 2019). Recently, neural networks have emerged as the paramterization of choice for forward models (Loper et al., 2015; Pavlakos et al., 2019; Abdal et al., 2019; Menon et al., 2020; Yu et al., 2021; Wang et al., 2019; Chen et al., 2021a; Zhang et al., 2021), which can be pretrained on large collections of data, and inverted at testing time in order to solve inference queries.

Optimization-based inference (OBI) has a number of advantages over feed-forward or encoder-based inference (EBI). Since there is no encoder, OBI provides flexibility to adapt to new tasks, allowing one to define new constraints into the objective during inference. When observations are partially missing, OBI can adapt without additional training. Moreover, OBI naturally supports generating multiple and diverse hypotheses when there is uncertainty. Finally, OBI has intrinsic advantages for robustness, both adapting to new data distributions as well as defending against adversarial examples.

However, the key bottleneck for OBI in practice is the computational efficiency and the speed of inference. Feed-forward models are fast because they only require a single forward pass of a neural network, but OBI requires many (often hundreds) steps of optimization in order to obtain strong results for one example. Forward models in OBI are often trained with generative or discriminative tasks, but they are not trained for the purpose of performing gradient descent in the input space. Fig. 8 visualizes the loss landscape for uncurated examples. The loss landscape is not guaranteed to be an efficient path from the initialization to the solution, causing the instability and inefficiency.

In this paper, we propose a framework to accelerate and stabilize the inversion of forward neural networks. Instead of optimizing over the original input space, we will learn a new input space such that gradient descent converges quickly. Our approach uses an alternating algorithm in order to learn the mapping between these spaces. The first step collects optimization trajectories in the new space until convergence. The second step updates the mapping parameters to reduce the distance between the convergence point and each point along the trajectory. By repeating these steps, our approach will learn a mapping between the spaces that allows gradient descent over the input to converge in significantly fewer steps.

Empirical experiments and visualizations on both generative and discriminative models show that our method can significantly improve the convergence speed for optimization. We validate our

approach on a diverse set of computer vision tasks, including GAN inversion (Abdal et al., 2019), adversarial defense (Mao et al., 2021), and 3D human pose reconstruction (Pavlakos et al., 2019). Our experiments show that our method converges an order of magnitude faster without loss in absolute performance after convergence. As our approach does not require retraining of the forward model, it can be compatible to all existing OBI methods with a differentiable forward model and objective function.

The primary contribution of this paper is an efficient optimization-based inference framework. In Sec. 2, we survey the related literature to provide an overview of forward model inversion problem. In Sec. 3, we formally define OBI (3.1); our method to learn an efficient loss landscape for OBI (3.2); a training algorithm for better generalization and robustness (3.3) . In Sec. 4, we experimentally study and analyze the effectiveness of the mapping network for OBI. We will release all code and models.

2 Related Work

The different approaches for inference with a neural network can be partitioned into either encoder-based inference, which is feedforward, or optimization-based inference, which is iterative. We briefly review these two approaches in the context of our work. In addition, we also review representative work in meta-learning and discuss the similarities and differences with our work.

2.1 Encoder-based Inference

Encoder-based inference trains a neural network F to directly map from the output space to the input space. Auto-encoder based approach (Pidhorskyi et al., 2020) learns an encoder that map the input data to the latent space. Richardson et al. (2021); Tov et al. (2021); Wei et al. (2021); Perarnau et al. (2016) learn an encoder from the image to the latent space in GAN. Encoder based inference requires training the encoder on the anticipated distribution in advance, which is often less effective and can fail on unexpected samples (Dinh et al., 2021; Kang et al., 2021). In addition, encoder-based method can only produce one reconstruction for under-constrained inverse problems (Yeh et al., 2017), even if multiple outcomes can all be true given the partial input.

2.2 Optimization-based Inference

OBI methods perform inference by solving an optimization problem with gradient-based methods such as Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) and Projected Gradient Descent (PGD) (Madry et al., 2017). In these cases, the objective function specifies the inference task. Besides these methods which use a point estimate for the latent variable, one can estimate the posterior distribution of the latent variable through Bayesian optimization, such as SGLD (Welling & Teh, 2011).

Gradient based optimization methods have been used to infer the latent code of query samples in deep generative models like GANs (Goodfellow et al., 2014) via GAN inversion (Karras et al., 2020; Jahanian et al., 2019; Shen et al., 2020; Zhu et al., 2016; Abdal et al., 2019; 2020; Bau et al., 2019; Huh et al., 2020; Pan et al., 2021). Style transfer relies on gradient based optimization to change the style of the input images (Jing et al., 2019). It can also create adversarial attacks that fool the classifier (Croce & Hein, 2020; Carlini & Wagner, 2017; Mao et al., 2019; Szegedy et al., 2013). Recently, back-propagation-based optimization has shown to be effective in defending adversarial examples (Mao et al., 2021) and compressed sensing (Bora et al., 2017). Wu et al. (2019) uses MAML (Finn et al., 2017) to accelerate the optimization process in compressed sensing.

Recently, constrained optimization was popularized for text-to-image synthesis by Crowson et al. (2022); Liu et al. (2021). They search in the latent space to produce an image that has the highest similarity with the given text as measured by a multi-modal similarity model like CLIP (Radford et al., 2021). Test-time constrained optimization is also related to the idea of 'prompt-tuning' for large language models. Lester et al. (2021) learns "soft prompts" to condition frozen language models to perform specific downstream tasks. Soft prompts are learned through backpropagation to incorporate signal from just a few labeled examples (few-shot learning).

A major challenge for optimization-based inference is how to perform efficient optimization in a highly non-convex space. To address this, input convex model (Amos et al., 2017) was proposed so that gradient descent can be performed in a convex space. Tripp et al. (2020) introduced a method

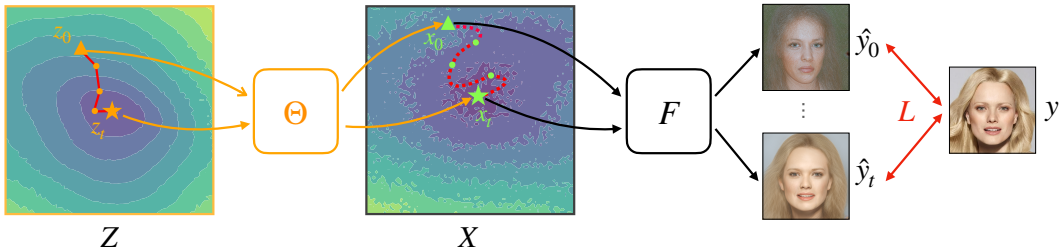


Figure 1: **Method**. The left and middle figure show the loss landscape for our latent space and the original latent space, respectively. While walking to the optimal solution in a few steps is hard in X space, it can be done in our learned loss landscapes.

to retrain the generative model such as it learns a latent manifold that is easy to optimize. When the model cannot be changed and retrained, bidirectional inference (Wang et al., 2019) and hybrid inference (Zhu et al., 2016; 2020) uses an encoder to provide a good initialization for the optimization-based inference in a non-convex space. Our method does not retrain the generative model, but instead maps a new latent space into the original input space, allowing more efficient optimization.

2.3 Meta-Learning

Given a distribution of tasks, meta-learning aims to adapt quickly when presented with a previously unseen task. MAML (Finn et al., 2017) and related methods (Finn et al., 2018; 2019) propose a method to learn a parameter initialization of a neural network by differentiaing through the fine-tuning process. To reduce the computational cost of MAML due to the 2nd degree gradient, Nichol et al. (2018) proposes a first-order meta-learning algorithm. Unlike MAML which include modifying the forward model (e.g. in Wu et al. (2019)), our approach is able to keep the forward model fixed in order to maintain its learned rich priors. Instead of meta-learning of model initialization, a line of work (Andrychowicz et al., 2016; Ravi & Larochelle, 2016; Chen et al., 2021b; Wichrowska et al., 2017) proposed to learn an optimizer, often in the form of an RNN, in order to accelerate gradient descent. Unlike learned optimizers that try to create better optimization algorithms, our approach instead learns to remap the loss landscape which is compatible with any choice of optimizer, including standard SGD or learned ones.

3 Learning Landscapes for Efficient Inference

We present our framework to learn an efficient loss landscape for optimization-based inference (OBI) methods. In Sec. 3.1, we will define OBI. In Sec. 3.2, we will introduce our framework as well as the training objective. In Sec. 3.3, we will describe how to train our model with an alternating optimization algorithm and an experience-replay buffer.

3.1 Optimization-based Inference

Let $F(\mathbf{x}) = \hat{y}$ be a differentiable forward model that generates an output $\hat{y}$ given an input variable $\mathbf{x} \in X$. For example, $\hat{y}$ might be an image, and $\mathbf{x}$ might be the latent variables for a generative model. Given an observation y, the goal of OBI is to find the input $\hat{\mathbf{x}} \in X$ such that an objective function $L(\hat{y}, y)$ is minimized. Formally, we write this procedure as:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in X}{\operatorname{argmin}}\, L(F(\mathbf{x}), y) \tag{1}$$

When the objective function L and the model F are both differentiable, we can perform the optimization over input space X with gradient descent.

3.2 Remapping the Input Space

Instead of operating in the original input space X, we will create a new space Z where gradient descent is efficient and converges in a small number of iterations. We will use a neural network $\Theta : Z \to X$ that maps from the new space Z to the original space X. The parameters of the mapping

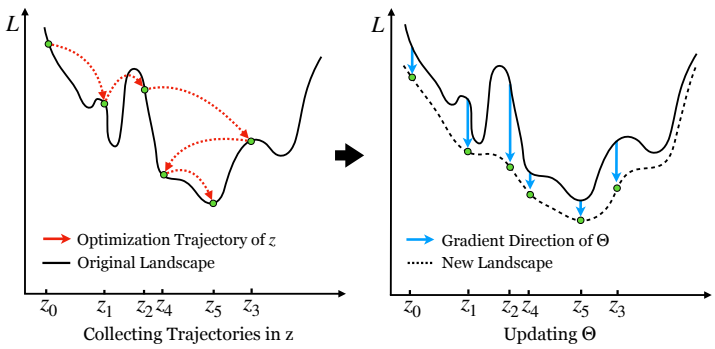


Figure 2: **Landscape Learning.** An optimization trajectory $\{\mathbf{z}_t\}_{t=0}^{5}$ collected is used to train Θ. $\mathbf{z}_i$ that corresponds to a higher L_i will yield a higher gradient when training Θ. Optimization over multiple steps along the trajectory causes Θ to learn *patterns* of trajectories and create a more efficient loss landscape.

Θ is the vector θ. The learning problem we need solve is to estimate θ so that there is a short gradient descent path in Z from the initialization to the solution. Fig. 1 shows an overview of this setup.

We formulate an objective by rolling out the gradient updates on $\mathbf{z}$, where we write $\mathbf{z}_t \leftarrow \mathbf{z}_{t-1} + \lambda \frac{\partial L}{\partial \mathbf{z}_{t-1}}$ as the tth update with a step size of λ. For gradient descent in space Z to be efficient, the goal of our introduced Θ is to move every step $\mathbf{z}_t$ as close as possible to the global minima:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \ \mathbb{E}_{(\mathbf{z},y)} \left[\sum_{t=0}^{T} L(F(\Theta(\mathbf{z}_t)), y) \right] \quad \text{where} \quad \mathbf{z}_t = \begin{cases} 0, & t = 0 \\ \mathbf{z}_{t-1} + \lambda \frac{\partial L}{\partial \mathbf{z}_{t-1}}, & t > 0 \end{cases} \tag{2}$$

We visualize this process with a toy example in Fig. 2. Gradient updates on Θ w.r.t multiple steps $\mathbf{z}_t$ along a trajectory will cause the loss value on each step to be lowered. By learning the parameters of Θ across many examples, Θ can learn the patterns of optimization trajectories in X. For example, Θ can learn to estimate the step size in X and dynamically adjust it to each example. Moreover, Θ can learn to reduce non-convexity of the landscape.

Once we obtain $\hat{\Theta}$ with parameters $\hat{\theta}$, we do inference on a new example y through the standard optimization-based inference procedure, except in Z space now. Given the observation y, we find the corresponding $\hat{\mathbf{x}}$ through the optimization:

$$\hat{\mathbf{x}} = \hat{\Theta}(\hat{\mathbf{z}}) \quad \text{where} \quad \hat{\mathbf{z}} = \underset{\mathbf{z} \in Z}{\operatorname{argmin}} \ L(F(\hat{\Theta}(\mathbf{z})), y) \tag{3}$$

When the inverse problem is under-constrained, one can infer multiple hypotheses for $\hat{\mathbf{x}}$ by repeating the above optimization multiple times with a different random initialization for $\mathbf{z}_0$.

3.3 Training

We use alternating optimization (AO) in order to train Θ jointly with estimating $\mathbf{z}$ for each example in the training set. Specifically, we first fix parameters of Θ and collect N optimization trajectories of $\mathbf{z}$, each with length T. Adopting the same terminology from the robotics community for learning on continuous states Mnih et al. (2013), we term this an *experience replay buffer*. Subsequently, we randomly sample data from this buffer and train Θ to optimize the loss function. We alternate between these two steps for a fixed number of iterations with gradient descent for both. Depending on the application, the training time for Θ varied from one hour to one day using a four GPU server. Please see the appendix for more implementation details.

We also experimented with an online version of the training algorithm, where we update parameters of Θ immediately after one update to $\mathbf{z}$. However, in our experiments, we found this resulted in a slower convergence rate. We show these comparisons in the ablation experiments.

4 Experiments

The goal of our experiments is to analyze how well our proposed method can be applied to various existing OBI methods to improve the optimization efficiency. We demonstrate application of our method to three diverse OBI methods in computer vision, including both generative models and discriminative models. For each OBI method, the inference-time optimization objective of the

Algorithm 1 Learning Mapping Network Θ

Input: Ground truth y, step size λ_z and λ_θ, number of buffers B, number of data samples in a buffer N, number of optimization steps per sample T, loss function L, and forward model F.
Output: Learned mapping network Θ
Randomly initialize a mapping network Θ with parameters θ
for $b = 1, ..., B$ **do**
  Initialize Experience Replay Buffer $[\{\mathbf{z}_{t,i}\}_{t=1}^{T}]_{i=1}^{N}$
  for $i = 1, ..., N$ **do**
    $\mathbf{z}_0 \leftarrow \mathbf{0}$
    for $t = 1, ..., T$ **do**
      $\mathbf{z}_t \leftarrow \mathbf{z}_{t-1} + \lambda_z \frac{\partial}{\partial \mathbf{z}_{t-1}} L(F(\Theta(\mathbf{z})), y)$
      $\mathbf{z}_{t,i} \leftarrow z_t$
    end for
  end for
  for $j = 1, ..., T \cdot N$ **do**
    Randomly sample $\mathbf{z}$ from previously collected buffer
    $\theta_j \leftarrow \theta_{j-1} + \lambda_\theta \frac{\partial}{\partial \theta_{j-1}} L(F(\Theta(\mathbf{z})), y)$
  end for
end for
Return θ =0

Collecting optimization trajectory in Z while keeping θ fixed

Updating θ with previously collected latents

baseline and ours can be written as:

$$\text{Baseline: } \hat{\mathbf{x}} = \min_{\mathbf{x} \in X} L(F(\mathbf{x}), y), \qquad \text{Ours: } \hat{\mathbf{z}} = \min_{\mathbf{z} \in Z} L(F(\Theta(\mathbf{z})), y) \tag{4}$$

Next, we provide the specific implementation of the loss term L for each application, along with quantitative and qualitative results. We also perform experiments to understand the loss landscape in Sec. A.1 and perform ablations on different parts of our approach in Sec. 4.4.

4.1 GAN Inversion

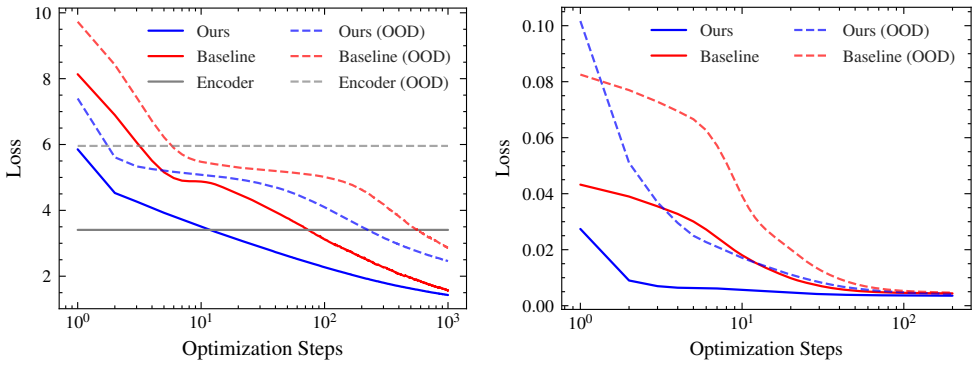
We first validate our method on StyleGAN inversion (Abdal et al., 2019). We take a pretrained generator of StyleGAN Karras et al. (2019) denoted as F. Let y be an observed image whose input variable we are recovering, we optimize the objective of Eq. 4 where the loss can be written as,

$$L(\hat{y}, y) = L_{lpips}(\hat{y}, y) + ||\hat{y} - y||_2^2 \tag{5}$$

where L_{lpips} is a perceptual similarity loss introduced in Zhang et al. (2018), $\hat{y} = F(\hat{x})$ for baseline and $\hat{y} = F(\Theta(\hat{z}))$ for ours. We train Θ on the train split of CelebA-HQ (Karras et al., 2017) dataset and evaluate on CelebA-HQ validation split for in-distribution experiments and LSUN-Cat(Yu et al., 2015) for distribution shifting (OOD) experiments. We compare the results from our method against the state-of-the-art encoder-based GAN inversion model (Richardson et al., 2021).

Quantitative Results. From Fig. 3a, we see that in all experiments, optimization in our space Z consistently outperforms the baseline from the first optimization step to after convergence. This gap in performance is even larger when evaluated on OOD data. This suggests that the improvement in performance is not caused by memorizing the training data. Note that our image reconstruction performance after only 2 steps of optimization is able to outperform or be on-par with the baseline at 20 steps of optimization, resulting in a 10-fold improvement in efficiency. Even after convergence (after 2000 optimization steps), our reconstruction performance improves over the baseline by 15% for in-distribution evaluation and 10% for OOD evaluation. When compared with encoder-based GAN inversion (Richardson et al., 2021), our method achieves better reconstruction after 11 steps of optimization for in-distribution data and 3 steps for OOD data.

Qualitative Results. From Fig. 7, we can see that our method already shows improvements on in-distribution data - it can almost perfectly reconstruct details like fine hair strands, the cap on the person's head, the object in the person's mouth as well-as the text on it. Interestingly, our method is able to discover and reconstruct latents for cats while the encoder-based model fails miserably as shown in Fig 7. The performance on OOD data truly highlights the benefits of our method. We also



(a) GAN Inversion

(b) 3D Human Pose Reconstruction

Figure 3: **Optimization Performance.** We visualize the trends of optimization performance compared with the baseline. In **GAN Inversion (Left)**, we evaluate all models on test splits of CelebA-HQ (Karras et al., 2017) and LSUN-cat (Yu et al., 2015) (OOD) with loss defined in Eq. 5. Since encoder-based inference doesn't involve optimization, we use a flat line to represent it. We perform 2000 steps of gradient descent for all models except encoder-based models. In **3D Human Pose Reconstruction (Right)**, we evaluate all models on test splits of GRAB (Taheri et al., 2020) and PROX (Hassan et al., 2019) (OOD) with loss defined in Eq. 6. We perform 200 steps of gradient descent for all models. For each step, we plot the average loss value of test splits.

visualize how the face generations evolve over the process of optimization in Fig. 4. We can see that in just 4 steps, our method is already able to reconstruct coarse features such as face orientation, skin tone and hair color, while the baseline has hardly deviated from the initialization in regard to any of these features. Further, in Fig. 6 we visualize reconstructions from partial observations where only the center of the face (row 1) or everything other than the mouth (row 2) is visible. We can see a variety of feasible possibilities for the hidden regions (e.g., different hairstyles, lip colors, expressions, etc) showcasing the diversity of the new latent space.

4.2 3D Human Pose Reconstruction

In addition to image generation, our framework also works for 3D reconstruction. For this, we use VPoser (Pavlakos et al., 2019) – a variational autoencoder (Kingma & Welling, 2013) that has learnt a pose prior over body pose. VPoser was trained on SMPL-X body pose parameters $y \in \mathbb{R}^{63}$ obtained by applying MoSh (Loper et al., 2014) on three publicly available human motion capture datasets: CMU Mocap CMU, training set of Human3.6M (Ionescu et al., 2014), and the PosePrior dataset (Akhter & Black, 2015).

We take a pretrained VPoser decoder denoted as F. Our trained mapping network Θ projects a vector from the new input space $\mathbf{z} \in Z$ to a vector in the original VPoser decoder's input space $\mathbf{x} \in X$. Similar to GAN Inversion, we optimize the objective of Eq. 4, where the loss function between predicted and ground truth pose parameters is,

$$L(\hat{y}, y) = ||\hat{y} - y||_2^2 \tag{6}$$

where $\hat{y} = F(\hat{x})$ for the baseline and $\hat{y} = F(\Theta(\hat{z}))$ for ours. For training Θ, we use the GRAB dataset (Taheri et al., 2020) which contains poses of humans interacting with everyday objects. We construct splits for novel video sequences – thus the test split will contain a seen human subject but a potentially unseen pose / demonstration by that subject. We evaluate on this test split for in-distribution experiments and on the PROX dataset (Hassan et al., 2019) for OOD experiments, which contains poses of humans interacting in 3D scenes (e.g., living room, office, etc).

Quantitative Results. For SMPL-X human pose reconstruction experiment, the results follow a similar trend as GAN inversion, with massive improvement in both convergence speed and final loss values after convergence (see Fig. 3b). Our method outperforms the baseline by 19% for in-distribution evaluation and 11% for OOD evaluation.

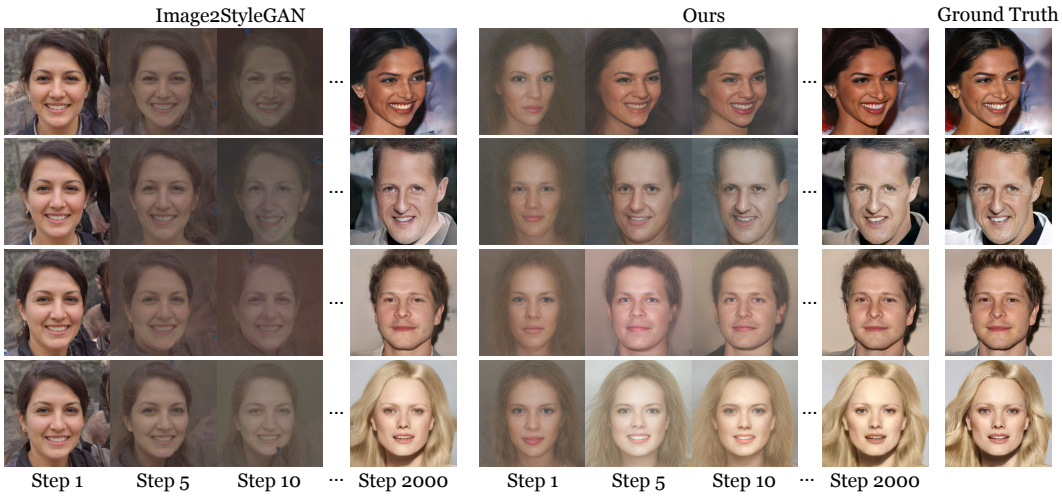


Figure 4: **Optimization Process for GAN Inversion.** Comparing optimization process of our method and the baseline in order to reconstruct the ground truth image. **Left** shows the results from the baseline where optimization is done in the original input space X. **Middle** shows the results from our method where optimization is done in our space Z. **Right** column contains the ground truth image to each example. Each row corresponds to the same example.

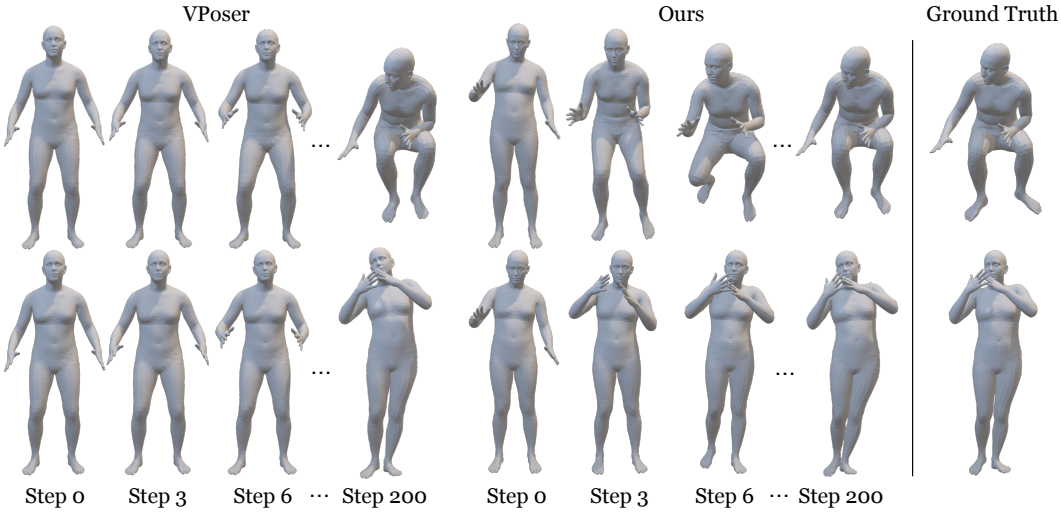


Figure 5: **Optimization Process for 3D Human Pose Reconstruction.** Results shown are for out-of-distribution PROX dataset for sitting (**Top**) and standing (**Bottom**) poses.

Qualitative Results. In Fig. 5 we visualize how the human pose reconstructions evolve over the process of optimization. Here, we observe that the reconstructions from steps 0 to 6 of the baseline are similar for both examples. On the other hand, our method caters to fast convergence for the varying examples, highlighting the general, yet efficient properties of our search space. Further, in Fig. 6 we visualize reconstructions from partial observations where the only joints visible are that of the upper body (row 3) or lower body (row 4). We obtain a wide range of feasible possibilities for the hidden joints demonstrating the diversity of the latent space.

4.3 Defending Adversarial Attacks

Our method is also applicable to discriminative models. A state-of-the-art defense (Mao et al., 2021) for adversarial attack optimizes the self-supervision task at inference time, such that the model can dynamically restore the corrupted structure from the input image for robust inference. Following the existing algorithm implementation, we optimize the input image via our method by minimizing the

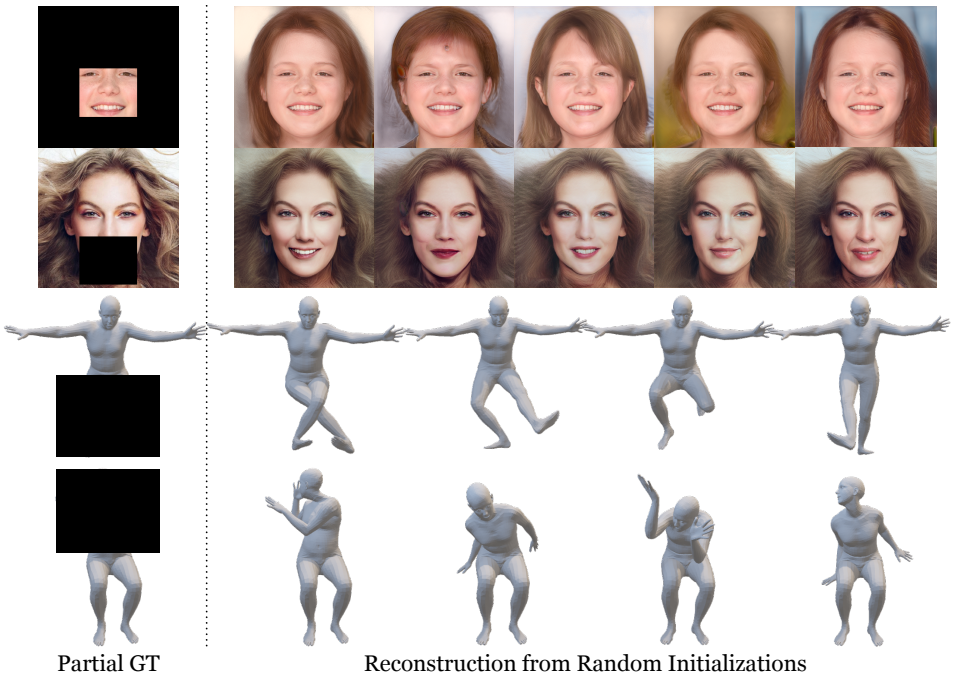


Figure 6: **Diversity of Masked Reconstructions.** We visualize reconstructions for partially observable inputs from random initializations. The masked regions are not considered for loss computation, i.e., the gradient is set to be zero. By optimizing only on the partial observation, we obtain diverse, feasible solutions for the hidden regions. 5 or 4 out of 20 most representative samples are presented.

following discriminative loss function:

$$L(F(\mathbf{r}+\mathbf{a}), y) = L(F(\Theta(\mathbf{z})+\mathbf{a}), y) = \mathbb{E}_{i,j}\left[-y_{ij}^{(s)} \log \frac{\exp(\cos(\mathbf{f}_i, \mathbf{f}_j))}{\sum_k \exp(\cos(\mathbf{f}_i, \mathbf{f}_k))}\right] + \lambda||\Theta(\mathbf{z})||_2^2, \quad (7)$$

where $\mathbf{a}$ is the adversarial attacks that we aim to defend against, $\mathbf{r} = \Theta(\mathbf{z})$ is our additive defense vector to optimize, $\mathbf{f}_i$ are the contrastive features produced by the neural network F from the ith image instance, and $\mathbf{y}_{ij}^{(s)}$ is the indicator for the positive pairs and negative pairs.

After obtaining the mapping network Θ and the input variable $\hat{\mathbf{z}}$, the prediction is $\hat{\mathbf{y}} = F'(\mathbf{a}+\Theta(\mathbf{z}))$, where F' is the classification model. Note that the a self-supervision loss is optimized as a proxy for increasing the robust classification accuracy. In addition, we add a L_2 norm decay term for the generated noise z to avoid generating reversal vector that is too large.

	Optimization Steps						
	No	1 step		3 steps		5 steps	
Model Type	Optimization	Baseline	Ours	Baseline	Ours	Baseline	Ours
RO	31.99	34.62	**44.65**	36.77	**44.23**	38.38	**43.43**
AWP	35.61	39.54	**51.39**	42.81	**51.67**	44.96	**51.05**
MART	35.66	39.77	**51.77**	42.50	**51.77**	45.42	**50.96**
SemiSL	29.78	34.53	**52.11**	37.27	**51.23**	40.93	**49.83**

Table 1: Experiment on improving adversarial robust accuracy. Our goal is to defend 200 steps of $L_2 = 256/255$ norm bounded attack (Madry et al., 2017), where the attack's step size is $64/255$. Our baseline is the SOTA test-time optimization-based defense (Mao et al., 2021), which minimizes the loss of self-supervision task.

Quantitative Results. We evaluate our method on four popular pretrained robust models (Rice et al., 2020; Wang et al., 2020; Wu et al., 2020; Carmon et al., 2019) on CIFAR-10 dataset. The results in Mao et al. (2021) require many steps to optimize the objective to improve the adversarial robustness, which slows down the inference by hundreds of times than the original forward pass. Ideally, we desire test-time optimization that can adapt to the attacked images in just one step, causing

Figure 7: **Comparison Against Encoder-Based Inference. Left** shows the results on the test split of CelebA-HQ; **Right** shows the results on the LSUN-cat dataset.

	Number of Steps	Full Model	Without AO and Buffer	Random Θ	Baseline
In-distribution	20	3.082	3.583	4.417	4.456
OOD	20	4.932	5.127	5.135	5.292
In-distribution	200	1.964	3.034	2.723	2.569
OOD	200	3.498	4.756	3.823	4.617

Table 2: **Ablation Study on mapping network.** Number of Steps indicates the number of optimization steps performed during inference.

the minimal delay at inference time. In Table 1, our method outperforms the gradient descent method in Mao et al. (2021) by up to 18% robust accuracy using a single step, providing a real-time robust inference method. Note that our method converges after 1 step of optimization, demonstrating the effectiveness of our approach.

4.4 Ablation Study

In this section, we present an ablation study by removing the proposed alternating optimization (AO) scheme and the experience replay buffer. We also compare against a Θ that is randomly initialized. From Table 2 in appendix, we discovered that for in-distribution, AO and training of Theta improves the optimziation performance by 14% and 30% respectively. Such gap becomes 35% and 28% for evaluation on 200 steps. For OOD data, the advantage is furthered enlarged as shown in Table 2.

One surprising result we discovered experimentally is that OBI under a randomly initialized mapping network Θ consistently outperforms the baseline. We believe this is due to the fact that adding a Gaussian distribution to an underlying latent distribution of StyleGAN is beneficial in smoothing out loss landscape, making it easier to perform OBI. Similar random projection can also be found in Wu et al. (2019).

5 Conclusion

This paper presents a method to accelerate optimization-based inference to invert a forward model. We propose an approach that learns a new space that is easier than the original input space to optimize with gradient descent at testing time. Our experiments and analysis on three different applications in computer vision have shown that by learning this mapping function, optimization becomes more efficient and generalizes better to out-of-distribution data. Through quantitative and qualitative analysis, we found that such improvement in optimization performance comes from a more efficient loss landscape. Since optimization-based inference has many advantages over encoder-based inference, we believe methods to accelerate them will have many impacts in a variety of applications.

6 ETHICS STATEMENT

Optimization-based inference has a wide variety of applications broadly in computer vision, robotics, natural language processing, assistive technology, security, and healthcare. Since our proposed method provides significant acceleration to these inference techniques, we expect our work to find positive impact in these applications, where speed, accuracy, and robustness are often critical.

Our algorithm is compatible with many different types of forward models – as long as a gradient can be calculated – including neural networks. However, learned forward models are known to acquire and contain biases from the original dataset. Our approach will also inherit the same biases. While our approach to inference offers some advantages to out-of-distribution data, it does not mitigate nor correct biases. The datasets in our experiments are likely not representative of the population, and consequently also contain biases. We acknowledge these limitations, and applications of this method should be mindful of these limitations, especially in potentially sensitive scenarios. As the community continues to address biases in models and datasets, our method can be applied to these models in the future too.

As authors of the paper, we have carefully reviewed and will adhere to the code of ethics provided at `https://iclr.cc/public/CodeOfEthics`.

7 REPRODUCIBILITY STATEMENT

We will release all code, models, and data. Here we provide the implementation details.

7.1 GAN INVERSION

Mapping network Θ is implemented with a 3-layer MLP. The input dimension (dimension of Z space) is the same as the output dimension (dimension of X space). For each intermediate output, we apply a Leaky ReLu function with a negative slope of 0.2 as an activation function. Hidden dimension of the MLP is 1024. For optimizing z (collecting optimization trajectories), we use an Adam optimizer with a learning rate of 0.1. For training the mapping network Θ, we use an AdamW optimizer with a weight decay of 0.1 with a learning rate of 0.0001. We used the following parameter set, $T = 20$, $N = 256$, $B = 500$. For baseline, we use the implementation of Abdal et al. (2019) provided here. The pretrained weights of StyleGAN converted to PyTorch are also provided in the same link.

7.2 3D HUMAN POSE RECONSTRUCTION

Mapping network Θ is implemented with a 3-layer MLP. The input dimension is 128 (dimension of Z) and the output dimension is 32 (dimension of X). For each intermediate output, we apply a Leaky ReLu function with a negative slope of 0.2 as an activation function. Hidden dimension of the MLP is 512. For optimizing z (collecting optimization trajectories), we use an Adam optimizer with a learning rate of 0.1. For training the mapping network Θ, we use an AdamW optimizer with a weight decay of 0.1 and a learning rate of 0.005. We use the following parameter set, $T = 200$, $N = 40960$, $B = 500$. For a fair comparison with the baseline, we tried a range of learning rate for $\mathbf{x} \in X$ $\{0.5, 0.1, 0.05, 0.01, 0.001\}$ and select the best performing configuration for comparison.

7.3 DEFENDING ADVERSARIAL ATTACKS

Mapping network Θ is implemented with a 3-layer MLP. The input dimension is 3072 (dimension of Z) and the output dimension is 3072 (dimension of X). For each intermediate output, we apply a Leaky ReLu function with a negative slope of 0.2 as an activation function. Hidden dimension of the MLP is 3072. For optimizing z (collecting optimization trajectories), we use an Adam optimizer with learning rate $0.2/255$. For training the mapping network Θ, we use an AdamW optimizer with learning rate of 0.0001 and a weight decay of 0.1. We use the following parameter set, $T = 5$, $N = 5120$, $B = 70$. For the regularization term λ that constrains the amplitude of the additive defense vector, we use $\lambda = 1$. We use random start instead of zero start for initializing the attack reversal vector.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4432–4441, 2019.

Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan++: How to edit the embedded images? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ijaz Akhter and Michael Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. 06 2015. doi: 10.1109/CVPR.2015.7298751.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4502–4511, 2019.

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pp. 537–546. PMLR, 2017.

Philémon Brakel, Dirk Stroobandt, and Benjamin Schrauwen. Training energy-based models for time-series imputation. *The Journal of Machine Learning Research*, 14(1):2771–2797, 2013.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, pp. 39–57, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/32e0bd1497aa43e02a42f47d9d6515ad-Paper.pdf`.

Boyuan Chen, Robert Kwiatkowski, Carl Vondrick, and Hod Lipson. Full-body visual self-modeling of robot morphologies. *arXiv preprint arXiv:2111.06389*, 2021a.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021b.

CMU. Cmu mocap dataset.

Jochen L Cremer, Ioannis Konstantelos, and Goran Strbac. From optimization-based machine learning to interpretable security rules for operation. *IEEE Transactions on Power Systems*, 34(5): 3826–3836, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Katherine Crowson, Stella Rose Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *ArXiv*, abs/2204.08583, 2022.

Tan M Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. Hyperinverter: Improving stylegan inversion via hypernetwork. *arXiv preprint arXiv:2112.00719*, 2021.

Justin Domke. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*, pp. 318–326. PMLR, 2012.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. *Advances in neural information processing systems*, 31, 2018.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2282–2292, 2019.

Carlos Hernandez, George Vogiatzis, and Roberto Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

Minyoung Huh, Richard Zhang, Jun-Yan Zhu, Sylvain Paris, and Aaron Hertzmann. Transforming and projecting images into class-conditional generative networks. In *European Conference on Computer Vision*, pp. 17–34. Springer, 2020.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014. doi: 10.1109/TPAMI.2013.248.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019.

Kyoungkook Kang, Seongtae Kim, and Sunghyun Cho. Gan inversion for out-of-range images with geometric transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13941–13949, 2021.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kyoung Mu Lee and C-CJ Kuo. Shape from shading with a linear triangular element surface model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8):815–822, 1993.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *ArXiv*, abs/2104.08691, 2021.

Xingchao Liu, Chengyue Gong, Lemeng Wu, Shujian Zhang, Haoran Su, and Qiang Liu. Fusedream: Training-free text-to-image generation with improved clip+gan space optimization. *ArXiv*, abs/2112.01573, 2021.

Matthew Loper, Naureen Mahmood, and Michael J. Black. Mosh: Motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6), nov 2014. ISSN 0730-0301. doi: 10.1145/2661229.2661273. URL https://doi.org/10.1145/2661229.2661273.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Chengzhi Mao, Mia Chiquier, Hao Wang, Junfeng Yang, and Carl Vondrick. Adversarial attacks are reversible with natural supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 661–671, 2021.

Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pp. 2437–2445, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10975–10985, 2019.

Guim Perarnau, Joost Van De Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. *arXiv preprint arXiv:1611.06355*, 2016.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning, 2020.

Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2287–2296, 2021.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252, 2020.

Veselin Stoyanov, Alexander Ropson, and Jason Eisner. Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 725–733. JMLR Workshop and Conference Proceedings, 2011.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. Grab: A dataset of whole-body human grasping of objects. In *European conference on computer vision*, pp. 581–600. Springer, 2020.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 33:11259–11272, 2020.

Hao Wang, Chengzhi Mao, Hao He, Mingmin Zhao, Tommi S Jaakkola, and Dina Katabi. Bidirectional inference networks: A class of deep bayesian networks for health profiling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 766–773, 2019.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Lu Yuan, Gang Hua, and Nenghai Yu. A simple baseline for stylegan inversion. *ArXiv*, abs/2104.07661, 2021.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Olga Wichrowska, Niru Maheswaranathan, Matthew W Hoffman, Sergio Gomez Colmenarejo, Misha Denil, Nando Freitas, and Jascha Sohl-Dickstein. Learned optimizers that scale and generalize. In *International Conference on Machine Learning*, pp. 3751–3760. PMLR, 2017.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

Yan Wu, Mihaela Rosca, and Timothy Lillicrap. Deep compressed sensing. In *International Conference on Machine Learning*, pp. 6850–6860. PMLR, 2019.

Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5485–5493, 2017.

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through gan inversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1768–1777, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European conference on computer vision*, pp. 592–608. Springer, 2020.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pp. 597–613. Springer, 2016.

APPENDIX

A VISUALIZATIONS

A.1 LOSS LANDSCAPE

To understand the underlying cause of the significant improvement in optimization brought by our mapping network θ, we visualize in Fig. 8 the loss landscape for performing optimization in the original input space X, and our projected space Z. To generate this visualization, we first perform 20 steps of optimization on the validation dataset to collect a set of recovered latents. We then perform principle component analysis (PCA) on these recovered latents to obtain two principle directions. Finally, for individual examples, we evaluate the loss for vertices on a meshgrid spanned by the two principle directions where the center is the last step of optimization.

From the visualization, we can see that the loss landscapes of the baseline are highly non-convex and contain points whose loss are significantly higher than its neighboring regions, while our loss landscapes are significantly smoother, with the "spikes" removed. Besides, our loss landscapes also tend to be steeper than the baseline ones. These two phenomena directly cause our method to perform gradient descent faster and stabler.

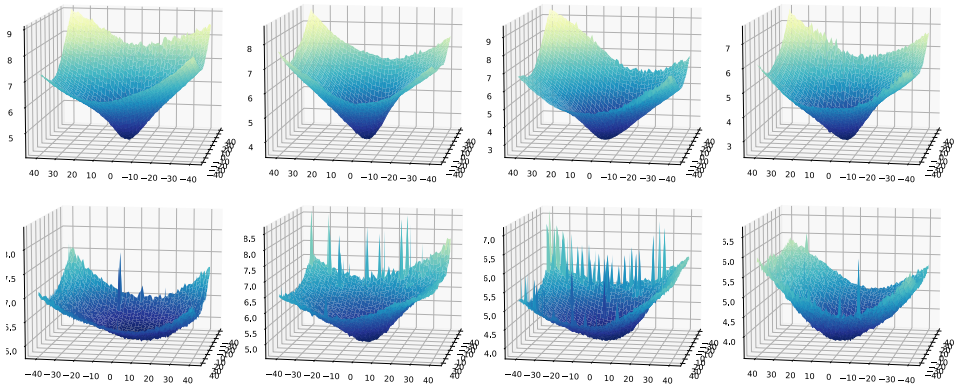


Figure 8: **Visualizing Loss Landscape (Uncurated).** Visualizing the loss landscape of StyleGAN inversion spanned by two principle directions. **Top** row shows 4 examples of the loss landscapes corresponding to our space Z. **Bottom** row shows the loss landscapes corresponding to the original input space X for the same 4 examples. Note that the top row and the bottom row have different minimum loss values on the landscape because the minimas are obtained by performing independent optimization runs in space Z and X respectively. Given a fixed number of optimization step, optimization in Z reaches lower loss values than optimization in X.

B LIMITATIONS

Optimization-based inference has intrinsic advantages to robustness, accuracy, and flexibility, which comes at the cost of additional computation time during inference. Encoder-based methods will usually perform faster because they only require a single forward pass of a neural network, while our approach requires several computational passes in both the forward and backward (gradient) direction. We believe that for many applications this trade-off will be desirable, especially in cases where accuracy is more important than speed. Our approach aims to minimize this additional computational overhead brought by optimization-based inference, and our experiments on multiple datasets show the significant computational savings compared to other optimization-based inference methods.

Unlike many other optimization-based inference algorithms, our approach also requires a training step in order to fit a suitable landscape, which requires both training time and training data. However, we believe this overhead is insignificant for most applications and we have designed our neural networks to be efficient. For example, Θ is relatively lightweight, making its training time fairly marginal compared to the training of the forward model F. In all our experiments, we found that the training time of Θ is orders of magnitudes faster than the training time for F.

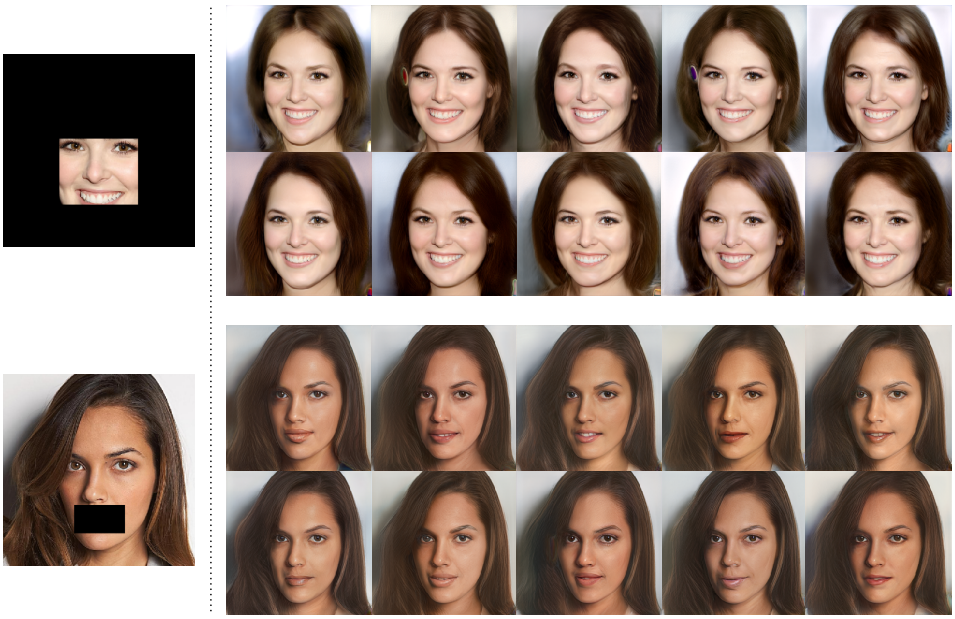


Figure 9: **Diversity of Masked Reconstructions.(Uncurated)** We visualize reconstructions for partially observable inputs from random initialization. The masked regions are not considered for loss computation, i.e., the gradient is set to be zero. By optimizing only on the partial observation, we obtain diverse, feasible solutions for the hidden regions. All reconstruction results presented are randomly sampled.

PSNR (dB) ↑							
Optimization Steps		1 step	2 steps	10 steps	20 steps	200 steps	2000 steps
In Distribution	Baseline	10.837	11.214	11.272	12.471	17.310	21.874
	Ours	**12.423**	**13.148**	**15.396**	**16.343**	**19.892**	**22.082**
OOD	Baseline	9.689	10.124	10.169	10.773	13.146	16.860
	Ours	**11.445**	**11.461**	**12.585**	**12.822**	**15.874**	**18.236**
MSE ($\times 10^{-2}$) ↓							
Optimization Steps		1 step	2 steps	10 steps	20 steps	200 steps	2000 steps
In Distribution	Baseline	8.817	8.214	8.405	6.247	2.105	0.740
	Ours	**6.217**	**5.315**	**3.265**	**2.643**	**1.200**	**0.732**
OOD	Baseline	11.558	10.615	10.833	9.153	5.403	2.591
	Ours	**7.879**	**7.906**	**6.071**	**5.755**	**3.115**	**1.972**
Perceptual Loss (VGG) Abdal et al. (2019) ↓							
Optimization Steps		1 step	2 steps	10 steps	20 steps	200 steps	2000 steps
In Distribution	Baseline	6.807	5.985	4.772	4.393	2.548	1.297
	Ours	**4.466**	**4.214**	**3.414**	**3.056**	**1.952**	**1.100**
OOD	Baseline	8.306	7.326	5.339	5.200	4.563	2.396
	Ours	**5.533**	**5.257**	**5.006**	**4.874**	**3.467**	**2.168**

Table 3: Further evaluation for GAN inversion experiments. In addition to the loss objective defined by 5, we further evaluate our results with 3 metrics including PSNR, MSE, Percuptual Loss with VGG features Abdal et al. (2019). These evaluations indicate the same trend we observed previously.