ORAL: Prompting Your Large-Scale LoRAs via Conditional Recurrent Diffusion

Rana Muhammad Shahroz Khan[1], Dongwen Tang[2], Pingzhi Li[1], Kai Wang[2], Tianlong Chen[1]

[1]University of North Carolina at Chapel Hill, [2]National University of Singapore

Abstract

Parameter generation has emerged as a novel paradigm for neural network development, offering an alternative to traditional neural network training by synthesizing high-quality model weights directly. In the context of Low-Rank Adaptation (LoRA) for evolving (*i.e.*, constantly updated) large language models (LLMs), this approach promises efficient adaptation without costly retraining. However, existing methods face critical limitations in simultaneously achieving scalability and controllability. In this paper, we introduce ORAL, a novel **conditional recurrent diffusion** framework that addresses these challenges. ORAL incorporates a novel conditioning mechanism that integrates model architecture and textual task specifications, enabling the generation of task-specific LoRA parameters that can seamlessly transfer across evolving foundation models. Our approach successfully scales to billions-of-parameter LLMs and maintains controllability. Through extensive experiments across seven language tasks, four vision tasks, and three multimodal tasks using five pre-trained LLMs, we demonstrate that ORAL generates high-quality LoRA parameters that achieve comparable or superior performance to vanilla trained counterparts.

1 Introduction

Recent advancements in generative AI have been fueled by the vast amount of valuable data throughout the Internet, demonstrating profound transformations in the creation of texts, images, and videos (Liu et al., 2024; Achiam et al., 2023; Zhao et al., 2024; Zhang et al., 2024; Lin et al., 2023; Liu et al., 2023; Brooks et al., 2024). Inspired by these breakthroughs, researchers have begun to investigate whether the vast number of online pre-trained model checkpoints can serve as an untapped resource for AI development. Building on weight-space learning (Eilertsen et al., 2020; Schürholt et al., 2021; Zhao et al., 2023; Horwitz et al., 2024), which treats neural network parameters as a distinct modality, parameter generation has emerged as a novel paradigm.

Table 1: Comparison of LLM parameter generation methods across three dimensions: *Scalability* (ability to generate parameters at the scale of hundreds of millions); *Controllability* (support for task-specific conditional generation); and *Portability* (capacity to adapt to evolving foundation models without retraining).

Method	Scalable	Controllable	Portable
P-Diff (Wang et al., 2024)	✗	✗	✗
Cond P-Diff (Jin et al., 2024)	✗	✓	✗
RPG (Wang et al., 2025)	✓	✗	✗
ORAL (Ours)	✓	✓	✓

Pioneering works in this direction, such as P-Diff (Wang et al., 2024), have demonstrated the potential of diffusion models to generate neural network parameters that match or even surpass the performance of conventionally trained networks. P-Diff primarily focuses on unconditional generation, treating parameter synthesis as a distribution modeling problem without explicit control mechanisms. While this approach has shown promising results for small-scale architectures, it lacks the ability to guide the generation process toward specific tasks or model architectures, limiting its practical applications in rapidly evolving LLM ecosystems.

A subsequent work, Cond P-Diff (Jin et al., 2024), introduced conditionality to parameter generation, enabling task-specific adaptation by incorporating textual guidance. This significant advancement allows the synthesis of Low-Rank Adaptation (LoRA) parameters tailored for particular downstream tasks. However, Cond P-Diff faces critical limitations in: (1) scaling, typically constrained to generating around one million parameters, and (2) flexibility to adapt to weight changes in foundation models, requiring costly retraining

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 1357–1370
November 4-9, 2025

whenever the model evolves.

The recent RPG (Wang et al., 2025) framework alleviates the scalability challenge through a recurrent diffusion architecture. Despite this breakthrough in scale, RPG focuses exclusively on unconditional generation, offering no mechanisms to guide the generation process toward specific tasks or adaptation requirements. This is a significant limitation for practical deployment scenarios where task specificity is crucial.

Meanwhile, the landscape of LLMs undergoes frequent updates to their parameters (Khan et al., 2024), necessitating parameter generation methods that can efficiently adapt to these changes without the need for complete retraining. As summarized in Table 1, existing methods have only partially addressed the challenges in LLM parameter generation. While Cond P-Diff enables controllability and RPG achieves scalability, no current approach successfully combines all three crucial properties: *scalability*, *controllability*, and *portability* across evolving foundation model. These challenges lead us to our center research question:

RQ: How can we design a parameter generation framework that flexibly adapts to continuously evolving foundation models while maintaining efficient scaling properties for practical deployment?

We introduce ORAL, a novel framework that enables evolvable and scalable LoRA parameter generation. Our approach, building upon the recurrent diffusion backbone introduced in RPG (Wang et al., 2025), proposes a novel conditioning mechanism that incorporates both model architectural and textual specifications of adaptation characteristics as inputs to the diffusion process. Specifically, Our contributions are:

- We introduce a novel conditioning mechanism that integrates both model architectural and textual task specifications, enabling flexible generation of LoRA parameters for specific downstream tasks.
- We develop a novel conditional parameter generation pipeline that facilitates seamless transfer of generated LoRA parameters to evolving foundation models without requiring resource-intensive retraining, ensuring compatibility with rapidly advancing foundation models.
- Through comprehensive experiments across seven language tasks, four vision tasks, three multimodal tasks, and five pre-trained LLMs, we demonstrate the efficacy of our approach. The results validate that ORAL achieves remarkable scaling efficiency, effectively handling models with up to 7B parameters while maintaining or exceeding the performance of traditional fine-tuning approaches.

2 Related Works

Diffusion Models. Diffusion models have transformed generative AI, successfully creating high-quality images and other complex data. These methods (Dhariwal and Nichol, 2021b; Hertz et al., 2022; Li et al., 2022, 2023) build on non-equilibrium thermodynamics principles (Jarzynski, 1997; Sohl-Dickstein et al., 2015; Peebles and Xie, 2023) and have followed a development path similar to GANs (Brock et al., 2018; Isola et al., 2017a; Zhu et al., 2017; Goodfellow et al., 2014), VAEs (Kingma et al., 2013; Razavi et al., 2019), and flow-based models (Dinh et al., 2014; Rezende and Mohamed, 2015). They've evolved from simple unconditional generation to sophisticated systems that respond to text, images, and other structured inputs. Research on diffusion models spans four main areas: The first focuses on improving visual quality—exemplified by DALL-E 2 (Ramesh et al., 2022), Imagen (Saharia et al., 2022), and Stable Diffusion (Rombach et al., 2022; Podell et al., 2023)—significantly enhancing the realism of generated outputs. The second addresses their initially slow sampling process through methods like DDIM (Song et al., 2020), Analytic-DPM (Bao et al., 2022), and DPM-Solver (Lu et al., 2022), which have dramatically accelerated generation speeds. The third area involves theoretical work reexamining diffusion through continuous mathematical frameworks (Feng et al., 2023; Song and Ermon, 2019; Salimans and Ho, 2022; Shi et al., 2025), particularly score-based generative modeling (Feng et al., 2023), deepening our understanding of these techniques. The fourth focuses on applying diffusion across varied domains like Audio (Kong et al., 2020), 3D Point Generation (Luo and Hu, 2021) and Anomaly Detection (Wolleb et al., 2022), demonstrating their versatility as a generative approach.

Conditional Generation. The area of conditional generation has experienced incredible progress over the last decade, evolving through several distinct methodological stages. Initial efforts focused on conditional GANs (Mirza and Osindero, 2014;

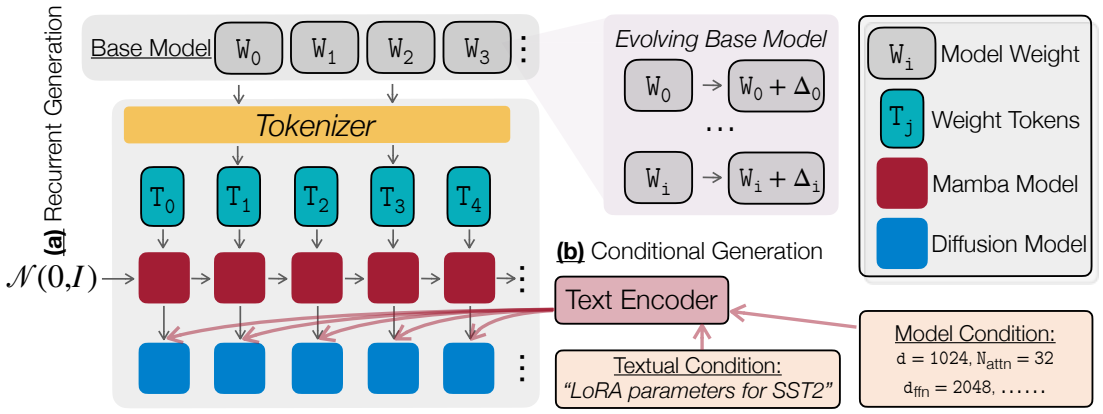

Figure 1: **Overview of our ORAL framework**. (a) Recurrent Generation: The foundation model weights are processed through a tokenizer to create weight tokens, which are then fed into a recurrent architecture consisting of both Mamba and Diffusion models to generate parameters from noise $\mathcal{N}(0, I)$. (b) Conditional Generation: Our approach supports **evolving** foundation model by adapting parameters $\mathtt{W}_i$ to updated foundation models $\mathtt{W}_i + \Delta_i$ without retraining, which is enabled by our novel conditioning mechanism that incorporates model architecture and textual task specifications.

Isola et al., 2017b; Zhu et al., 2017; Ho et al., 2020), which pioneered the integration of specific control signals into the generative process. Although these methods showcased the potential for guided synthesis, they often faced challenges related to training instability and limited output diversity. The advent of conditional VAEs (Sohn et al., 2015; Yan et al., 2016) initiated a notable transition toward probabilistic models. These models provided a more principled approach to managing uncertainty and enhancing sample diversity, although this sometimes impacted the quality of the generated outputs. They established critical theoretical frameworks for modeling the interactions between conditioning information and generated results.

Parameter Generation. The domain of parameter generation has significantly evolved, addressing the challenge of learning distributions over neural network weights. Early methods in parameter learning included stochastic neural networks (Sompolinsky et al., 1988; Bottou, 1991; Graves, 2011) and Bayesian neural networks (Neal, 1995; Kingma et al., 2013; Gal and Ghahramani, 2015), aimed at modeling probability distributions over weights to improve generalization and uncertainty estimation. While theoretically appealing, these faced scalability issues with modern architectures. HyperNetworks (Ha et al., 2016) advanced the field by generating parameters for other networks. This idea expanded in SMASH (Brock et al., 2017), enhancing architecture support through memory read-write mechanisms. More detailed recent works are discussed in Appendix B.

3 Method

3.1 Overview

In this section, we introduce ORAL (**LoRA** via conditional), our large-scale conditional LoRA generation framework. Building on the idea of recurrent diffusion from RPG (Wang et al., 2025), our approach fuses two distinct condition sources: (1) base-model encoding that identify which pretrained foundation model we are adapting, and (2) textual instructions that specify the target task or style. These two conditions feed directly into a diffusion-based architecture, augmented by a recurrent module. By decomposing the LoRA matrices into tokens, we enable scalable parameter generation for hundreds of millions of weights, while retaining explicit conditional control. Detailed preliminaries with primers on diffusion models, conditional generation and LoRA adapters have been provided in Appendix C.

3.2 Two-Part Conditional Modules

We aim to synthesize LoRA updates Θ suitable for adapting a foundation model W_0 to a specific downstream task. Formally, a LoRA update seeks to fine-tune via a low-rank update $\Delta W = BA$, with $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$. However, in practice, a large transformer typically has multiple trainable matrices across different layers; we use $\Theta = \{\Delta W^{(l)}\}$ to denote all LoRA updates, where each $\Delta W^{(l)}$ factors into $B^{(l)}A^{(l)}$. Merging these updates with W_0 yields the adapted weights for inference. Our approach aims to generate these

updates ΔW directly via a diffusion model conditioned on: (1) the base model and (2) a textual description of the task.

Base-Model Encoding. We rely on an encoder f_{base} that maps the entire base model weights W_0 to a compact embedding $c_{model} = f_{base}(W_0)$. We flatten certain structural metadata from W_0 (dimension, layer count, special tokens), and convert this into a string. Then we pass this string through an encoder which will generate an embedding vector. This embedding vector ensures that the generative process "knows" the architecture and dimension of W_0, so that the LoRA updates we synthesize are structurally compatible with it.

Textual Prompt Encoding. We simultaneously feed in a textual description $\mathcal{T}$ of the downstream task–e.g. "LoRA adapter for sentiment classification", or "LoRA Adapter for SST-2 task" alongside some fewshot examples from the dataset itself. An encoder f_{text} converts $\mathcal{T}$ into a condition vector $c_{text} = f_{text}(\mathcal{T})$. This can be realized by a pretrained text encoder like CLIP (Radford et al., 2021) or T5 (Raffel et al., 2019), depending on the application. This textual embedding imposes semantic constraints on the desired LoRA adapter.

Finally, we combine these two embeddings through concatenation into a global condition c:

$$c = [c_{model}\ ;\ c_{text}] \tag{1}$$

Throughout training and inference, c is provided to the generative modules so that both model encodings and textual instructions guide the LoRA generation.

3.3 LoRA Tokenization and Recurrent Prototypes

Tokenization. To handle LoRA updates at large scale, we adopt a tokenization approach similar to RPG (Wang et al., 2025). In particular, we break each $\Delta W^{(l)} \in \mathbb{R}^{d_l \times d_l}$ into fixed-size tokens:

❶ **Flatten and Group**: We flatten each layer's LoRA update $\Delta W^{(l)}$ and then apply normalization per layer to reduce any distributional shifts.

❷ **Splitting into Tokens**: We slice flattened vector at a predetermined token size k. Large layers simply produce more tokens; small layers produce fewer, and to make them of the same lengths, we pad them accordingly.

❸ **Positional Annotations**: Each token u_j is annotated with a layer index l and an intra-layer offset to preserve ordering. We represent these via simple sinusoidal embeddings p_j concatenated with the tokens.

After repeating the above process for all layers, we obtain a sequence $\{u_1, \ldots, u_T\}$. Each LoRA checkpoint Θ_i thus yields a "token list" plus conditions c_i.

Recurrent Prototypes. Following the insights from RPG (Wang et al., 2025), we incorporate a recurrent module f_ϕ to handle the token sequence. Given the tokens u_j, the module produces "prototype" vectors p_j, that summarize local correlation. Formally,

$$p_j, h_j = f_\phi(u_j, h_{j-1}) \tag{2}$$

where h_{j-1} is the recurrent hidden state from token $j-1$. We use a small memory-efficient module based on Mamba (Gu and Dao, 2023). Once we reach the final token, the set of prototypes $\{p_1, \ldots, p_T\}$ collectively encodes the entire LoRA adapter Θ in a compact representation. This ensures that the network can scale to large LoRA sets, without needing to flatten them into one huge input.

3.4 Conditional Diffusion

We adopt diffusion-based generative model, specifically 1D convolution network, to denoise each token, conditioning on both the recurrent prototypes p_j and global conditions c. Let $u_{j,0} = u_j$ be the original clean token. The forward process adds noise across time steps T:

$$u_{j,t} = \alpha_t u_{j,0} + \sigma_t \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{3}$$

Yielding highly corrupted tokens at large t. The reverse process is learned by a denoising network ϵ_θ, which predicts the noise given the current state, the prototype p_j, and the condition c. We train this network using the usual noise prediction loss:

$$\mathcal{L}_{diff}(\theta, \phi) = \sum_{t=1}^{T} \mathbb{E}\left[\|\epsilon - \epsilon_\theta(u_{j,t}, p_j, c, t\|^2\right] \tag{4}$$

This allows us to train the diffusion model, as well as pass the gradient through the recurrent module, without having to explicitly train it.

Table 2: Experimental results on different Image Generation and Multi-modal Tasks.

(a) FID scores (lower is better) for Stable-Diffusion 2.1 adapted to four target styles. "Stable-Diffusion 2.1" uses the original model directly, "Original LoRA" denotes standard fine-tuning, and "Ours". Best results are represented by **bold**.

	Pokemon FID(↓)	PixelArt FID(↓)	Cartoon FID(↓)	Retro FID(↓)
Stable-Diffusion 2.1	89.45	104.56	110.22	145.23
Original LoRA	24.40	26.89	**25.61**	42.39
ORAL (Ours)	**23.95**	**26.32**	25.77	**41.50**

(b) Performance comparison on Flickr30K, NoCaps, and DocVQA datasets. Retrieval@1 is reported for Flickr30K and NoCaps, and accuracy is reported for DocVQA. Best results are in **bold**.

	Flickr30K Retrieval@1(↑)	NoCaps Retrieval@1(↑)	DocVQA Accuracy(↑)
Qwen-7B-VL	85.78	121.42	65.11
Original LoRA	96.65	132.76	**84.53**
ORAL (Ours)	**96.72**	**134.81**	84.49

Table 3: Performance comparison on seven NLP datasets (BoolQ, SST-2, MRPC, RTE, Winogrande, WNLI, GSM8K) using accuracy as the evaluation metric. We report results for the Mistral-7B base model, Original LoRA baseline, and our conditional recurrent diffusion method.

	BoolQ Accuracy(↑)	SST-2 Accuracy(↑)	MRPC Accuracy(↑)	RTE Accuracy(↑)	Winogrande Accuracy(↑)	WNLI Accuracy(↑)	GSM8K Accuracy(↑)
Mistral-7B	83.58	66.86	65.20	67.51	74.11	57.76	6.37
Original LoRA	91.01	95.99	89.46	87.73	85.95	83.11	32.04
ORAL (Ours)	90.67 ↓ 0.25	96.01 ↑ 0.02	89.23 ↓ 0.23	86.34 ↓ 1.39	86.10 ↑ 0.15	83.11 ↑ 0.00	34.67 ↑ 0.63

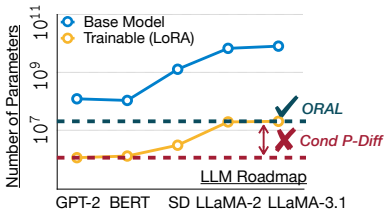

Figure 2: Comparison of parameter generation capacity between our proposed method, ORAL, and the baseline Cond P-Diff. ORAL effectively generates LoRA adapters at a significantly larger scale (e.g., 7B Models), surpassing the capacity of Cond P-Diff, which fails to operate efficiently at higher parameter scales. This demonstrates ORAL's ability to handle large-scale parameter synthesis, crucial for adapting modern large language models.

4 Experiments

In this section, we will evaluate our conditional large-scale LoRA generation method to generate parameters across multiple tasks, models, and modalities. We first report our experimental setup that is constant across all experiments, like training details, and then experiment-dependent settings and results are contained within their own sections.

4.1 Experimental Setup

Training Details. We train our conditional recurrent diffusion model for $100,000$ iterations on eight A6000 GPUs. Each run processes tokens of fixed size 8196, leveraging a Mamba (Gu and Dao, 2023) block as the recurrent backbone alongside a 1D convolution diffusion network. We encode the base model via a text description passed through a BERT-Base-Uncased (Devlin et al., 2018) encoder, while task prompts are handled by the CLIP (Radford et al., 2021). Within each experiment, we devote one training pipeline per task, ensuring the model fully learns the distribution of LoRA parameters tailored to that specific objective. For all the experiments we set our LoRA rank to be 8, however, an ablation study on various ranks was done.

Inference Details and Comparisons. During inference, our approach takes as input both the textual prompt and the base-model description. We initialize each token with Gaussian noise and then run through our learned 1D convolutional diffusion network, conditioned on the recurrent prototypes from Mamba and the global conditions. Thanks to the tokenization scheme, inference scales to hundreds of millions of LoRA parameters without memory overflow. Notably, no existing method can generate large-scale LoRA parameters conditionally in this manner, so we do not compare against external baselines for a like-for-like evaluation. Instead, we focus on our method's ability to effectively synthesize LoRA adapters across various tasks and model sizes and compare them to the original results. For evaluation, we synthesize LoRA parameters three times (via different random seeds) for each target condition. We merge these generated LoRAs into the base model and measure downstream performance, reporting the average (and sometimes best/min) accuracy.

4.2 Performance on Image Generation Task

Setup and Datasets. In this experiment, we test our method's ability to adapt Stable-Diffusion 2.1 (Rombach et al., 2022), to four distinct styles, each realized as a LoRA Adapter: Pokemon, PixelArt, Cartoon, and Retro. Concretely, we treat

Stable-Diffusion as the base model and synthesize separate LoRA parameters to specialize it for each of the four styles. We evaluate image generation quality using the FID score, sampling images from the adapted model in each style and comparing their statistics to a style-specific reference set. We compare the results from our generated LoRA with the original LoRA as well as the zero-shot performance of the base model to motivate the need for fine-tuning.

Results. Table 2a reports the FID scores for all four tasks and we can summarize our results as follows: ❶**Comparison with original LoRA:** Our method closely matches or slightly surpasses standard LoRA fine-tuning in terms of style fidelity, as measured by the FID. In particular, we observe improvements in FID for Pokemon (0.45 decrease) and PixelArt (0.57 decrease), while achieving nearly identical results for Cartoon (0.16 increase) and a clear improvement for Retro (0.89 decrease). These results demonstrate that our conditional recurrent diffusion method successfully captures and reproduces most of the style-specific characteristics traditionally obtained through gradient-based fine-tuning. ❷**Comparison with Base Model:** Compared to the original Stable-Diffusion 2.1 (without LoRA), our generated LoRA adapters significantly enhance image-generation quality across all evaluated styles. Specifically, we achieve substantial FID reductions for the evaluated styles—Pokemon ($89.45 \rightarrow 23.95$), PixelArt ($104.56 \rightarrow 26.32$), Cartoon ($110.22 \rightarrow 25.77$), and Retro ($145.23 \rightarrow 41.50$). These improvements confirm our model's effectiveness at synthesizing style-specific adapters, greatly enhancing style consistency.

4.3 Performance on Multi-modal Tasks

Setup and Datasets. The goal of this experiment is to evaluate our framework for LoRA parameter generation on multi-modal tasks. The experimental setup consists of adapting the base model Qwen-7B-VL (Bai et al., 2023), using LoRA adapters specialized for three distinct multimodal tasks: (1) Flickr30K (Plummer et al., 2015): a captioning task, (2) NoCaps (Agrawal et al., 2019): another captioning task, and (3) DocVQA (Mathew et al., 2020): a document based Visual Questioning Answering Task. We compare our method against the zero-shot performance of the base model, as well as the original LoRA adapter to showcase the generation. For Flickr30K and NoCaps, we utilize the Image-to-Text Retrieval at 1 metric, while for the DocVQA task we use Accuracy.

Results. The results for all three multi-modal tasks are summarized in Table 2b. Our method closely matches or slightly surpasses the Original LoRA baseline for image-to-text retrieval tasks, as measured by Retrieval@1. Specifically, we observe improvements in Retrieval@1 for Flickr30K (0.07 increase) and NoCaps (2.05 increase). However, for the DocVQA dataset, our accuracy slightly trails Original LoRA by 0.04%. These results confirm that our conditional recurrent diffusion method effectively captures task-specific characteristics, demonstrating comparable or superior performance to gradient-based LoRA fine-tuning.

4.4 Performance on NLP Tasks

Setup and Datasets. We would also like to evaluate the quality of our LoRA adapters for NLP Tasks. We test our framework's ability to adapt the base model, Mistral-7B (Jiang et al., 2023), using LoRA adapters specifically tailored for seven diverse NLP datasets where five of them: BoolQ (Boolean Question-Answering Task), SST-2 (Sentiment Analysis Task), MRPC (Paraphrase Detection Task), RTE (Recognizing Textual Entailment Task), WNLI (Natural Language Inference Task) are from SuperGLUE Benchmark (Wang et al., 2019) as well as 2 other tasks: Winogrande Benchmark (Sakaguchi et al., 2019) (Commonsense Reasoning Task), and GSM8K (Cobbe et al., 2021) (Grade-school Math Problems). For each downstream task, we calculate the accuracy and report it for comparison.

Results. Our third experiment evaluates our method's effectiveness in generating LoRA parameters for natural language processing tasks using the Mistral-7B model. As shown in Table 3, our generated LoRA parameters demonstrate strong performance across seven diverse NLP tasks. When compared to the Original LoRA baseline, our method achieves competitive or superior results, with slight improvements on SST-2 ($+0.02\%$), Winogrande ($+0.15$), and GSM8K ($+0.63$), identical performance on WNLI, and only marginally lower accuracy on BoolQ (-0.25), MRPC (-0.23), and RTE (-1.39). More importantly, our generated parameters substantially outperform the Mistral-7B base model across all tasks, with remarkable improvements ranging from $+11.99$ on Winogrande to $+29.15$ on SST-2 and $+28.30$ on GSM8K. The most significant gains appear in specialized rea-

Dataset	$R=2$		$R=4$		$R=8$		$R=16$		$R=32$	
	Original	Ours	Original	Ours	Original	Ours	Original	Ours	Original	Ours
BoolQ	90.52	91.43	90.81	91.27	91.01	90.76	91.07	90.74	90.00	88.44
MRPC	87.30	88.42	89.42	89.96	89.46	89.23	89.22	88.99	88.97	87.04

Table 4: Performance comparison between Original LoRA and our method on BoolQ and MRPC datasets across different LoRA ranks R. Results are reported in accuracy (%).

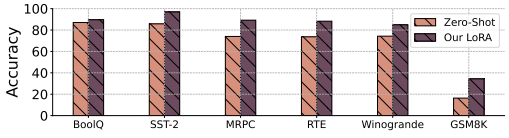

Figure 3: Accuracy comparison of our synthesized LoRA adapters against zero-shot base models on the unseen evolved Mistral continually pretrained on AlpacaGPT4.

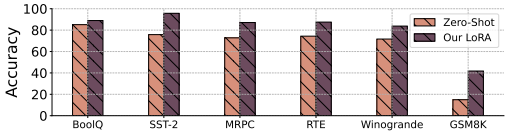

Figure 4: Accuracy comparison of our synthesized LoRA adapters against zero-shot base models on the unseen evolved Mistral continually pretrained on GPT4LLM.

soning and classification tasks, demonstrating that our conditional generation approach can effectively capture task-specific adaptations without traditional gradient-based fine-tuning. These results validate our proposed method's capability to generate high-quality LoRA parameters that match or occasionally exceed traditional fine-tuning approaches while offering the advantages of our scalable, conditional generation framework.

4.5 Generalization on Evolving Models

Setup and Datasets. In this experiment, we investigate the generalization capability of our conditional recurrent diffusion framework to synthesize LoRA Adapters for evolving base models not seen during training. PortLLM (Khan et al., 2024) shows that most of the closed source models go through data updates (i.e., evolution) that changes the model weights, hence requiring re-fine-tuning, which can be quite costly sometimes, or can be a hindrance due to the lack of data availability (time-sensitive access). To tackle this, we showcase the generalizability of our method in such an evolving setting, so that we do not have to re-fine-tune a newer evolved model on a downstream task, instead we can prompt our framework to generate the LoRA parameters that can be directly plugged in any iteration of the base model. Specifically, we use LoRA adapters trained on base models of the same architecture but different pertaining data to simulate the evolution. For the downstream tasks we utilize six datasets: BoolQ, SST-2, MRPC, RTE, Winogrande and GSM8K, and for the pretrained base model we utilize the 3 different time steps (evolutions): Base Mistral-7B Model ($t=0$), Mistral continually pretrained on {OpenOrca} (Lian et al., 2023) ($t=1$), and Mistral continually pretrained on {OpenOrca,OpenPlatypus (Lee et al., 2023)} ($t=2$). For each of these base models, we curate all 6 LoRAs, so we have a total of 18 LoRAs in our training set. Then we synthesize LoRA adapters for $t=3$ and $t=4$ for base models: Mistral continually pretrained on {OpenOrca, OpenPlatypus, AlpacaGPT (Taori et al., 2023)} and Mistral continually pretrained on {OpenOrca, OpenPlatypus, AlpacaGPT, GPT4LLM (Peng et al., 2023)}, which are both unseen base models for the trained pipeline. Then we compare the zero-shot performance of these pretrained models, against our generated LoRAs to showcase the improvement.

Results. The results for the experiment are summarized in Figures 3 and 4, comparing zero-shot performance on two unseen evolved base models (AlpacaGPT4 and GPT4LLM) against LoRA adapters generated by our framework ORAL. ❶**Performance Improvement over base models:** Our generated LoRA adapters demonstrate notable improvements, with the highest accuracy increases observed on SST-2 (10%) and GSM8K (30%) tasks. ❷**Generalization Capability:** The synthesized LoRA adapters effectively generalize to unseen evolved models ($t=3$ and $t=4$), significantly outperforming the zero-shot performance of base models. The results demonstrate that our conditional recurrent diffusion framework successfully transfers learned adaptation knowledge to new evolving base models without requiring additional fine-tuning, underscoring the robust generalization capabilities of our method. ❸**Consistency Across Models and Tasks:** The synthesized adapters consistently demonstrate improved or comparable performance across diverse NLP tasks. Particularly, the task-specific synthesized adapters for BoolQ, SST-2, MRPC, and GSM8K consistently outperform the baseline,

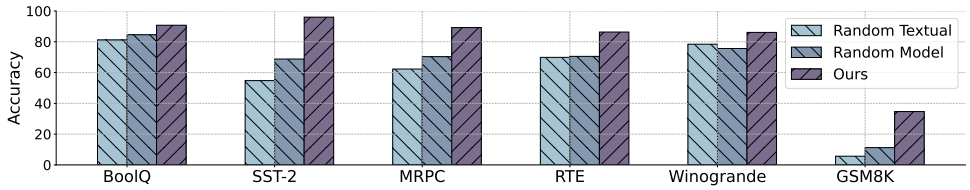

Figure 5: Ablation results showing accuracy comparisons across NLP tasks using random model embeddings, random textual embeddings, and our method with meaningful embeddings. Higher accuracy achieved by our conditional embeddings highlights their importance in guiding effective LoRA adapter generation.

indicating the broad applicability of ORAL.

4.6 Effect of Random Embeds.

In this ablation study, we evaluate the impact of random embeddings on our conditional recurrent diffusion framework by comparing performance across six NLP datasets: BoolQ, SST-2, MRPC, RTE, Winogrande, and GSM8K. Specifically, we compare the accuracy achieved by our method using meaningful conditional embeddings against two baseline scenarios—random model embeddings and random textual embeddings. The results are summarized in Figure 5.

From the results, we observe that using meaningful embeddings consistently outperforms both random embedding scenarios across all evaluated datasets. This indicates the importance of meaningful conditioning for synthesizing high-quality LoRA adapters. Interestingly, we see a more substantial performance drop with random textual embeddings than with random model embeddings, particularly noticeable in tasks requiring nuanced semantic understanding, such as GSM8K and SST-2. For example, in GSM8K, random textual embeddings show substantially lower accuracy than our method, underscoring the critical role of textual conditioning in generating high-quality, task-specific LoRA adapters. Conversely, random model embeddings still demonstrate moderate performance, suggesting that while base-model conditioning contributes to effective parameter generation, textual conditioning provides more crucial guidance.

4.7 Effect of LoRA Rank

In this ablation study, we analyze the impact of varying the LoRA rank (R) on the quality of the adapters generated by our conditional recurrent diffusion method. We compare the performance across two NLP datasets: BoolQ and MRPC, using LoRA ranks of 2, 4, 8, 16, and 32.

The results in Table 4 provide some interesting insights: ❶**Comparison with Original LoRA:** Our method consistently matches or surpasses the performance of Original LoRA across all ranks, showing the effectiveness of conditional generation even with limited parameter capacity. ❷**Performance Variation Across LoRA Ranks:** We observe that as the LoRA rank increases, performance initially improves and then slightly declines. For example, on BoolQ, our method achieves the highest accuracy at rank $R = 4(91.27\%)$ but then sees a slight drop at higher ranks (90.76% for $R = 8$ and further decreases at $R = 32$). Similar trends are observed on MRPC, with peak accuracy at rank 4. These observations highlight the importance of carefully choosing an optimal LoRA rank, balancing parameter efficiency and performance. Lower ranks appear sufficient to achieve competitive performance.

5 Conclusion

We introduced ORAL, a novel framework for conditional recurrent diffusion that enables scalable generation of Low-Rank Adaptation (LoRA) parameters at large scale. Our approach is designed to address the critical challenge of adapting to constantly evolving large language models without the computational burden of retraining. By incorporating both model architecture and textual task specifications as conditional inputs to the diffusion process, ORAL achieves remarkable flexibility and portability across model architectures while maintaining performance comparable or superior to traditionally trained LoRA parameters.

Our comprehensive experiments across various diverse tasks, three state-of-the-art pre-trained models (Stable-Diffusion2.1, Mistral, Qwen-7B-VL), and two modalities demonstrate the practical viability of our approach for real-world deployment. ORAL successfully scales to generate hundreds of millions of parameters while preserving adaptation quality. In the future, exploring the interpretability of generated parameters could provide insights into how different tasks influence parameter properties.

Limitations

While ORAL introduces a novel and effective framework for conditional, scalable, and portable LoRA parameter generation, several limitations should be acknowledged to provide a comprehensive perspective on the current work and to outline avenues for future research:

❶Generalization to Unseen Architectures and Tasks: Our experiments demonstrated ORAL's efficacy across a diverse set of LLMs, vision models, and various tasks. However, the extent to which ORAL can generalize to fundamentally different or highly novel model architectures not encountered during its training remains an area for further investigation. Similarly, its performance on tasks that are drastically different from those in its training data, or tasks requiring extremely nuanced or specialized adaptations, would need more extensive testing. The reliance on specific encoders (e.g., BERT for model architecture, CLIP for task prompts) for conditioning might also introduce limitations if these encoders do not adequately capture the semantics of new model types or task descriptions not well-represented in their own pertaining

❷Quality and Granularity of Conditioning Information: The performance of ORAL is contingent upon the quality and specificity of the base-model architectural encoding and the textual task descriptions. Ambiguous, incomplete, or poorly formulated textual prompts, or overly simplified architectural descriptions, might lead to suboptimal LoRA parameter generation. The current method of flattening structural metadata into a string for model conditioning, while effective, might not capture all architectural nuances or inter-dependencies that could influence optimal LoRA adaptation. Future work could explore more sophisticated and structured model conditioning mechanisms.

❸Computational Cost and Data Requirements for Training ORAL: While the inference of LoRA parameters using a trained ORAL model is efficient, and the use of LoRA itself is parameter-efficient, the initial training of the ORAL framework (the conditional recurrent diffusion model) can be computationally intensive. It requires a substantial and diverse dataset of existing LoRA parameters paired with their respective base models (and their architectural descriptions) and task specifications. Curating such a dataset across many models and tasks can be a significant undertaking, and the availability of such data may be a bottleneck for training ORAL for new, less common domains or model families.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. nocaps: novel object captioning at scale. International Conference on Computer Vision, pages 8947–8956.

Maor Ashkenazi, Zohar Rimon, Ron Vainshtein, Shir Levi, Elad Richardson, Pinchas Mintz, and Eran Treister. 2022. Nern - learning neural representations for neural networks. ArXiv, abs/2212.13554.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Universal guidance for diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 843–852.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. 2022. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. arXiv preprint arXiv:2201.06503.

Léon Bottou. 1991. Stochastic gradient learning in neural networks.

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.

Andrew Brock, Theodore Lim, James M. Ritchie, and Nick Weston. 2017. Smash: One-shot model architecture search through hypernetworks. ArXiv, abs/1708.05344.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. 2024. Video generation models as world simulators.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Adrien Corenflos, Zheng Zhao, Simo Särkkä, Jens Sjölund, and Thomas B Schön. 2024. Conditioning diffusion models by explicit forward-backward bridging. arXiv preprint arXiv:2405.13794.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

Prafulla Dhariwal and Alex Nichol. 2021a. Diffusion models beat gans on image synthesis. ArXiv, abs/2105.05233.

Prafulla Dhariwal and Alexander Nichol. 2021b. Diffusion models beat gans on image synthesis. In Advances in Neural Information Processing Systems, volume 34, pages 8780–8794. Curran Associates, Inc.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516.

Gabriel Eilertsen, Daniel Jönsson, Timo Ropinski, Jonas Unger, and Anders Ynnerman. 2020. Classifying the classifier: dissecting the weight space of neural networks. In ECAI 2020, pages 1119–1126. IOS Press.

Ziya Erkoç, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2023. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), pages 14254–14264.

Berthy T Feng, Jamie Smith, Michael Rubinstein, Huiwen Chang, Katherine L Bouman, and William T Freeman. 2023. Score-based diffusion models as principled priors for inverse imaging. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10520–10531.

Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In International Conference on Machine Learning.

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

Alex Graves. 2011. Practical variational inference for neural networks. In Neural Information Processing Systems.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. ArXiv, abs/2312.00752.

David Ha, Andrew M. Dai, and Quoc V. Le. 2016. Hypernetworks. ArXiv, abs/1609.09106.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2022. Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626.

Jonathan Ho. 2022. Classifier-free diffusion guidance. ArXiv, abs/2207.12598.

Jonathan Ho, Ajay Jain, and P. Abbeel. 2020. Denoising diffusion probabilistic models. ArXiv, abs/2006.11239.

Eliahu Horwitz, Bar Cavia, Jonathan Kahana, and Yedid Hoshen. 2024. Learning on model weights using tree experts. Preprint, arXiv:2410.13569.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. ArXiv, abs/2106.09685.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017a. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017b. Image-to-image translation with conditional adversarial networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5967–5976.

Christopher Jarzynski. 1997. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. Physical Review E, 56(5):5018.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. ArXiv, abs/2310.06825.

Xiaolong Jin, Kai Wang, Dongwen Tang, Wangbo Zhao, Yukun Zhou, Junshu Tang, and Yang You. 2024. Conditional lora parameter generation. Preprint, arXiv:2408.01415.

Rana Muhammad Shahroz Khan, Pingzhi Li, Sukwon Yun, Zhenyu Wang, Shahriar Nirjon, Chau-Wai Wong, and Tianlong Chen. 2024. Portllm: Personalizing evolving large language models with training-free and portable model patches. arXiv preprint arXiv:2410.10870.

Diederik P Kingma, Max Welling, and 1 others. 2013. Auto-encoding variational bayes.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2020. Diffwave: A versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761.

Ariel N. Lee, Cole J. Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms.

Alexander C Li, Mihir Prabhudesai, Shivam Duggal, Ellis Brown, and Deepak Pathak. 2023. Your diffusion model is secretly a zero-shot classifier. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2206–2217.

Muyang Li, Ji Lin, Chenlin Meng, Stefano Ermon, Song Han, and Jun-Yan Zhu. 2022. Efficient spatially sparse inference for conditional gans and diffusion models. Advances in neural information processing systems, 35:28858–28873.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. 2024a. Autoregressive image generation without vector quantization. ArXiv, abs/2406.11838.

Zexi Li, Lingzhi Gao, and Chao Wu. 2024b. Text-to-model: Text-conditioned neural network diffusion for train-once-for-all personalization. ArXiv, abs/2405.14132.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/datasets/Open-Orca/OpenOrca.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122.

Lequan Lin, Dai Shi, Andi Han, Zhiyong Wang, and Junbin Gao. 2024. Unleash graph neural networks from heavy tuning. ArXiv, abs/2405.12521.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. Preprint, arXiv:2304.08485.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan LI, and Jun Zhu. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In Advances in Neural Information Processing Systems, volume 35, pages 5775–5787. Curran Associates, Inc.

Shitong Luo and Wei Hu. 2021. Diffusion probabilistic models for 3d point cloud generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 2837–2845.

Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. 2020. Docvqa: A dataset for vqa on document images. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 2199–2208.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Radford M. Neal. 1995. Bayesian learning for neural networks.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In International Conference on Machine Learning.

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4195–4205.

William S. Peebles, Ilija Radosavovic, Tim Brooks, Alexei A. Efros, and Jitendra Malik. 2022. Learning to learn with generative models of neural network checkpoints. ArXiv, abs/2209.12892.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277.

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. International Journal of Computer Vision, 123:74 – 93.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952.

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. 2021. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21:140:1–140:67.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2):3.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. In Advances in Neural Information

Processing Systems, volume 32. Curran Associates, Inc.

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1530–1538, Lille, France. PMLR.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, and 1 others. 2022. Photorealistic text-to-image diffusion models with deep language understanding. Advances in neural information processing systems, 35:36479–36494.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande. Communications of the ACM, 64:99 – 106.

Tim Salimans and Jonathan Ho. 2022. Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512.

Konstantin Schürholt, Boris Knyazev, Xavier Gir'o i Nieto, and Damian Borth. 2022. Hyper-representations as generative models: Sampling unseen neural network weights. ArXiv, abs/2209.14733.

Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. 2021. Self-supervised representation learning on neural network weights for model characteristic prediction. NeurIPS, 34:16481–16493.

Mingjia Shi, Ruihan Lin, Xuxi Chen, Yuhao Zhou, Zezhen Ding, Pingzhi Li, Tong Wang, Kai Wang, Zhangyang Wang, Jiheng Zhang, and 1 others. 2025. Make optimization once and for all with fine-grained guidance. arXiv preprint arXiv:2503.11462.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 2256–2265, Lille, France. PMLR.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In Advances in Neural Information Processing Systems, volume 28. Curran Associates, Inc.

Haim Sompolinsky, Andrea Crisanti, and H. J. Sommers. 1988. Chaos in random neural networks. Physical review letters, 61 3:259–262.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502.

Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Bedionita Soro, Bruno Andreis, Hayeon Lee, Song Chong, Frank Hutter, and Sung Ju Hwang. 2024. Diffusion-based neural network weights generation. ArXiv, abs/2402.18153.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. ArXiv, abs/1905.00537.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. 2024. Neural network diffusion. Preprint, arXiv:2402.13144.

Kai Wang, Dongwen Tang, Wangbo Zhao, Konstantin Schürholt, Zhangyang Wang, and Yang You. 2025. Recurrent diffusion for large-scale parameter generation. Preprint, arXiv:2501.11587.

Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C Cattin. 2022. Diffusion models for medical anomaly detection. In International Conference on Medical image computing and computer-assisted intervention, pages 35–45. Springer.

Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. 2016. Attribute2image: Conditional image generation from visual attributes. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14, pages 776–791. Springer.

Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, Sam Dodge, Keen You, Zhen Yang, Aleksei Timofeev, Mingze Xu, Hong-You Chen, Jean-Philippe Fauconnier, Zhengfeng Lai, Haoxuan You, and 4 others. 2024. Mm1.5: Methods, analysis & insights from multimodal llm fine-tuning. Preprint, arXiv:2409.20566.

Bo Zhao, Robert M Gower, Robin Walters, and Rose Yu. 2023. Improving convergence and generalization using parameter symmetries. arXiv preprint arXiv:2305.13404.

Xinyu Zhao, Guoheng Sun, Ruisi Cai, Yukun Zhou, Pingzhi Li, Peihao Wang, Bowen Tan, Yexiao He, Li Chen, Yi Liang, and 1 others. 2024. Model-glue:

Democratized llm scaling for a large model zoo in the wild. arXiv preprint arXiv:2410.05357.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2223–2232.

A Use of Generative AI

To enhance clarity and readability, we utilized LLMs exclusively as a language polishing tool. Its role was confined to proofreading, grammatical correction, and stylistic refinement—functions analogous to those provided by traditional grammar checkers and dictionaries. This tool did not contribute to the generation of new scientific content or ideas, and its usage is consistent with standard practices for manuscript preparation.

B Expanded Related Works

Conditional Generation. Recently, conditional diffusion models (Corenflos et al., 2024; Bansal et al., 2023; Dhariwal and Nichol, 2021a; Ho, 2022; Nichol et al., 2021) have surfaced as the leading paradigm, fundamentally transforming expectations for conditional generation tasks. By intertwining the mathematical elegance of gradual noise reduction with advanced conditioning mechanisms, these models deliver unparalleled accuracy in converting conditioning signals—especially text descriptions—into coherent visual outputs. Our research extends the boundaries of conditional diffusion in a groundbreaking way, applying these techniques to generate neural network parameters. This marks a significant shift from conventional media-focused applications, highlighting the broader potential of conditional diffusion as a comprehensive framework for structured output synthesis across various problem domains.

Parameter Generation. Recent work has utilized diffusion models for parameter generation (Li et al., 2024a,b; Lin et al., 2024; Soro et al., 2024; Wang et al., 2024; Erkoç et al., 2023; Peebles et al., 2022). Notable progress has been made in representation learning; NeRN (Ashkenazi et al., 2022) showed that neural representations can directly encode weights of pre-trained convolutional networks, while Hyper-Representations (Schürholt et al., 2022) used autoencoders to capture latent distributions. Furthermore, conditional parameter generation has emerged as a promising area. COND P-DIFF (Jin et al., 2024) and Tina (Li et al., 2024b) introduced text-controlled methods for parameter generation, facilitating semantic guidance of weight synthesis. Yet, many current methods struggle with large-scale architectures like ResNet, ViT, or ConvNeXt. RPG (Wang et al., 2025) proposes a tokenization strategy with recurrent diffusion for large-scale generation, but conditional generation remains limited. Our work builds on these studies, proposing a new method for large-scale conditional parameter generation for LoRA parameters that maintains high performance across various tasks.

C Preliminary

C.1 Conditional Diffusion Models

A diffusion model (Ho et al., 2020) is the state-of-the-art generation model, normally consisting of a forward process that progressively adds noise to training data, and a reverse process that learns to remove noise step by step. When conditioned on auxiliary information, these models further enable controllable or guided generation, often referred to as a conditional diffusion model. In this section we go over the basics of (conditional) diffusion models.

Forward Process. Let $x_0 \in \mathcal{X}$ be a data sample drawn from the true distribution $q(x_0)$. For a total of T time steps, the forward process incrementally corrupts x_0 into $\{x_1, x_2, \ldots, x_T\}$ by adding Gaussian Noise:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \quad (5)$$

Here β_t is a small variance scheduled over steps $t = 1, \ldots, T$, and I is the identity matrix. After many steps x_T is nearly pure noise and thus easy to sample from a known Gaussian. Crucially, this forward process does not require training: it is defined by a simple Markov Chain. An important convenience is that at any step t, one can directly draw x_t from x_0 via

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1+\alpha_t}\epsilon, \ \alpha_t = \prod_{s=1}^{t}(1-\beta_s)$$

This closed-form reparameterization is often used to simplify training objectives.

Reverse Process. While the forward process is explicitly defined, the reverse process is more challenging: given a noisy sample x_t, the model must

predict x_{t-1}. One assumes a Gaussian form:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right) \tag{6}$$

where μ_θ and Σ_θ are learned neural networks with parameters θ. Training proceeds by matching this learned reverse process to the true (unknown) reverse distribution. A common simplified objective is the mean-squared error between the model's predicted noise and the actual noise injected in the forward process:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{7}$$

where x_t is obtained by sampling noise ϵ and applying the forward formula above. Minimizing $\mathbb{L}(\theta)$ forces the reverse process to remove noise properly at each time step. At inference time, one starts from a random Gaussian sample x_T and sequentially applies $p_\theta(x_{t-1}, x_t)$ to denoise it from $t = T$ down to $t = 1$. The result is a generated sample x_0 that (approximately) follows the data distribution.

Conditional Diffusion Models. To allow controllable generation, a conditional diffusion model augments each step with a condition c. This condition can represent any auxiliary information relevant to the task—for instance, a text prompt, a domain label, or a style embedding. The forward process typically remains unchanged:

$$q(x_t|x_{t-1}, c) = q(x_|x_{t-1}) \tag{8}$$

but the reverse process is now defined by

$$p_\theta(x_{t-1}|x_t, c) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t, c), \Sigma_\theta(x_t, t, c)\right) \tag{9}$$

We then modify the training objective so that at each step t, the model sees not only the noisy sample x_t but also the condition c. A practical variant of the training loss becomes:

$$\mathcal{L}_{cond}(\theta) = \sum_{t=1}^{T} \mathbb{E}_{x_0,\epsilon,c}\left[\|\epsilon - \epsilon_\theta(x_t, t, \tau(c)\|^2\right] \tag{10}$$

where $\tau(c)$ is an encoder or projection function that maps c into a representation usable by the network. At sampling time, we repeatedly apply the reverse distribution

$$x_t \sim p_\theta(x_{t-1}|x_t, c), \quad t = T, \ldots, 1 \tag{11}$$

ensuring that the resulting x_0 is guided toward the desired condition. This approach thus enables flexible, high-quality generation with explicit control over the style, content, or any other information captured by c.

C.2 Low-Rank Adaptation

Let $W_0 \in \mathbb{R}^{d\times d}$ be a pretrained weight matrix in a transformer model—for example, the weight of a particular layer. A typical fine-tuning procedure would update W_0 directly, requiring substantial storage and computation for large-scale models. Low-Rank Adaptation or simply LoRA (Hu et al., 2021) offers a more parameter-efficient approach by assuming that the update $\Delta W \in \mathbb{R}^{d\times d}$ to W_0 (the difference between W_0 and the fully fine-tuned matrix) can be factorized into a low-rank form:

$$\Delta W = BA \tag{12}$$

where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$ are trainable matrices, and $r \ll d$ is the rank of the decomposition. This drastically cuts the number of learnable parameters: instead of adjusting all d^2 entries of ΔW, LoRA only requires tuning $2rd$ parameters (the entries in A and B). Thus, after fine-tuning, the adapted weight matrix is

$$W_{new} = W_0 + \Delta W = W_0 + BA \tag{13}$$

enabling the model to learn task-specific adaptations using significantly fewer parameters. This approach maintains much of the flexibility of conventional fine-tuning while being more efficient in both computation and memory.