The Ingredients for Robotic Diffusion Transformers

Sudeep Dasari[1], Oier Mees[2], Sebastian Zhao[2], Mohan Kumar Srirama[1], Sergey Levine[2]

Abstract— **In recent years roboticists have achieved remarkable progress in solving increasingly general tasks on dexterous robotic hardware by leveraging high capacity Transformer network architectures and generative diffusion models. Unfortunately, combining these two orthogonal improvements has proven surprisingly difficult, since there is no clear and well-understood process for making important design choices. In this paper, we identify, study and improve key architectural design decisions for high-capacity diffusion transformer policies. The resulting models can efficiently solve diverse tasks on multiple robot embodiments, without the excruciating pain of per-setup hyper-parameter tuning. By combining the results of our investigation with our improved model components, we are able to present a novel architecture, named DiT-Block Policy, that significantly outperforms the state of the art in solving long-horizon (1500+ time-steps) dexterous tasks on a bi-manual ALOHA robot. In addition, we find that our policies show improved scaling performance when trained on 10 hours of highly multi-modal, language annotated ALOHA demonstration data. We hope this work will open the door for future robot learning techniques that leverage the efficiency of generative diffusion modeling with the scalability of large scale transformer architectures. Code, robot dataset, and videos are available at: https://dit-policy.github.io**

I. INTRODUCTION

Modern machine learning has achieved remarkable success by leveraging highly expressive deep neural networks to generate and model samples from extensive offline imitation datasets [1], [2], [3]. Inspired by these advances, the field of robotics is adopting similar techniques to develop general policies and controllers for manipulation [4], [5] and locomotion tasks [6], [7]. However, robotics tasks present multiple challenges that hinder the straightforward application of these methods. First, the policy must learn to process high-dimensional observation streams from multiple cameras, without overfitting to spurious correlations in the data. For example, the policy may learn to regress actions directly from proprioceptive signals or a specific camera view. Thus, during test time it would entirely ignore signals from other modalities (e.g., wrist cameras) that are critical for solving highly dexterous tasks with potential occlusions. This often results in catastrophic failure during deployment. Second, the robot must make extremely precise action predictions, due to the low error tolerance in object manipulation. This is especially important when solving long horizon tasks, where the robot may need to achieve multiple sub-goals in sequence before the trajectory ends. For example, a robot tasked with preparing a sushi dish would need to reach multiple "cutting" sub-goals, which each have millimeter level error thresholds, as showcased in Fig. 1. Finally, policy learning needs to

[1]Carnegie Mellon University. [2]University of California, Berkeley.
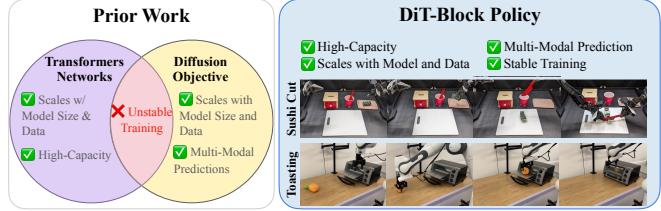
Fig. 1: **Overview:** We introduce Diffusion Transformer Block Policies (i.e., DiT-Block Policies), a novel architecture that combines the scalability of Transformer backbones with generative modeling, without the excruciating pain of per-setup hyper-parameter tuning.

contend with multi-modal action distributions – i.e., different ways of solving the same task. Simply learning the average action from this distribution will often result in an indecisive and error-prone behaviors. Handling action multi-modality becomes particularly important as the dataset size increases, since different experts will naturally demonstrate different behaviors. Failing to address these challenges will result in an unreliable and unsafe policy during deployment.

Recent advancements have begun to address these issues, by developing higher-capacity network architectures for dexterous task [8], and leveraging improved generative modeling frameworks like diffusion [9] for effective multi-modal action learning [10]. Combining these two orthogonal improvements could yield highly capable policies, but has proven surprisingly challenging so far. For example, the original diffusion policy paper [10] proposed a naïve cross-attention Transformer [11] implementation for the policy network that was (according to their own analysis) extremely difficult to train. As a result, most follow-up works [12] build upon their U-Net architecture [13], which is easier to tune but imposes strict requirements on the task setup (e.g., action signals must be sufficiently smooth [10]). As a result, high-capacity diffusion modeling remains inaccessible for a wide range of robotics applications.

This work's key insight is that unstable transformer diffusion policy training is not a fundamental problem, and can be largely resolved with a novel policy architecture. Our contributions are: **(1) Scalable Attention Blocks**: we propose a key improvement (inspired by Peebles et. al. [14]) to stabilize training by adding adaptive Layer Norm (adaLN) blocks to the diffusion transformer policy layers. This simple trick improves performance by 30%+ on long horizon, dexterous, real-world manipulation tasks containing over 1000 decisions! **(2) Efficient Observation Tokenization**: we compare several methods to tokenize multiple camera observations, such as Vision Transformers [15] and ResNet [16] encoders.

Again, we find that a relatively simple implementation (ResNet image tokenizer + Transformer policy) can provide a substantial ($40\%+$) performance boost over competing strategies. **(3) DiT-Block Policy:** We integrate the best performing components in a unified framework, coined DiT-Block Policy. Our model achieves State Of The Art (SOTA) performance on both a bi-manual, low-cost ALOHA [8] robot, and on a single-arm DROID Franka setup [17]. **(4) Open Source Models and Data:** We open-source all of our data, code and models for the community's benefit. This includes BiPlay, a new language annotated dataset containing 7023 clips of dexterous, bi-manual manipulation tasks

II. RELATED WORK

a) Encoding high dimensional observations: In order to perceive their environment, robots typically make use of multiple sensory observations. Therefore, how to best combine information from multiple sensors is a age-old question in robotics and computer vision [18], [19], [20], [21], [22]. For example, bi-manual robots like ALOHA [8] must combine information from global cameras that view the whole scene and wrist cameras that get a close-up view of the manipulation itself. The most straight-forward way to handle this problem is to learn a single shared network/encoder that operates across all the input modalities simultaneously [4], [23]. However, these systems often learn brittle features that overfit to specific inputs, e.g., proprioceptive data and global cameras, while ignoring others entirely. Possible solutions from prior work include using separate high-capacity image encoders for each visual stream [8], [10], injecting 3D aware spatial biases into the representation network [24], [25], and properly regularizing the features using observation dropout [26], [27]. Our findings reveal that a combination of these tricks provide a roughly 40% boost on long-horizon, bi-manual tasks, and that these seemingly small details are crucial for successful visuo-motor control.

b) Predicting multi-modal action distributions: Modeling multi-modal action distributions – i.e., scenarios where the robot could take multiple entirely different actions from the same observation/goal – is a well known challenge for BC methods [28]. This challenge often intensifies as the amount of expert data increases, since different demonstrations may showcase different solutions for the same task. Potential solutions include action space discretization [29], [30], [31], [32], [33], modifying π to predict higher capacity action distributions [23], [34], [35], implicitly modeling the action distribution [36], [37], [38], [39], [40], and using a generative modeling objective like diffusion [41], [42], [10], [43], [44], [45]. Diffusion in particular has shown state-of-the-art results in robotics [10]: it can learn complex 3D-aware policies [46], [47], and concurrent work even showed state-of-the-art manipulation results on bi-manual robotic arms [48]. However, the model architectures/hyperparameters are very sensitive and difficult to tune [10], [12]. This is a major barrier to scaling, since higher-capacity network architectures, such as Transformers [11], are crucial to fitting large and more diverse datasets. In contrast, our approach alleviates these issues by replacing the standard cross/joint attention conditioning blocks in a transformer decoder, with one better suited for diffusion [14].

III. PROBLEM SETTING

We consider the problem of acquiring a robotic controller via imitation learning [51], [52], [53], [54], [55], [56] that can perform challenging, dexterous manipulation behaviors when prompted to via language instructions. Specifically, the robot must learn a goal-conditioned policy $\pi_\theta (a_t \mid o_t, g)$ that predicts an action distribution $a_t \sim \pi(\cdot|o_t, g)$, given a new input observation (o_t) and a desired language goal (g), under environment dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, with $o_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$. The policy π is optimized via behavioral cloning [57], [28], [58] (BC) to match the optimal action distribution given a demonstration dataset $\mathcal{D} = \{\tau_1, \ldots, \tau_n\}$, where each trajectory $\tau_i = \{g, o_0, a_0, o_1, \ldots\}$ was collected from an expert agent (e.g., human tele-op data). During test time, actions are sampled from π and executed on the robot. We choose: o_t to be a set of image observations from the robot; a_t to be a chunk of H joint/Cartesian state actions, and g to be a text description of the task. This setting allows us maximum flexibility and generality for a wide range of robotics tasks, where precise states are difficult to infer and goals are free-form natural language instructions.

A. Training Objective

Our policy π is formulated as a conditional Denoising Diffusion Probabilistic Model [41] (DDPM), a type of generative model where the output is sampled using a denoising process [59]. Given the initial Gaussian noise x^k and a noise prediction network $\epsilon_\theta(x^k, k, o_t, g)$ the DDPM process produces $x^{k-1} = \alpha(x^k - \gamma\epsilon_\theta(x^k, k, o_t, g) + \mathcal{N}(0, \sigma^2 I))$, where k is the diffusion time index and α, γ, σ are parameters associated with the diffusion noise schedule [41]. When ϵ_θ is properly trained, this process will yield a sequence terminating in the optimal action: $x^k, x^{k-1}, \ldots, x^0 \simeq a_t$. Thus, our goal is to learn ϵ_θ via gradient descent [60], [61] using the following MSE objective: $\mathcal{L} = ||\epsilon^k - \epsilon_\theta(a_t + \epsilon^k, k, o_t, g)||_2^2$. Note that we use $k = 100$ diffusion steps during training, a cosine noise schedule [62], and a standard deterministic sampling process to reduce the number of samples needed (to $k = 10$) during test time [63].

IV. INTRODUCING THE DiT-BLOCK POLICY

Our method – DiT-Block Policy– is a Transformer neural network architecture designed specifically to be a highly performant conditional noise network (ϵ_θ from above) for robotic diffusion policies. The DiT-Block Policy architecture is visualized in Fig. 2. First, the text goal and robot proprioception inputs are encoded into observation vectors. Similarly, the time-step k is turned into an embedding vector using sinusoidal Fourier features [11], [64] and a small MLP network. Then, all these embedding vectors are combined with the input noise vector (x^k) using an encoder-decoder Transformer architecture to produce the denoising output ϵ^k. We now describe a few ingredients that are key to enable stable training and improved action prediction performance.
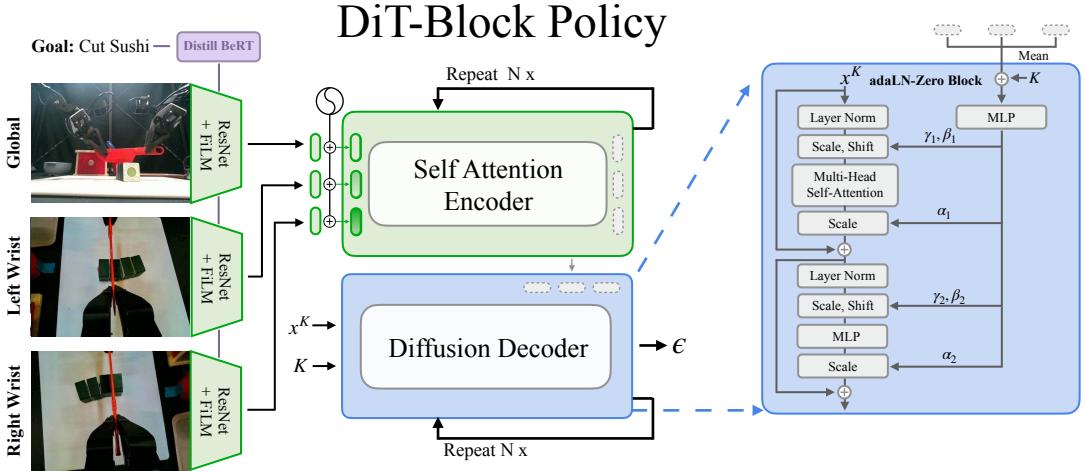
DiT-Block Policy

Fig. 2: **Policy Architecture:** Our DiT-Block Policy architecture enables scalable, goal-conditioned policy learning for various robotics tasks. Image observations are tokenized using separate ResNet-26 [16] encoders. The text goal is tokenized and encoded into an embedding vector using a pre-trained Distill BeRT model [49]. This vector is incorporated into the observations tokens using FiLM Layers [50]. The observation tokens are passed into a encoder-decoder transformer network (middle), which is responsible for predicting the noise epsilon (ϵ) used for diffusion. For stable training, the decoder block leverages a custom adaLN-Zero architecture (right), enabling the transformer to scalably optimize the diffusion objective.

a) Processing diverse multi-camera observations: Before passing through the transformer backbone, the input images, text goal, and joint angle observations need to be tokenized. The input images from each camera are processed *separately*, using Convolutional Neural Network (CNN) backbones [65]. While other vision transformers [15], [66] may skip this stage entirely, the intensive spatial reasoning and limited data in many robotics tasks can benefit from the spatial priors in higher-capacity CNNs. Thus, we used ResNet-26 [16] as the encoder. The text goals are incorporated into the vision encoder via FiLM layers [50]. This enables the text goals to influence the network's visual attention at all layers of the network. Finally, the proprioceptive inputs are regularized with a per-dimension observation dropout [26], [27], before tokenization. After the initial tokenization, learned positional encodings [11] are added to the input tokens, and processed together using the Block Attention transformer encoder implementation from Octo [4]. These results in a series of transformer joint embedding tokens $e^{(1)}, \ldots, e^{(L)}$, where L is number of layers.

b) Leveraging adaLN-Zero attention blocks for policy learning.: In parallel, a transformer decoder (with L layers) processes the current (noised) input (x^k), time-step (k), and encoder embeddings. We note that each decoder block i processes its corresponding embedding from the encoder $e^{(i)}$. Typically, this processing occurs via a standard cross attention mechanism, enabling the decoder to index into $e^{(i)}$ using its input tokens. Our key insight is, that this default attention implementation explains the poor training dynamics of prior diffusion policy transformer implementations [10], [12]. Thus, we propose replacing standard cross-attention blocks with an adaptive Layer-Norm (adaLN) mechanism that plays a key role in stabilizing diffusion transformers in image generation tasks [14]. These blocks work by injecting the conditioning vector into the Transformer's LayerNorm

blocks, by shifting and scaling the input vectors: $x = a(e^{(i)}, k) * x + b(e^{(i)}, k)$. We choose a and b to be simple dense layers that operate on the mean encoder embedding and the time vector: $a(e^{(i)}, t) = \texttt{tokenmean}(e^{(i)}) + t$. In addition, the output scales projection layers, before residual layer, are initialized to 0 (hence adaLN-Zero). This essentially initializes the noise network with identity skip connections, and thus further improves its learning dynamics [67].

V. DiT-Block Policies for Bi-Manual Tasks

Inspired by prior work on data scaling [68], [69], [17], [5], [29], [70], we seek to understand how DiT-Block Policies will behave as they are trained on increasingly diverse demonstrations data. However, the few (open-source) bi-manual datasets that do exist [8], [71] only consist of a handful of tasks, collected using the same controlled scenes/objects. As a result, they are not useful for testing generalization in our bi-manual setting. To address this shortcoming, we collected and annotated BiPlay, a more diverse bi-manual manipulation dataset with randomized objects and background settings as shown in Fig. 3. We collected BiPlay as a series of 3.5 minute long episodes. For each episode, we constructed a random scene with various objects, and solved a sequence of tasks within that scene. After collection, the episodes were broken into clips that were in turn annotated with appropriate language task descriptions. The final dataset contains 7023 clips spanning 10 Hrs of robot data collection.

A. Training Protocol

To train our models, we collected a fixed set of demonstrations (100+ demos) for each of our evaluation tasks (see Sec. VI-A). In addition, we compiled open sourced data from prior work (ALOHA [8] dataset and optimal policy rollouts from YaY [71]), and added it to the training mix for regularization. The full mix of data is presented in Table I. All DiT-Block Policies were trained on this data-mix for

Fig. 3: **Introducing BiPlay:** To create this dataset we constructed 326 scenes in an ALOHA play-pen that we used to collect 7023 unique interaction sequences, with diverse objects, goals, language annotations and tasks.

250K iterations, using the AdamW [61] optimizer and a cosine learning schedule [72]. Finally, instead of predicting a single action at each step, we trained DiT-Block Policy models to predict a chunk of $H = 100$ actions. This acted as regularization during training, and allowed us to employ temporal ensembling [8], to improve stability at runtime.

VI. EXPERIMENTAL SETUP

Our experiments are designed to understand DiT-Block Policy's limits and capabilities. First, we defined a series of manipulation tasks using two different robot morphologies in Sec. VI-A. Then, we trained the policies on separate mixes of task demonstration data, grouped by morphology.

A. Task Setups

Our first task set considers a bi-manual, low-cost ALOHA robot [8], which enables us to investigate challenging scenarios with highly dexterous, precise behaviors. We now describe these tasks and their success criteria in detail: **(1) Pick Place:** Given a text instruction (like $g =$ "pick up the corn and place it in the bowl") the robot must find the target object, grasp it, and then drop it into the target plate/bowl. There are always two objects and two possible targets in the scene, so the robot must properly ground its behavior

Dataset	Make-Up	Scenes	Tasks	Length
BiPlay	7k Play Clips	326	200+	9.7 Hrs
ALOHA [8]	855 Demos	15	16	2.9 Hrs
YaY [71]	4k Rollouts	3	3	15.4 Hrs
Pen Uncap	100 Demos	1	1	0.3 Hrs
Sushi Cut	256 Demos	1	1	2.7 Hrs
Pick Place	863 Demos	1	1	1.4 Hrs
Dough Cut	150 Demos	1	1	1.8 Hrs
Open Drawer	115 Demos	1	1	2.7 Hrs

TABLE I: **Training Mix:** We train DiT-Block Policy policies on: BiPlay, prior bi-manual manipulation datasets, and expert demonstration data collected for each task.
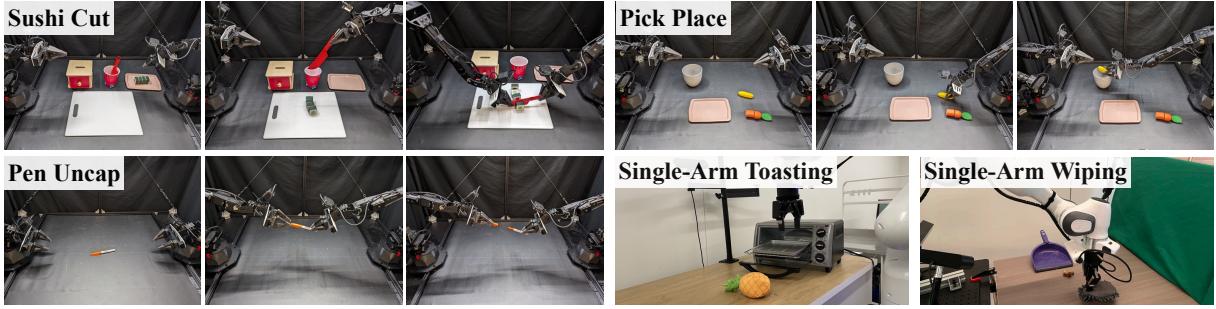
in the text instruction. A trial is marked successful if the object ends in the correct receptacle. **(2) Pen Uncap:** This task evaluates the robot's precision grasping capabilities and its ability to control both arms simultaneously. The robot must pick up the sharpie with one arm, bring it to the other arm, and then remove the cap from the pen. A trial is marked successful if it ends with the pen uncapped. **(3) Sushi Cut:** This task evaluated the robot's ability to chain precise, dexterous manipulation tasks over a long horizon. It is the most challenging task, since even a small error over a 2 min episode could derail the policy. The robot must place the sushi on the cutting board, pick up a knife from the cup, re-grasp it with the other hand, and then cut the sushi into four pieces. The task is marked successful if it ends with the sushi split into four, but partial credit is given for the fraction of successful cuts (e.g., one cut gets $1/3$).

Our next task set uses a single-arm Franka FR3 robot. While the dexterity is more limited, the Franka allows us to test generalization to an entirely new morphology and control space (Cartesian velocity). We consider the following tasks: **(1) Toasting:** In this long-horizon task, the robot must pick up the target object, place it in the toaster, and then shut the toaster. A trial is marked as successful if the toaster if the full sequence is completed, and is marked as half successful if the object is only placed in the toaster. **(2) Wiping:** The robot must localize the sponge, grasp it, and then push the debris into the dustpan. The trial is marked successful if all debris is wiped at the end of the run.

VII. RESULTS

This section evaluates DiT-Block Policy on our task suite in order to contextualize its performance and analyze the source of its improvements. First, we compare DiT-Block Policy against the strongest baselines from the field in Sec. VII-A, and find an average improvement of 20%. Next, the ablation studies (see Sec. VII-B) reveal that the diffusion head implementation is critical for stable training, and observation tokenizer architecture provides a significant performance boost. Finally, we show that these findings generalize to different robot hardware in Sec. VII-C, and provide a standardized sim evaluation (see Sec. VII-D).

A. Comparison to Prior SOTA Architectures

Our first experiments compare DiT-Block Policy against SOTA baselines from the field in order to contextualize its performance. These baselines include: **(1) Action Chunking Transformers [8] (ACT):** ACT is built with a standard encoder-decoder transformer architecture, concretely DeTR [73]). The encoder processes input observation tokens, which include camera observations (encoded via ResNet-18 [16]), goal conditioning vectors, and (optionally) a latent plan vector computed from ground truth actions during training (randomly sampled during inference). The network is optimized via BC, using a L1-regression loss on expert actions. We implemented this baseline using the recommended hyper-parameters, and omitted the latent plan vector based on advice from the authors. In many respects, this

Fig. 4: **Evaluation Tasks:** We evaluate DiT-Block Policies on a set of 3 Bi-Manual and 2 Single-Arm manipulation tasks.

	Pick Place		Pen Uncap		Sushi Cut		Average	
Using BiPlay?	Yes	No	Yes	No	Yes	No	Yes	No
DiT-Block Policy	**50%**	37.5%	**100%**	90%	**29%**	13%	**60%** $\pm$ 9%	51% $\pm$ 9%
ACT [8]	37.5%	25%	40%	70%	21%	17%	33% $\pm$ 8%	37% $\pm$ 8%
D.P. U-Net [10]	31.3%	18.8%	90%	70%	4%	0%	42% $\pm$ 9%	30% $\pm$ 9%
D.P. Transformer [10]	0%	0%	0%	0%	0%	0%	0% $\pm$ 0%	0% $\pm$ 0%

TABLE II: **Baseline Comparison:** We compare DiT-Block Policy against SOTA baselines from the field (ACT [8], Diffusion Policy w/ U-Net [10], and Diffusion Policy with Transformer [10]). Our method is able to outperform the baselines by 20%.

model is analogous to DiT-Block Policy, but with a standard attention block and no diffusion loss. **(2) Diffusion Policy w/ U-Net [10] (D.P. U-Net):** This is the original Diffusion Policy implementation from Chi et. al. [10]. The camera observations are first processed into representation maps (via separate ResNets [16]), and then the dimensionality is reduced into a vector using spatial softmax [74]. This observation vector is then fed into a conditional U-Net network [13] that functions as the noise network. The policy is trained using the DDPM diffusion training objective. **(3) Diffusion Policy w/ Transformer [10] (D.P. Transformer):** This is the same setup as described previously, but with the U-Net noise network replaced with a Transformer, which uses a standard causal cross attention block. While higher capacity, this model is notoriously hard to train [10], [12].

All three baselines were compared against DiT-Block Policy on our bi-manual evaluation tasks. Each method was trained twice, once with just the demonstration data and once with BiPlay, in order to understand their data scaling properties. Full results are presented in Table II. We find that DiT-Block Policy is able to outperform the strongest by roughly 20% when trained with BiPlay, and by 10% when trained on task data alone. This indicates that DiT-Block Policy delivers SOTA performance, while also scaling better than the baselines. In addition, our method is able to deliver solid performance on all three tasks. In contrast, each of the other baselines has a task where it falls flat – e.g., ACT struggles with pen uncap, and D.P. U-Net struggles with sushi cutting. Finally, note that the D.P. Transformer baseline is unable to solve *any* of our tasks, because unstable training caused noisy/unsafe action prediction. Thus, we conclude that DiT-Block Policy learns diffusion policy transformers more stably than the baseline does.

B. Ablation Studies

a) Ablating the attention mechanism: A key finding from the prior section is that DiT-Block Policy's transformer

implementation enables more stable training and policy inference. But is this inherent to the transformer architecture, or a factor of some other hyper-parameter? Thus, we conduct an apples-to-apples comparison in order to answer this question. We compare DiT-Block Policy against 3 ablations that use the same exact setup, but with a different attention block: **(1) Cross Attention:** The diffusion decoder uses a standard per-layer cross attention block [11] to condition on memory embeddings from the encoder stack (i.e., ACT [8] + diffusion). Concurrent work [48] demonstrated SOTA results with this architecture, though with a much larger dataset (26K episodes) and extensive tuning. **(2) In-Context:** The memory embeddings from the encoder are added to the decoder in context, and all further processing happens with standard causal self-attention. **(3) Non-Zero Initialization:** This is an adaLN block, but without zero-initializing the final layers.

We compare these ablations against a DiT-Block Policy on the pick place and uncapping tasks. Results are presented in Table III. We find that the cross attention and in-context attention blocks are far less stable during training. It is still possible to generate stable actions during evaluation, by significantly increasing the number of diffusion steps during inference. However, the performance is still significantly reduced v.s. our DiT-Block Policy, and the slow inference speed results in jerky trajectories when deployed on the

Method	DDIM Iters.	Pick Place	Pen Uncap
Ours	10	**50%** $\pm$ 12%	**100%** $\pm$ 0%
No Zero-Init.	10	38% $\pm$ 11%	80% $\pm$ 13%
Cross Attn. [48]	10	0% $\pm$ 0%	0% $\pm$ 0%
Cross Attn. [48]	100	38% $\pm$ 15%	70% $\pm$ 11%
In Context	100	0% $\pm$ 0%	0% $\pm$ 0%

TABLE III: **Attention Block Ablation:** Our proposed attention architecture significantly improves v.s. baselines.

Method	Parameters	Pick Place	Pen Uncap
Ours	$115M$	**50%** $\pm 12\%$	**100%** $\pm 0\%$
No ResNet	$85M$	$0\% \pm 0\%$	$0\% \pm 0\%$
No ResNet	$150M$	$13\% \pm 8\%$	$20\% \pm 13\%$

TABLE IV: **Encoder Ablation:** We ablate our choice of ResNet encoder tokenization, which effectively shifts more compute/parameters below the transformer layers.

robot. In contrast, we find that the zero initialization ablation is able to effectively train and predict actions with fewer inference steps. But it still under-performs the DiT-Block Policy by 16%. Altogether, we conclude that the DiT-Block Policy's architecture offers a critical boost for diffusion transformer policy performance, and that the initialization scheme provides an additional boost on top.

b) Ablating the image tokenization scheme: We evaluate our method's observation tokenizer by testing against ablations that move these parameters into the transformer encoder itself. Specifically, the ResNets are replaced with three small convolutional stem layers [4], [65] that produce an equivalent number of tokens (49 per image). Then, we train using these ablated observation tokens, and scale up the parameters to compensate. Results comparing these ablations against the full DiT-Block Policy are presented in Table IV. We find that DiT-Block Policy significantly outperforms the ablation with an even parameter count, and that even the significantly scaling up ablation is unable to compensate. This suggests that CNNs should still be considered as encoders for robotics tasks, particularly for low-data regimes.

C. Generalization to Other Robot Morphologies

The final experiments test if our findings still generalize to a new robot embodiment. Specifically, we test generalization to a single-arm Franka robot. While this setup is morphologically simpler, there are a few important differences that could prove challenging in practice. First, we evaluate the policies with a single external camera so they will need to gracefully handle occlusion during manipulation. Second, these robots
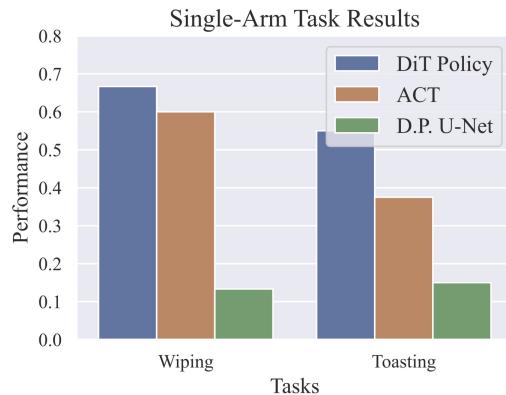

Fig. 5: **Single Arm Real World Tasks:** We evaluate our DiT-Block Policies on the Franka real world single arm tasks, and find that it outperforms the strongest baseline by over 20%.

Task	DiT-Block Policy	D.P. Transformer [10]
Lift	100%	100%
Can	98%	100%
Square	84%	100%
Tool Hang	72%	76%
Avg (Sim)	88.5%	94%
Avg (ALOHA)	60%	0%

TABLE V: **Sim Evaluation:** We compare DiT-Block Policy against the original D.P. Transformer implementation [10] on 4 tasks from the robomimic simulation eval suite [23].

use a velocity action space, which may prove more difficult to learn. We evaluate the two strongest baselines against DiT-Block Policy on the toasting and wiping tasks. Results are presented in Fig. 5. Note that DiT-Block Policy again provides SOTA performance on these tasks: it outperforms ACT by 20% on average and D.P. U-Net by 35%. This suggests that DiT-Block Policy can generalize to new robots and is not overly sensitive to the particular choice of action and observation space, unlike the Diffusion Policy U-Net [10].

D. Standardized Evaluation in Simulation

While real-hardware evaluations are the ultimate test, it is often still useful to compare methods on reproducible, simulated task settings. Thus, we evaluate our DiT-Block Policy against the reference Diffusion Policy Transformer (D.P. Transformer) baseline from Chi et. al. [10] on the robomimic simulated task suite [23]. The results are reported in Table. V. We find that DiT-Block Policy almost completely matches D.P. Transformer on the simulated tasks, despite doing almost no task specific tuning (unlike D.P. Transformer). In addition, our method heavily out-performs the baseline on the real world experiments, which should carry more weight given the sim-to-real evaluation gap [27].

VIII. CONCLUSION

This paper presents DiT-Block Policy, an improved transformer architecture that enables stable diffusion policy learning and efficient inference. Our experiments show that DiT-Block Policies provide SOTA performance across 5 tasks and 2 different robots, which have radically different observation spaces, action spaces, and morpologies. We find that DiT-Block Policy outperform the strongest baselines by 20%, and are able to scale better with diverse play data. Our ablation study reveals that the exact configuration of DiT-Block Policy's transformer block is responsible for this increase. Standard joint-attention mechanisms are simply not able to learn policies as stably as DiT-Block Policy can. In addition, an ablation of our observation tokenizer reveals that using separate ResNet CNNs for image encoding provides stronger performance than using transformers alone. Even scaling the transformers is not enough to make up for this difference. Finally, we open-source the BiPlay dataset used in our experiments. This is the first language annotated, bimanual dataset with diverse scenes, tasks, and objects.

REFERENCES

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," *arXiv preprint arXiv:2403.03206*, 2024.

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[4] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[5] Open X-Embodiment Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, A. Raffin, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Ichter, C. Lu, C. Xu, C. Finn, C. Xu, C. Chi, C. Huang, C. Chan, C. Pan, C. Fu, C. Devin, D. Driess, D. Pathak, D. Shah, D. Büchler, D. Kalashnikov, D. Sadigh, E. Johns, F. Ceola, F. Xia, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Schiavi, H. Su, H.-S. Fang, H. Shi, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Kim, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Wu, J. Luo, J. Gu, J. Tan, J. Oh, J. Malik, J. Tompson, J. Yang, J. J. Lim, J. Silvério, J. Han, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Zhang, K. Majd, K. Rana, K. Srinivasan, L. Y. Chen, L. Pinto, L. Tan, L. Ott, L. Lee, M. Tomizuka, M. Du, M. Ahn, M. Zhang, M. Ding, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, P. R. Sanketi, P. Wohlhart, P. Xu, P. Sermanet, P. Sundaresan, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martín-Martín, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Moore, S. Bahl, S. Dass, S. Song, S. Xu, S. Haldar, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Dasari, S. Belkhale, T. Osa, T. Harada, T. Matsushima, T. Xiao, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Wang, X. Zhu, X. Li, Y. Lu, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho, Y. Lee, Y. Cui, Y. hua Wu, Y. Tang, Y. Zhu, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Xu, and Z. J. Cui, "Open X-Embodiment: Robotic learning datasets and RT-X models," 2023.

[6] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.

[7] R. Doshi, H. Walke, O. Mees, S. Dasari, and S. Levine, "Scaling cross-embodied learning: One policy for manipulation, navigation, locomotion and aviation," in *Conference on Robot Learning*, 2024.

[8] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[9] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[10] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," *arXiv preprint arXiv:2405.07503*, 2024.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[14] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4195–4205.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[17] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn, "Droid: A large-scale in-the-wild robot manipulation dataset," *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[18] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in neural information processing systems*, vol. 27, 2014.

[19] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Advances in neural information processing systems*, vol. 25, 2012.

[20] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*. Springer, 2014, pp. 297–312.

[21] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, 2016.

[22] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust rgb-d object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 681–687.

[23] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *arXiv preprint arXiv:2108.03298*, 2021.

[24] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *arXiv preprint arXiv:2110.14217*, 2021.

[25] J. Kerr, L. Fu, H. Huang, Y. Avigal, M. Tancik, J. Ichnowski, A. Kanazawa, and K. Goldberg, "Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects," in *6th annual conference on robot learning*, 2022.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[27] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta, "An unbiased look at datasets for visuo-motor pre-training," in *Conference on Robot Learning*. PMLR, 2023.

[28] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[29] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet, "Learning latent plans from play," *Conference on Robot Learning (CoRL)*, 2019.

[30] S. Dasari and A. Gupta, "Transformers for one-shot visual imitation," in *Conference on Robot Learning*. PMLR, 2021, pp. 2071–2084.

[31] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.

[32] O. Mees, L. Hermann, and W. Burgard, "What matters in language conditioned robotic imitation learning over unstructured data," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 11 205–11 212, 2022.

[33] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, "Latent plans for task agnostic offline reinforcement learning," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, Auckland, New Zealand, 2022.

[34] N. M. Shafiullah, Z. Cui, A. A. Altanzaya, and L. Pinto, "Behavior transformers: Cloning k modes with one stone," *Advances in neural information processing systems*, vol. 35, pp. 22 955–22 968, 2022.

[35] R. Rahmatizadeh, P. Abolghasemi, A. Behal, and L. Bölöni, "From virtual demonstration to real-world manipulation using lstm and mdn," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[36] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*. PMLR, 2022, pp. 158–168.

[37] Y. Song and D. P. Kingma, "How to train your energy-based models," *arXiv preprint arXiv:2101.03288*, 2021.

[38] B. Wang, G. Wu, T. Pang, Y. Zhang, and Y. Yin, "Diffail: Diffusion adversarial imitation learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 14, 2024, pp. 15 447–15 455.

[39] C.-M. Lai, H.-C. Wang, P.-C. Hsieh, Y.-C. F. Wang, M.-H. Chen, and S.-H. Sun, "Diffusion-reward adversarial imitation learning," *arXiv e-prints*, pp. arXiv–2405, 2024.

[40] S.-F. Chen, H.-C. Wang, M.-H. Hsu, C.-M. Lai, and S.-H. Sun, "Diffusion model-augmented behavioral cloning," in *Forty-first International Conference on Machine Learning*.

[41] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[42] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[43] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, *et al.*, "Imitating human behaviour with diffusion models," in *The Eleventh International Conference on Learning Representations*.

[44] V. Saxena, Y. Koga, and D. Xu, "Constrained-context conditional diffusion models for imitation learning," *arXiv preprint arXiv:2311.01419*, 2023.

[45] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-conditioned imitation learning using score-based diffusion policies," *arXiv preprint arXiv:2304.02532*, 2023.

[46] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[47] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *ICRA 2024 Workshop on 3D Visual Representations for Robot Manipulation*, 2024.

[48] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," in *Proceedings of the 7th Conference on Robot Learning (CoRL)*, Munich, Germany, 2024.

[49] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[50] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[51] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[52] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics*, vol. 59, no. BOOK_CHAP, 2008.

[53] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[54] B. Kang, Z. Jie, and J. Feng, "Policy optimization with demonstrations," in *ICML*. PMLR, 2018.

[55] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, *et al.*, "Deep q-learning from demonstrations," in *AAAI*, 2018.

[56] L. Weihs, U. Jain, I.-J. Liu, J. Salvador, S. Lazebnik, A. Kembhavi, and A. Schwing, "Bridging the imitation gap by adaptive insubordination," *NeurIPS*, 2021.

[57] S. Ross and D. Bagnell, "Efficient reductions for imitation learning," in *AISTATS*, 2010.

[58] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[59] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer, 2011, pp. 681–688.

[60] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[62] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International conference on machine learning*. PMLR, 2021, pp. 8162–8171.

[63] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020.

[64] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in neural information processing systems*, vol. 33, pp. 7537–7547, 2020.

[65] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better," *Advances in neural information processing systems*, vol. 34, pp. 30 392–30 400, 2021.

[66] A. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your vit? data, augmentation, and regularization in vision transformers," *arXiv preprint arXiv:2106.10270*, 2021.

[67] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[68] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du, *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[69] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn, "Robonet: Large-scale multi-robot learning," in *Conference on Robot Learning*. PMLR, 2020, pp. 885–897.

[70] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7327–7334, 2022.

[71] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn, "Yell at your robot: Improving on-the-fly from language corrections," *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[72] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2016.

[73] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[74] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.