Learned Optimizer with Diffusion Models

Chaitanya Patel*
chpatel1@stanford.edu

Apoorv Srivastava*
apoorv1@stanford.edu

Abstract

Inverse problems are ubiquitous in engineering applications and often are ill-conditioned i.e. they frequently have one-to-many mapping. The deterministic optimization-based techniques for solving inverse problems result in a single estimate that often corresponds to a suboptimal local minima. These problems are better suited to a probabilistic interpretation under which a probability distribution over the solution space effectively captures the many-to-one relations. Denoising Diffusion Probabilistic Models (DDPMs) have emerged as a powerful technique to learns high-dimensional probability distribution from data. Sampling from *conditional* DDPM involves starting with a pure noise sample and iteratively leveraging a denoising process to reconstruct samples from the *conditional* data distribution that aligns with the provided condition. The iterative nature of the denoising process behaves similarly to the iterative optimization techniques, and additionally facilitates the learning of a prior over the solution space. This provides a principled way to sample multiple solutions of an ill-conditioned inverse problem. In addition, classifier-free guidance provides a novel way to control the alignment between the generated sample and the input condition. In this project, we explore the use of conditional DDPMs to learn optimizers for solving such ill-conditioned inverse problems. We show encouraging results on a challenging problem of estimating 3D human pose from input 2D keypoints. Additionally, we introduce a novel transformer-based design that significantly enhances the capacity of DDPMs compared to conventional MLPs for this inverse problem. Our implementation can be found at here as well as here.

1 Introduction

Inverse problems play a crucial role in bridging the gap between mathematical models and observational data, with numerous applications in engineering sciences such as medical imaging, remote sensing, and signal processing. The primary objective of inverse problems is to determine the input(s) that result in a given output when passed through a forward operator, which maps inputs to outputs. However, for high-dimensional problems, the inverted mapping from outputs to inputs often exhibits a one-to-many behavior, making it challenging to address using traditional optimization-based techniques[8] that typically identify only a single input for any given output. Moreover, if the forward model is complex, the optimization process can become trapped in a local minima, leading to a suboptimal solution. Additionally, the forward model is often computationally expensive, making it impractical to carry out the optimization process repeatedly to identify the set of all possible inputs.

Alternatively, the ill-conditioning of inverse problems can be addressed by adopting a probabilistic interpretation of the problem. The goal is to learn a distribution over the input space conditioned on the output value, where the support of the conditional distribution is ideally restricted to the points that lie in the solution set of the inverse problem. However, this requires access to the probability distribution over inputs and outputs, which is very high-dimensional and intractable with classical tools.

*Equal contribution

Project Report for Deep Generative Models (CS236), Stanford University.

Deep generative models, such as GANs, VAEs, and auto-regressive models[11, 7, 2], have shown great promise in learning high-dimensional (conditional) probability distributions using an offline dataset. These learned distributions can later be effectively used for sampling from the conditional distribution of inputs given some output. This is also desirable since, under an expensive and complex forward model, the learned distribution can act as a guide for learning an optimizer that can exploit the prior over the forward model and the data.

Recently, Denoising Diffusion Probabilistic Models (DDPMs)[18, 9] have demonstrated remarkable success in modeling complex data distributions with exceptionally high sample quality. These models are particularly promising for our problem because they offer a natural means of controlling the alignment between generated samples and provided conditions through classifier-free guidance[10], unlike other generative models. Specifically, the weight parameter associated with the classifier-free guidance enables us to regulate the degree to which generated samples conform to the given condition. This feature is especially useful in scenarios where the observation is noisy(or accurate), as it allows the generated samples to rely more(or less) on the unconditional prior.

We evaluate our hypothesis and present findings on the challenging task of predicting 3D human pose[25] from input 2D keypoints. Estimating 3D human pose (represented by the 3D angles of body joints) from 2D keypoints detected from an image is a classical vision problem of great significance, and it is appropriate for our case since the inverse mapping from 2D keypoints to 3D pose is a complex one-to-many mapping prone to poor local minima. We demonstrate that an optimizer based on DDPM can learn this mapping using a 3D motion capture dataset. Furthermore, we show that we can regulate the degree of 'fit' between the generated 3D pose and input 2D keypoints using classifier-free guidance. We also show that expressing the body pose as a high-dimensional vector of combined joint rotations makes it challenging to regularize the model since it is prone to learning spurious correlations from the dataset. To tackle this problem, we introduce a novel transformer-based[20] architecture that significantly improves the modeling capabilities of the DDPM while utilizing substantially fewer model parameters.

2 Related Work

Several prior works have studied such problems from various viewpoints. The limitations of standard gradient-based optimizers like Adam, RMSProp, AdaDelta, etc., which are designed to work in a problem-agnostic way, are studied in [1, 3, 21, 16, 5]. These studies suggest training an optimizer model to enhance update steps tailored for a particular set of optimization problems. Several research efforts have framed this objective as an inverse problem, wherein the model tries to anticipate the true input signal from a corrupted observation [24] under an assumed prior over the true signal. However, these optimization methods only output one sample from the distribution of possible solutions. Precious studies, such as [13], have employed normalizing flows to generate distributions for inverse problems. Given the improved generative modeling capabilities shown by DDPMs, it is natural to investigate the application of DDPMs in similar contexts. It is worth noting that all conditional generative models (such as conditional VAE, conditional GAN, etc.) could potentially serve as solutions for this problem. However, our specific focus is on conditional DDPMs[9] as they have shown superior modeling power with their iterative generation process, and classifier-free guidance[10] provides an intuitive way to control the effect of conditioning over generated samples.

Some earlier studies have looked into similar directions. An optimization technique based on diffusion models was presented in [6]. This method enforces physics constraints on DDPMs by aligning the trajectory of the DDPM with that of a physics-based model, and serves as a motivation to undertake this challenge. However, like other optimization-based methods, it result in a single optimal estimate and fails to capture the one-to-many relationship of the inverse mapping. Additionally, the effectiveness of the proposed method is constrained by the ability to obtain the physics-based optimization trajectory, as noted by the authors.

Diffusion Optimization Models (DDOM) are introduced in [12] to address the ill-conditioning of the inverse problems. The DDOM learns to reproduce the level sets of a function, demonstrating its ability to address ill-posed inverse problems. Another method termed Bayesian Algorithm Execution (BAX) introduced in [17] can be used to identify level sets under a function evaluation budget given an algorithm to compute the level sets, which in our case can potentially be obtained through DDOM.

However, DDOM focused on black-box optimization and showed results on simpler inverse problems. Our focus for this project is to explore DDPMs for complex inverse problems like 3D pose estimation.

3 Method

3.1 Problem Setup

Given a sophisticated forward model $f : \mathbb{R}^m \to \mathbb{R}^n$ and $y \in \mathbb{R}^n$, the goal of the inverse problem is to find the level set $\mathcal{X} = \{x \in \mathbb{R}^m | f(x) = y\}$, or equivalently to learn a mapping $g = f^{-1}$ such that $\mathcal{X} = g(y) = f^{-1}(y)$. Classical optimization-based techniques find the solution of

$$\min_x \text{objective}(y, x, f)$$

where objective(y, x, f) is usually $\|y - f(x)\|_2^2$ with an optional prior over x. The optimization is often an iterative gradient-based method which is initialized with a random x_0 and finds one solution after convergence. When f is a many-to-one mapping, different initializations can find different solutions, however carrying out multiple optimizations till convergence through a complex forward model is expensive. More importantly, there is no principled way to control the distribution $p(x|y)$ produced by the optimization for a given initialization distribution $p(x_0)$. For high-dimensional problems, learning a prior over x (to guide the optimization) itself is a hard problem.

On the other hand, conditional generative models based on GAN, VAE, score-based models, etc. have shown great results in modeling $p_\theta(x|y)$ with parameters θ from a dataset of $\{x_i, y_i\}$. Following this, the inverse problem of finding $\mathcal{X}$ can be reformulated as finding the support of the conditional distribution $p(x|y)$ which can be learned from the dataset of $\{x_i, y_i\}$ where $y_i = f(x_i)$. The values of $p(x|y)$ signify the probabilities of different values of $x \in \mathcal{X} = g(y)$ and could potentially inform the uncertainties associated with processes that subsequently use the identified inputs.

3.2 Score-based Generative Models

Score-based generative models[19] represent $p(x)$ as the score function $s_\theta(x) \approx \nabla \log_x p(x)$. Stochastic gradient Langevin dynamics[23] can be used to produce samples from $p(x)$ using $\nabla_x \log p(x)$ using an iterative markovian process shown below.

$$x_i = x_{i-1} + \frac{\delta}{2}\nabla_x \log p(x_{i-1}) + \sqrt{\delta}\epsilon_i, \quad \text{where } \epsilon_i \sim \mathcal{N}(0, I)$$

3.3 Denoising Diffusion Probabilistic Models

Denoising Diffusion Probabilistic Models(DDPMs)[9] are a class of score-based generative models inspired by non-equilibrium thermodynamics. Given a datapoint from a distribution $x_0 \sim p(x)$, the forward diffusion process (not to be confused with our forward model f) adds increasing amounts of Gaussian noise to produce a sequence of noisy samples $(x_1, ..., x_T)$ where x_T is almost equivalent to pure noise (isotropic Gaussian distribution).

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$
$$p(x_{1:T}|x_0) = \prod_{t=1}^{T} p(x_t|x_{t-1})$$

where β_t's are dependent on the noise schedule.

Under reasonable assumptions, the forward diffusion process is reversible and the reverse diffusion process $p(x_{t-1}|x_t)$ can be used to generate samples of $p(x)$ starting from $x_T \sim \mathcal{N}(\mathbf{0}, I)$. Each step of the reverse diffusion process can be learned by a model $p_{\theta_t}(x_{t-1}|x_t)$ using any generative modeling approach. If it is modeled using the score function $s_{\theta_t}(x_t)$, [22] showed that learning this model is equivalent to learning a denoising function $\epsilon_{\theta_t}(x_t)$ which predicts the noise ϵ_t added during the forward diffusion. [9] proposed DDPM as a single model $\epsilon_\theta(x_t, t)$ can learn this whole reverse process using a simple denoising objective.

$$L = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\|\epsilon_t - \epsilon_\theta(x_t, t)\|^2\right]$$

Refer to the original DDPM paper[9] for more details.

3.4 Conditional DDPMs

[4] showed that the conditional generation using DDPM can be done by using the gradient of the condition $\nabla_x \log p(y|x_t)$ to guide the denoising. In particular, using the Bayes rule,

$$\begin{aligned}\nabla_{x_t} \log p(x_t, y) &= \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t) \\ -\lambda_t \bar{\epsilon}_\theta(x_t, t) &\approx -\lambda_t \epsilon_\theta(x_t, t) + w \nabla_{x_t} \log f_\phi(y|x_t)\end{aligned}$$

where $-\lambda_t$ is appropriate scaling between score objective and denoising objective. $\bar{\epsilon}$ is a new classifier-guided predictor. The weight parameter w controls the extent of condition guidance during sampling. Setting $w = 0$ will recover unconditional generative model.

Note that $\log p(x_t)$ is unconditional generative model discussed above. $\nabla_{x_t} \log p(y|x_t)$ nudges the updates to align more with the condition y. [4] showed conditional generation on MNIST by using a pretrained classifier $c_\phi(y|x_t, t)$ as a proxy for $p(y|x_t)$. However, note that $p(y|x)$ in our case is simply the forward model $y = f(x)$. This means that if our forward model f is accessible and differentiable, then we can train an unconditional DDPM on x and then use the gradient of the forward model to guide the sampling towards the provided condition y. Although this formulation works for MNIST conditional generation where the condition is a discrete class label, it is not straightforward to define $\nabla_{x_t} \log p(y|x_t)$ when f is a many-to-one continuous mapping. For many-to-one mapping f, $p(y|x)$ is non-zero only when $y = f(x)$. In this case, $\nabla_{x_t} \log p(y|x_t)$ fails to provide meaningful guidance to align the samples with the condition y.

3.5 Classifier-free guidance

[10] showed that it is possible to do conditional generation without classifier guidance. Using Bayes rule,

$$\begin{aligned}\nabla_{x_t} \log p(y|x_t) &= \nabla_{x_t} \log p(x_t|y) - \nabla_{x_t} \log p(x_t) \\ &= -\lambda_t \Big(\epsilon_\theta(x_t, t, y) - \epsilon_\theta(x_t, t) \Big)\end{aligned}$$

Here, $\epsilon_\theta(x_t, t, y)$ is a predictor trained with condition input and $\epsilon_\theta(x_t, t)$ is unconditional predictor. Following the classifier-guided formulation above, the new classifier would be

$$\begin{aligned}-\lambda_t \bar{\epsilon}_\theta(x_t, t, y) &= -\lambda_t \epsilon_\theta(x_t, t, y) + w \nabla_{x_t} \log p(y|x_t) \\ \bar{\epsilon}_\theta(x_t, t, y) &= (w+1)\epsilon_\theta(x_t, t, y) - w\epsilon_\theta(x_t, t)\end{aligned}$$

This formulation doesn't contain any dependency on the classifier. It requires conditional predictor $\epsilon_\theta(x_t, t, y)$ and an unconditional predictor $\epsilon_\theta(x_t, t)$. In practice, unconditional predictor is trained along with $\epsilon_\theta(x_t, t, y)$ by providing null conditioning $y = \emptyset$. Intuitively, each update is pushed towards the conditional prediction aligning with y and away from unconditional prediction aligning with any random y. This classifier-free formulation allows us to use it for our learned optimizer of inverse problem. The value of w controls the extent to which our solution x aligns with y. This provides a novel way to dynamically control the optimization based on our confidence in the observation y. Throughout our experiments, we use this conditional DDPM with classifier-free guidance to learn the optimizer for inverse problem.

4 Toy Experiments

The initial experiments with conditional DDPMs were carried out on the MNIST dataset a tutorial from (https://github.com/TeaPearce/Conditional_Diffusion_MNIST/tree/main). The trade-off between sample diversity and conditioning was studied by varying the mixing parameter and we were able to reproduce the baseline results. The samples generated from the conditional DDPM corresponding to different conditioning are shown in Fig. 1.

Next, as a toy example of a *continuous* forward model $y = f(x)$, we studied the negative of standard 2D Branin function in domain $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$ following [12].

$$f(x_1, x_2) = -a(x_2 - bx_1^2 + cx_1 - r)^2 - s(1-t)\cos(x_1) - s, \tag{1}$$

with $a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, r = 6, s = 10$, and $t = \frac{1}{8\pi}$. As shown in Fig. 2, it has three global minimas (maximas in the original Branin function) with convoluted contour lines and is a good test

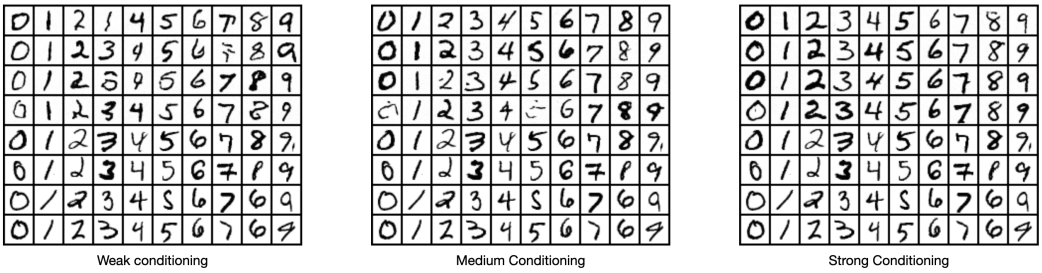

Figure 1: MNIST samples from conditional DDPMs corresponding to different conditioning strengths.

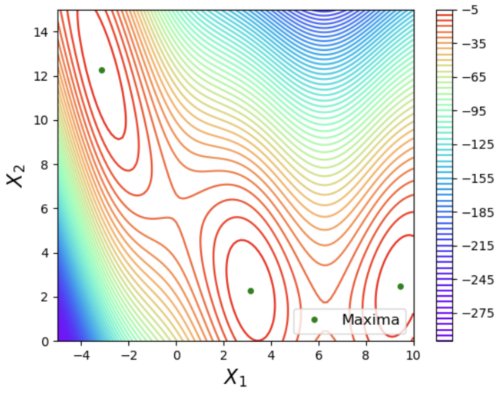

Figure 2: Negative 2D Branin Function.

case to study the inverse problem. We were able to train a simple MLP based model to generate samples $x \in \mathbb{R}^2$ for a given $y \in \mathbb{R}$. Note that the conditional distribution $p(x|y)$ is represented by the contour line in Fig. 2. After verifying our implementation with these toy examples, we explored our hypothesis with a much more challenging problem of 3D human pose estimation.

5 3D Human Pose Estimation

To test our method on a real-world application, we use standard 2D-to-3D pose estimation setting. SMPL [14] body model is a standard 3D body model in pose estimation literature as shown in Fig. 3. Given body pose $x \in \mathbb{R}^{24\times3}$ denoting the 3D rotations of 24 body joints, it transforms a template 3D body mesh in that pose and computes 3D locations of 25 keypoints. 3D keypoints can be projected onto an image plane using camera parameters c to get 2D keypoints $y \in \mathbb{R}^{25\times2}$. For this project, we fix camera parameters c and define x-to-y mapping as a forward model $y = f(x)$. The optimization problem is to find the set of *reasonable* body poses x from input 2D keypoints y as a conditional distribution $p(x|y)$.

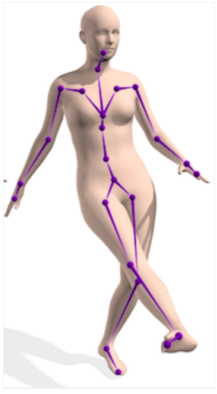

Figure 3: SMPL 3D body model we use for our 3D pose estimation problem.

Note that this is a complex function involving forward kinematics and perspective camera project - both of which reduces the degrees of expressiveness. Multiple 3D poses x can map to the same 2D keypoints y. For example, one particular pose x_1 is shown with corresponding 2D keypoints y in Fig. 3. It is easy to see that there exists another pose x_2 where the right leg leans backward of the body plane (instead of forward as shown in the figure) which will give the same 2D keypoints. However such x_2 is not practical despite aligning perfectly with y. The right shoulder of the body can rotate along the straight right hand axis while still maintaining the same set of 2D keypoints. A classical

optimization technique to find x for a given y will easily get trapped into a local minima. Thus, a probabilistic learned optimization based on conditional DDPM is suitable for this task.

5.1 Dataset & Metric

We use AMASS dataset[15] which is a large motion capture database containing diverse motions (body pose sequence) on the SMPL body model. It combines multiple other motion capture datasets under the same SMPL representation and contains more than 11K motions and 4M frames. We train our models on the frames of train sequences and report our metrics on the frames of validation sequences. Each joint rotation $x_j \in \mathbb{R}^3$ is converted into an equivalent 6D representation following [26]. For evaluation, we use the mean absolute difference between the input 2D keypoints y and the projected 2D keypoints $f(x)$ of the generated poses x, normalized by the image size. We also show qualitative results of the generated poses.

5.2 Architecture

Since x and y can be represented as vectors, our DDPM $\epsilon(x_t, y, t)$ can be implemented as a simple MLP model. This may work on simpler problems but we show that it leads to suboptimal results on 3D pose estimation for two main reasons: (1) Forward mapping x-to-y is very skewed. Rotation of torso joint affects all keypoints of the upper body whereas the rotation of palm only affects the palm joint. (2) Concatenating rotations of all joints in a single vector suffers from the curse of dimensionality. Any model on such representation is prone to learning spurious correlations from the dataset and hence requires careful tuning of heavy regularization, leading to worse performance.

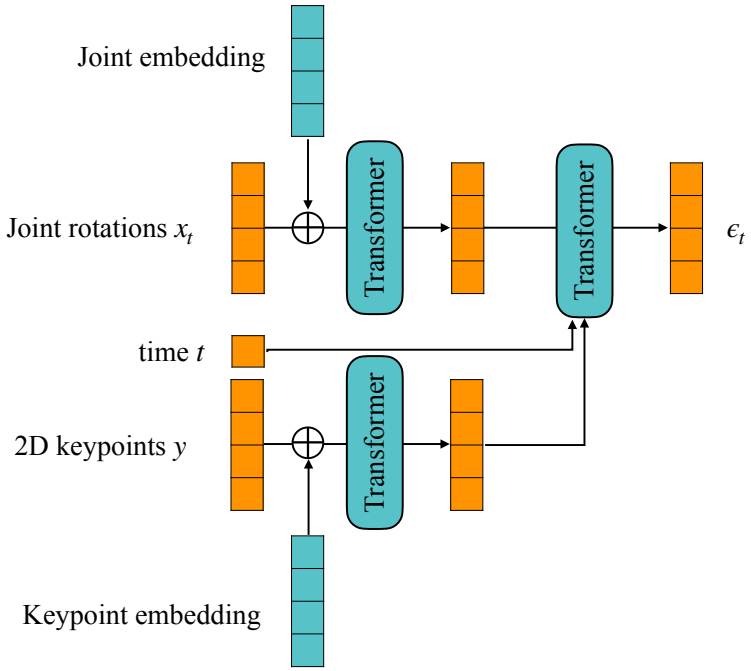

Figure 4: Architecture of our transformer-based DDPM for 3D human pose estimation.

To solve this problem, we propose a transformer-based architecture where each joint rotation $x_i \in \mathbb{R}^3$ and each 2D keypoint $y_j \in \mathbb{R}^2$ is considered a separate token. As shown in Fig. 5, the learnable joint embeddings are added to the sequence of joint tokens and are encoded by the transformer encoder. Similarly, the keypoint tokens are also encoded by a separate transformer encoder. Then a transformer decoder with several decoder layers decodes the joint tokens with cross-attention from time embedding and keypoint embedding. As opposed to the top-down architecture of an MLP where the model has all the information available from the first layer (in form of concatenated vector), the transformer has a bottom-up architecture where the 'local' tokens are processed to get a global view of the pose and keypoint configuration. This enables parameter efficient learning as we will show that

a transformer model with 11M parameters significantly outperforms a well-tuned MLP architecture with 27M parameters.

5.3 Results

Model	$w = 0.0$	$w = 0.5$	$w = 2.0$
MLP	0.382	0.223	0.241
Transformer	0.080	0.068	0.058

Table 1: Pixel alignment error for MLP and transformer DDPM with different alignment weights.

We show our quantitative results in Table 1. We compute the pixel alignment metric (as elaborated in 5.1) for MLP based DDPM (27M parameters) and our novel transformer based DDPM (11M parameters). DDPM models are trained once and evaluated by carrying out inference on the validation set with three different values of the weight parameter w. We can see that the transformer model achieves significantly lower alignment error than the MLP model. We also noticed that transformer based model was robust to different sets of hyper parameters whereas MLP model required some careful tuning. Transformer model was also very fast to converge during training. As we increase the value of w, we can see that both models show increasing alignment (and hence decreasing alignment error). This confirms our hypothesis that the DDPM based learned optimizer indeed allows us to control the degree of alignment.

We show quantitative results of our best transformer-based model at the end. To limit the pdf size, we are including only few results in this pdf. Please check more results are our code repository. We can see that the DDPM based learned optimizer can indeed generate plausible body poses x corresponding to the input 2D keypoints y. As we increase the weight parameter w, the extent of alignment increases. In many cases, even though the generated pose is different from the groundtruth, the generated pose is also plausible and aligns with the input 2D keypoints. In almost all cases, the model avoids generating weird poses (with extreme rotations of joints). This confirms our hypothesis that the model has learned the prior over the plausible set of poses and can generate many poses aligning with the input while avoiding implausible poses.

6 Conclusion & Future Work

In this project, we explore DDPMs based learned optimizers for complex inverse problems. Going beyond prior works[12], we analyze the performance of DDPM on a more challenging task of 3D pose estimation. We qualitatively and quantitatively prove the effects of the weight parameter of classifier-free guidance, and show that DDPM provides a unique way to control the degree of alignment while solving inverse problems. In addition, we also propose a novel transformer-based architecture for 3D pose estimation that generates significantly better samples with half the number of model parameters.

In future, we plan to explore the use of DDPM on other inverse problems. For 3D pose estimation, we can increase the difficulty of the problem by also estimating the camera parameters c along with body pose x. This will require re-designing the transformer based architecture to incorporate camera parameters c. Going beyond keypoints, we plan to directly use image of a person as conditioning instead of 2D keypoints. However, this will require significant research efforts to source appropriate dataset and design DDPM architecture.

References

[1] 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.

[2] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models.

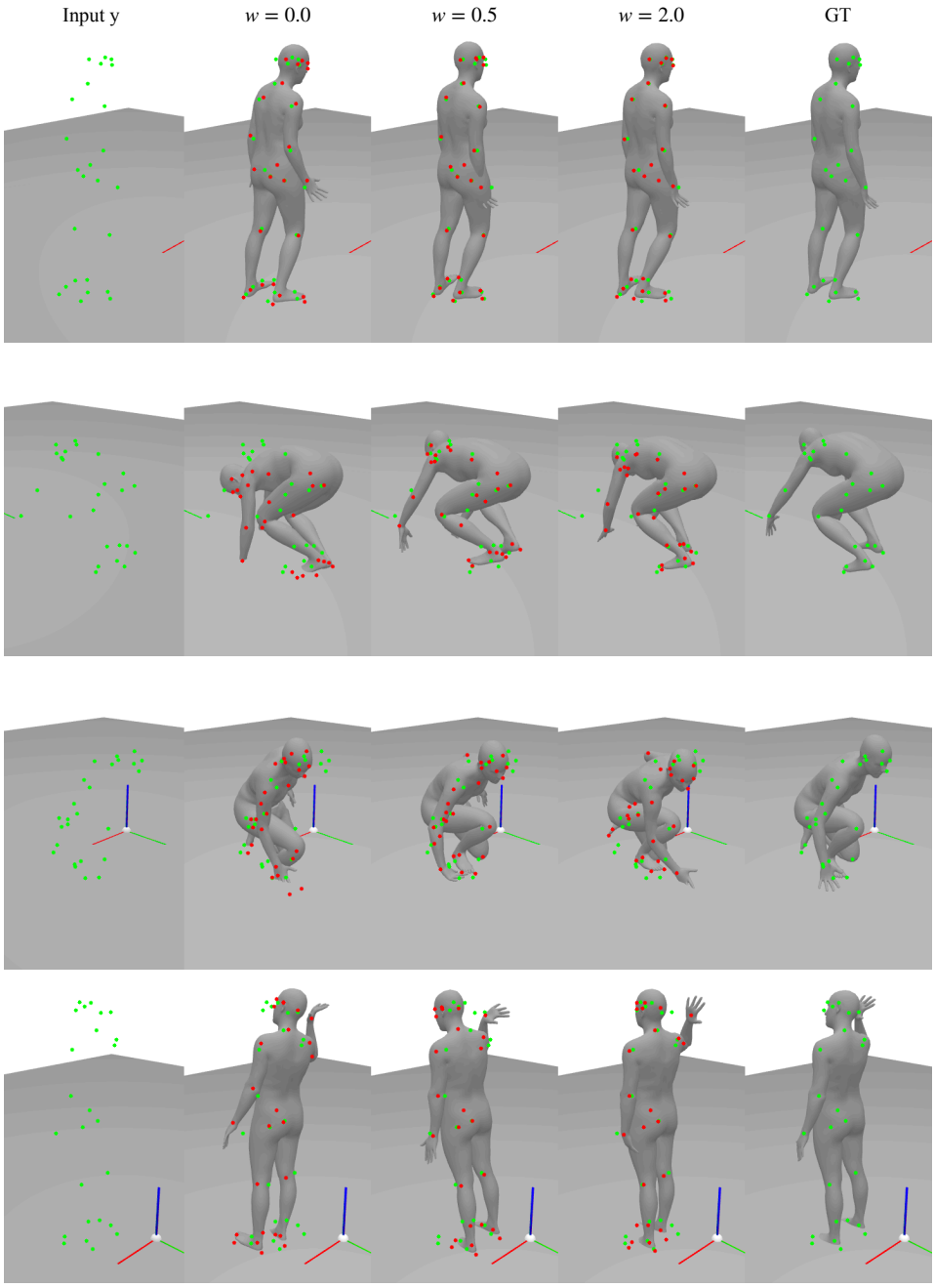

Figure 5: Input 2D keypoints (left), generated poses with three values of w(middle), and groundtruth(right).

IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(11):7327–7347, November 2022.

[3] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Zhangyang Wang, Howard Heaton, Jialin Liu, and Wotao Yin. Learning to optimize: A primer and a benchmark. *The Journal of Machine Learning Research*, 23(1):8562–8620, 2022.

[4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

[5] Erik Gärtner, Luke Metz, Mykhaylo Andriluka, C Daniel Freeman, and Cristian Sminchisescu. Transformer-based learned optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11970–11979, 2023.

[6] Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation, 2023.

[7] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[8] Eldad Haber, Uri M Ascher, and Doug Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse problems*, 16(5):1263, 2000.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.

[12] Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion models for black-box optimization, 2023.

[13] Tianci Liu, Tong Yang, Quan Zhang, and Qi Lei. Optimization for amortized inverse problems. 2023.

[14] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[15] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, October 2019.

[16] Luke Metz, C Daniel Freeman, James Harrison, Niru Maheswaranathan, and Jascha Sohl-Dickstein. Practical tradeoffs between memory, compute, and performance in learned optimizers. In *Conference on Lifelong Learning Agents (CoLLAs)*, 2022.

[17] Willie Neiswanger, Ke Alexander Wang, and Stefano Ermon. Bayesian algorithm execution: Estimating computable properties of black-box functions using mutual information. In *International Conference on Machine Learning*. PMLR, 2021.

[18] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[19] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[21] Paul Vicol, Luke Metz, and Jascha Sohl-Dickstein. Unbiased gradient estimation in unrolled computation graphs with persistent evolution strategies. In *International Conference on Machine Learning*, pages 10553–10563. PMLR, 2021.

[22] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[23] 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)*, pages 681–688, 2011.

[24] Nan Ye, Farbod Roosta-Khorasani, and Tiangang Cui. Optimization methods for inverse problems. *2017 MATRIX Annals*, pages 121–140, 2019.

[25] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey, 2023.

[26] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.