

# An Analytical and Comparative Review of Deep Learning-Based Sequence Generation Models

Roohullah Fasihi<sup>1</sup> , Ahmad Farid Rahimi<sup>2</sup> 

1. Visiting Lecturer, Department of Software Engineering, Faculty of Computer Science, Kateb University, Kabul, Afghanistan (Corresponding Author). Email: [Fasihi20@gmail.com](mailto:Fasihi20@gmail.com)
2. Visiting Lecturer, Department of Software Engineering, Faculty of Computer Science, Kateb University, Kabul, Afghanistan.

Article Info	ABSTRACT
<p><b>Article type:</b> Research Article</p> <p><b>Article history:</b> Received: 22/02/2026 Received in revised form: 23/02/2026 Accepted: 12/03/2026 Available online: 19/03/2026</p> <p><b>Keywords:</b> Sequence generation, sequence generation model, Generative Adversarial Nets, GAN, SeqGAN, LSTM</p>	<p>Sequence generation has become a central problem in deep learning with applications in natural language processing, speech synthesis, and time-series analysis. This paper presents a structured analytical review of three major deep learning-based sequence generation paradigms: Long Short-Term Memory (LSTM) networks, Generative Adversarial Networks (GANs), and SeqGAN. The study compares these models in terms of learning mechanism, handling of discrete and continuous data, computational complexity, training stability, and sequence-level optimization capability. The analysis demonstrates that LSTM models provide stable probabilistic learning but suffer from exposure bias, while GANs achieve powerful distribution modeling yet face instability and limitations in discrete sequence generation. SeqGAN addresses this limitation by integrating reinforcement learning and policy gradient optimization, enabling adversarial training for discrete sequences at the cost of higher computational cost and variance in training. The paper further identifies the theoretical trade-offs among these approaches and discusses their suitability for different sequence generation tasks. Finally, future research directions involving transformer-based architectures and hybrid adversarial-reinforcement learning models are highlighted.</p>

**Cite this article:** Fasihi, R. & Rahimi, A. (2026). An Analytical and Comparative Review of Deep Learning-Based Sequence Generation Models, *Kateb Scientific-Research Journal of Technology and Engineering*, 1 (1), 23-34.



## Introduction

Sequence generation is a fundamental problem in machine learning and artificial intelligence, involving the modeling and prediction of ordered data such as text, speech, music, and time-series signals. Unlike static data modeling, sequence generation requires capturing temporal dependencies and long-range structural relationships within data.

Recurrent Neural Networks (RNNs) have traditionally served as the primary framework for sequence modeling. However, standard RNN architectures suffer from limitations in learning long-term dependencies due to vanishing and exploding gradient problems [1]. To address these issues, the Long Short-Term Memory (LSTM) architecture was introduced [2], incorporating gated memory cells that enable the preservation of contextual information over extended time intervals. LSTM-based models have demonstrated strong performance in modeling complex sequential data [3].

Generative Adversarial Networks (GANs) introduced an adversarial learning paradigm capable of modeling high-dimensional data distributions. However, traditional GANs face major limitations in discrete sequence generation due to the non-differentiability of sampling operations [4].

To overcome this limitation, SeqGAN was proposed as an extension of GAN [5]. SeqGAN formulates sequence generation as a reinforcement learning problem. The generator is modeled as a stochastic policy and optimized using policy gradient methods. This approach enables adversarial training in scenarios involving discrete token sequences.

Despite their widespread adoption, a structured comparison of the theoretical foundations, training mechanisms, and limitations of these models remains necessary. This paper provides an analytical comparison of LSTM, GAN, and SeqGAN as major paradigms of deep learning-based sequence generation. We systematically examine their learning strategies, ability to handle discrete and continuous data, computational complexity, and training stability.

The remainder of this paper is organized as follows: Section 2 discusses sequence generation using LSTM networks. Section 3 presents the GAN framework. Section 4 introduces SeqGAN and its reinforcement learning formulation. Section 5 provides a comparative analysis and discussion. Finally, Section 6 concludes the paper and outlines potential future research directions.

## Research Gap and Motivation

Although numerous studies have investigated sequence generation using LSTM networks, GANs, and reinforcement learning-based adversarial models, several important gaps remain in the literature. First, many previous studies focus primarily on performance evaluation rather than providing a structured theoretical comparison of the underlying learning paradigms. Second, many comparative studies focus on empirical performance while overlooking trade-offs among stability, adversarial optimization, and computational cost. Third, limited review studies analyze how reinforcement learning bridges the gap between likelihood-based sequence modeling and adversarial learning for discrete sequence generation. Therefore, this paper aims to provide a structured analytical comparison of LSTM, GAN, and SeqGAN frameworks from theoretical, computational, and optimization perspectives.

Modern transformer-based language models such as BERT have further improved contextual sequence representation learning in natural language processing tasks [6].

The novelty of this study lies in its structured comparison of probabilistic, adversarial, and reinforcement learning-based sequence generation models. Unlike many previous review studies that focus primarily on empirical performance, this paper emphasizes optimization mechanisms, statistical trade-offs, computational complexity, and training stability across the three paradigms.

## Problem Statement

Sequence generation remains a challenging task in deep learning due to the need for modeling long-range dependencies, handling discrete sequential outputs, and maintaining training stability. Traditional likelihood-based models such as LSTM networks provide stable learning but often suffer from exposure bias and token-level optimization limitations. On the other hand, adversarial approaches such as GANs offer powerful distribution learning but cannot directly handle discrete token generation due to gradient discontinuity. SeqGAN addresses these limitations through reinforcement learning but introduces higher computational complexity and optimization instability. Despite extensive research in this field, a comprehensive analytical comparison of these paradigms remains limited. This study therefore investigates the theoretical foundations, optimization mechanisms, strengths, and limitations of LSTM, GAN, and SeqGAN models in sequence generation tasks.

## Theoretical Framework

The theoretical foundation of this study is based on three major paradigms in deep learning-based sequence generation: probabilistic autoregressive modeling, adversarial learning, and reinforcement learning-based sequence optimization.

The first paradigm is represented by LSTM networks, which rely on recurrent probabilistic modeling and maximum likelihood estimation for sequence prediction. This framework focuses on learning temporal dependencies through gated memory mechanisms.

The second paradigm is adversarial learning, represented by GANs, where a generator and discriminator are jointly optimized in a minimax framework to approximate the underlying data distribution.

The third paradigm combines adversarial learning with reinforcement learning, as implemented in SeqGAN. In this framework, sequence generation is formulated as a Markov Decision Process (MDP), and the generator is optimized using policy gradient methods based on sequence-level reward signals.

The conceptual relationship among these paradigms can be summarized as follows:

- **LSTM** → Probabilistic token-level learning
- **GAN** → Adversarial distribution learning
- **SeqGAN** → Sequence-level adversarial reinforcement learning

This framework provides the analytical basis for comparing the strengths, limitations, optimization strategies, and computational characteristics of the studied models.

## Research Methodology

This study adopts a Systematic Literature Review (SLR) methodology to analyze and compare major deep learning–based sequence generation models. The review process was conducted in several stages.

First, relevant scientific publications were collected from major academic databases including IEEE Xplore, SpringerLink, ScienceDirect, ACM Digital Library, and arXiv.

Second, article selection was performed using predefined inclusion criteria:

- Studies related to sequence generation using LSTM, GAN, or SeqGAN
- Peer-reviewed journal articles and highly cited conference papers
- Studies published between 2013 and 2024
- Articles written in English

The exclusion criteria included:

- Studies unrelated to deep learning sequence generation
- Duplicate publications
- Papers lacking sufficient methodological details

More than 40 papers were initially identified, and 18 highly relevant studies were selected for detailed review.

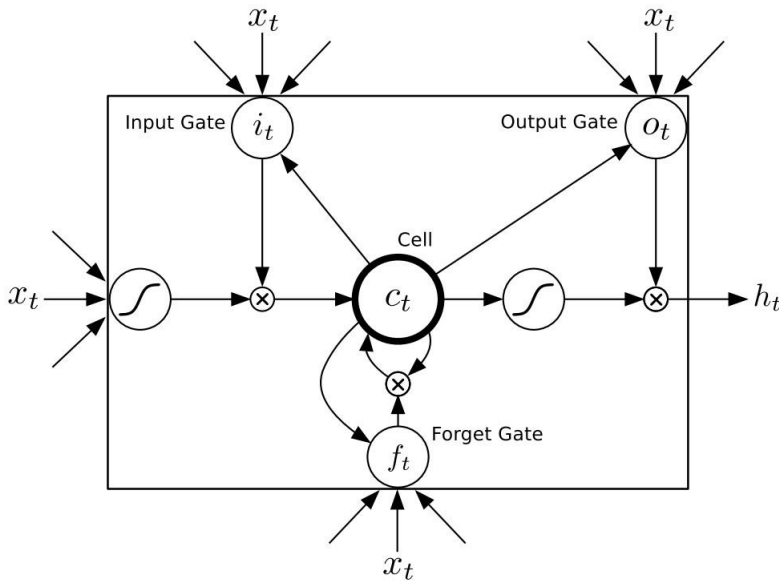
The selected studies were comparatively analyzed based on:

- Learning mechanism
- Optimization strategy
- Capability in handling discrete and continuous data
- Computational cost
- Training stability
- Sequence-level performance characteristics

The analytical framework of this review follows a qualitative comparative approach rather than empirical experimentation.

## Recurrent Neural Networks and LSTM for Sequence Generation

Recurrent Neural Networks (RNNs) are a class of neural architectures specifically designed for modeling sequential data. Unlike feedforward neural networks, RNNs incorporate recurrent connections that allow information to persist across time steps. At each time step  $t$ , the hidden state  $h_t$  is computed as a function of the current input  $x_t$  and the previous hidden state  $h_{t-1}$ , enabling the network to capture temporal dependencies within the sequence.



**Figure 1:** Long Short-Term Memory (LSTM) Cell Architecture

Note. Adapted from “Long Short-Term Memory” by the author based on [2].

In sequence generation tasks, the RNN defines a conditional probability distribution over sequences:

$$P(x_1, x_2, \dots, x_T) = \prod_{t=1}^T P(x_t | x_1, \dots, x_{t-1})$$

Despite their theoretical capability to model arbitrarily long dependencies, standard RNNs face practical difficulties during training. In particular, the vanishing and exploding gradient problems significantly limit their ability to learn long-term dependencies through gradient descent [1]. During Backpropagation Through Time (BPTT), gradients may vanish or explode, leading to unstable learning [7].

### Long Short-Term Memory (LSTM)

To address these limitations, the Long Short-Term Memory (LSTM) architecture was introduced [2]. LSTM augments the standard RNN structure with memory cells and gating mechanisms that regulate information flow. Each LSTM cell contains gating mechanisms and a memory cell that regulate information flow across time steps.

These gates control the extent to which new information is stored, previous information is retained, and hidden states are exposed to the next layer.

The gating mechanism enables LSTM networks to preserve relevant information over extended time intervals while discarding irrelevant signals. This significantly mitigates the vanishing gradient problem and improves the stability of sequence modeling.

## Training Considerations

LSTM networks are typically trained using maximum likelihood estimation by minimizing the negative log-likelihood of the observed sequences [3]. The training process relies on Backpropagation Through Time (BPTT) [7]. However, due to potential gradient instability, gradient clipping is commonly applied to prevent numerical divergence.

While LSTM-based models are effective in modeling both continuous and discrete sequential data, they rely on likelihood-based training, which may lead to exposure bias and accumulation of prediction errors during sequence generation.

## Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Ian Goodfellow et al. [4], represent a fundamentally different paradigm for generative modeling compared to likelihood-based approaches such as LSTM networks. Instead of explicitly estimating the data distribution through maximum likelihood, GANs learn to generate data by means of an adversarial training process.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))]$$

## Adversarial Learning Framework

The GAN framework consists of two neural networks:

- **A generator  $G$** , which maps a random noise vector  $z \sim p_z(z)$  to the data space.
- **A discriminator  $D$** , which estimates the probability that a given sample originates from the real data distribution rather than from the generator

The two networks are trained simultaneously in a minimax game [4]. The discriminator distinguishes real samples from generated ones, while the generator attempts to produce realistic data samples.

The generator learns to approximate the data distribution using gradient feedback from the discriminator, eliminating the need for explicit likelihood estimation [4].

GANs are typically trained using minibatch stochastic gradient descent through iterative optimization of the generator and discriminator networks in a minimax framework [4].

## Advantages of GAN-Based Modeling

GANs offer several important advantages in generative modeling. Unlike traditional probabilistic generative approaches, they avoid explicit probability density estimation and do not require inference procedures during training. Furthermore, GANs can model complex high-dimensional distributions, and data generation requires only a single forward pass. [4]. These characteristics have made GANs highly successful in continuous data domains such as image generation.

## Training Challenges and Limitations

Despite their conceptual elegance, GANs are notoriously difficult to train. Common issues include:

- Training instability

- Mode collapse (where the generator produces limited diversity)
- Sensitivity to hyperparameter selection
- Difficulty in achieving equilibrium between generator and discriminator

Furthermore, a critical limitation of standard GANs lies in their incompatibility with discrete data generation. GAN training relies on gradient propagation, but discrete token sampling is non-differentiable, preventing direct optimization in text generation tasks.

This limitation becomes particularly problematic in sequence generation tasks, where decisions are made step by step.

From a sequence modeling perspective, standard GANs face two primary limitations:

- Inability to directly handle discrete tokens due to gradient discontinuity.
- Lack of intermediate reward signals for partially generated sequences.

These limitations motivated the development of SeqGAN [5], which integrates reinforcement learning techniques to enable adversarial training in discrete sequence generation tasks. The next section presents this extension in detail.

In contrast with likelihood-based frameworks such as LSTM, GANs adopt a fundamentally different learning paradigm. Unlike likelihood-based models, GANs rely on adversarial optimization and do not define an explicit likelihood function. Their training process is often unstable and sensitive to the balance between generator and discriminator.

However, GANs offer several significant computational and statistical advantages:

- **Efficient gradient computation:** Instead of relying on Markov chains for sampling, GANs use backpropagation to compute gradients directly.
- **No inference during learning:** Unlike variational methods, GANs do not require inference networks during training.
- **Architectural flexibility:** GANs can incorporate diverse loss functions and network architectures without restrictive constraints.
- **Statistical advantage:** The generator learns from gradients provided by the discriminator rather than directly from data examples, enabling it to capture subtle statistical patterns in the data distribution without explicit density estimation

## SeqGAN: Reinforcement Learning Formulation for Discrete Sequence Generation

While Generative Adversarial Networks (GANs) [4] have demonstrated remarkable success in modeling continuous data distributions, their direct application to discrete sequence generation is fundamentally limited by the non-differentiability of sampling operations. In domains such as text generation, the generator must sample tokens from a categorical distribution, which prevents gradients from propagating from the discriminator to the generator through standard backpropagation.

To address this limitation, SeqGAN [5] reformulates sequence generation as a reinforcement learning (RL) problem. The generator is modeled as a stochastic policy, while the discriminator provides reward signals to guide learning.

SeqGAN combines adversarial learning with policy gradient reinforcement learning to enable discrete sequence generation [5].

### Limitation of Standard GAN in Discrete Sequence Generation

In the standard GAN framework [4], the generator receives gradient signals from the discriminator via backpropagation. This mechanism assumes that the generator output is differentiable with respect to its parameters.

However, in discrete sequence generation, the generator defines a probability distribution over tokens:

$$G_{\theta}(y_t | y_{1:t-1})$$

At each timestep, a token is sampled from this distribution. Since sampling from a categorical distribution is non-differentiable, gradients cannot be propagated through the discrete output tokens.

Moreover, the discriminator provides feedback only for complete sequences. This results in sparse reward signals, making it difficult to assign credit to intermediate decisions during generation.

These two limitations — gradient discontinuity and lack of intermediate reward — motivate the reinforcement learning reformulation proposed in SeqGAN [5].

### Reinforcement Learning Reformulation

SeqGAN [5] models sequence generation as a finite-horizon Markov Decision Process (MDP). The components of this formulation are defined as follows:

State  $s_t$ : partially generated sequence  $(y_1, \dots, y_{t-1})$

Action  $a_t$ : selection of the next token  $y_t$

Policy  $G_{\theta}$ : the generator network

Reward  $R(Y_{1:T})$ : discriminator output for a complete sequence

The objective is to maximize the expected reward over generated sequences:

$$J(\theta) = \mathbb{E}_{Y_{1:T} \sim G_{\theta}}[R(Y_{1:T})]$$

Since the generator is stochastic, SeqGAN employs policy gradient methods inspired by reinforcement learning theory. The gradient of the objective function is computed as:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{Y_{1:T} \sim G_{\theta}}[R(Y_{1:T}) \nabla_{\theta} \log G_{\theta}(Y_{1:T})]$$

Using the autoregressive factorization:

$$\log G_{\theta}(Y_{1:T}) = \sum_{t=1}^T \log G_{\theta}(y_t | y_{1:t-1})$$

### Monte Carlo Rollout for Intermediate Reward Estimation

A major challenge in SeqGAN [5] is that the discriminator provides reward only for complete sequences. To estimate the expected future reward for intermediate states, SeqGAN employs Monte Carlo (MC) rollouts.

The action-value function is defined as:

$$Q(s_t, a_t) = \mathbb{E}[R(Y_{1:T}) \mid s_t, a_t]$$

SeqGAN estimates intermediate rewards using Monte Carlo rollout simulations [5].

$$Q(s_t, a_t) = \frac{1}{N} \sum_{n=1}^N D_\phi(Y_{1:T}^{(n)})$$

where  $D_\phi$  denotes the discriminator [5].

This Monte Carlo approximation introduces variance in gradient estimation.

### Strengths and Limitations of SeqGAN

Although SeqGAN [5] successfully enables adversarial learning for discrete sequences, it introduces several optimization challenges.

#### High Variance Gradient Estimation

Policy gradient methods are known to suffer from high variance, particularly when rewards are sparse. The Monte Carlo rollout mechanism further increases variance, potentially destabilizing training.

#### Computational Complexity

Monte Carlo rollout procedures increase the computational cost of SeqGAN compared to standard LSTM training.

$$O(TN)$$

This is considerably more expensive than maximum likelihood training in LSTM-based models (Graves, 2013), which relies solely on Backpropagation Through Time (BPTT) [5].

#### Adversarial Instability

SeqGAN inherits the minimax optimization instability of GANs [4]. Maintaining balance between generator and discriminator remains critical. Overtraining the discriminator may lead to vanishing reward signals, while undertraining may result in noisy gradients and mode collapse.

#### Sensitivity to Training Schedule

The frequency of updating the generator versus the discriminator significantly affects convergence. Improper scheduling may lead to oscillatory behavior or divergence.

From an analytical perspective, SeqGAN offers several important advantages for discrete sequence generation. By combining reinforcement learning with adversarial training, SeqGAN optimizes sequence-level objectives and reduces exposure bias. In addition, the framework avoids explicit likelihood computation and allows adversarial learning in scenarios involving discrete sequential outputs [5].

## Analytical Comparison with LSTM-Based Models

Unlike LSTM-based sequence generation trained with maximum likelihood [3], [2] . which optimizes token-level log-probabilities, SeqGAN [5] optimizes expected sequence-level reward.

Maximum likelihood estimation provides stable low-variance gradients but suffers from exposure bias and training–inference mismatch. SeqGAN addresses this mismatch by optimizing generation behavior directly, yet introduces higher variance and greater computational burden.

Therefore, SeqGAN can be interpreted as a trade-off between statistical stability (MLE-based LSTM) and global adversarial optimization.

**Table 1** presents a structured comparison between LSTM, GAN, and SeqGAN based on learning strategy, computational cost, training stability, and sequence-level optimization capability.

**Table 1**  
 Comparative Analysis of LSTM, GAN, and SeqGAN Models

Feature	LSTM	GAN	SeqGAN
<b>Learning Strategy</b>	Maximum Likelihood	Adversarial Learning	Reinforcement + Adversarial
<b>Data Type</b>	Continuous & Discrete	Mostly Continuous	Discrete Sequences
<b>Training Stability</b>	High	Low	Medium-Low
<b>Computational cost</b>	Moderate	Moderate	High
<b>Exposure Bias</b>	Yes	No	Reduced
<b>Gradient Variance</b>	Low	Medium	High
<b>Sequence-Level Optimization</b>	Weak	Medium	Strong

Source: Compiled by the author based on [3], [4], [2] and [5]

## Future Work

Future research may extend this comparative framework toward transformer-based architectures and attention-based sequence generation models, which have recently become dominant in natural language processing tasks [8]. These models represent different trade-offs among statistical efficiency, computational cost, and optimization stability. Another promising direction involves integrating diffusion-based generative models with sequence modeling frameworks.

Recent advances in large language models have demonstrated that transformer-based architectures can significantly improve long-range sequence modeling and contextual generation capabilities [9].

Another promising research direction involves integrating diffusion-based generative models with sequence generation frameworks to improve training stability and generation diversity [10].

## Conclusion

This paper provided an analytical and comparative review of three major deep learning paradigms for sequence generation: LSTM-based recurrent networks [2], Generative Adversarial Networks (GANs) [4], and SeqGAN [2]. Each of these approaches represents a distinct learning philosophy with different theoretical assumptions, optimization strategies, and practical implications.

LSTM-based models provide stable probabilistic sequence modeling and effectively capture long-range dependencies through recurrent memory mechanisms. However, their reliance on maximum likelihood training may lead to exposure bias and discrepancies between training and inference phases [3], [2]

GANs introduce an adversarial learning paradigm capable of modeling complex high-dimensional data distributions without explicit likelihood estimation. However, their dependence on differentiable outputs limits their direct applicability to discrete sequence generation, while training instability remains a major challenge [4].

SeqGAN extends adversarial learning to discrete sequence generation by integrating reinforcement learning and policy gradient optimization. While this approach reduces exposure bias and enables sequence-level optimization, it also introduces higher computational complexity and training instability [5].

From a theoretical perspective, these three models can be interpreted as representing different trade-offs between statistical efficiency, computational cost, and optimization stability. LSTM-based models provide stable and statistically efficient learning but may suffer from objective mismatch. GANs offer powerful distribution matching for continuous data but lack direct applicability to discrete tokens. SeqGAN bridges this gap, achieving adversarial sequence-level optimization while introducing reinforcement learning-related challenges.

In conclusion, the choice among LSTM, GAN, and SeqGAN depends on the nature of the data (continuous vs. discrete), the desired optimization objective (likelihood vs. adversarial reward), and computational constraints. Understanding these trade-offs is important for selecting suitable sequence generation architectures.

## References

- [1] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, doi: 10.1109/72.279181.
- [2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [3] Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850. <https://arxiv.org/abs/1308.0850>
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.
- [5] L. Yu, W. Zhang, J. Wang, and Y. Yu, “SeqGAN: Sequence generative adversarial nets with policy gradient,” arXiv preprint arXiv:1609.05473, 2016. [Online]. Available: <https://arxiv.org/abs/1609.05473>
- [6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [7] R. J. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networks,” in *Backpropagation: Theory, Architectures, and Applications*, 1995, pp. 433–486.
- [8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- [9] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- [10] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695. <https://doi.org/10.1109/CVPR52688.2022.01042>