Article

Not peer-reviewed version

Generative AI for Research: Paradigms, Tasks, Evaluation, and Best Practices

Antu Kumar Guha , Prosenjit Das , Amit Bala , Sabah Ummie * , Muhammad Enayetur Rahman * , Md Nurul Absar Siddiky * , Muhammad Rezaur Rahman

Posted Date: 6 November 2025

doi: 10.20944/preprints202511.0421.v1

Keywords: generative AI; generative adversarial network; diffuision model; variational autoencoder

 doi:10.20944/preprints202511.0421.v1

Article

Generative AI for Research: Paradigms, Tasks, Evaluation, and Best Practices

Antu Kumar Guha [1,†], Prosenjit Das [1,†], Amit Bala [1], Sabah Ummie [2,*], Muhammad Enayetur Rahman [3], Md Nurul Absar Siddiky [4,*] and Muhammad Rezaur Rahman [5]

[1] Electronics and Communication Engineering, Khulna University, Khulna, 9208, Bangladesh
[2] Electrical and Computer Engineering, University of Rhode Island, Kingston, RI 02881, USA
[3] Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, 23509, USA
[4] Electrical and Computer Engineering, University of North Carolina at Charlotte, Charlotte, NC, 28223, USA
[5] Research and Innovation Department, Agile Crafts, Dhaka, 1219, Bangladesh
* Correspondence: sabah.ummie@gmail.com (S.U.); msiddiky@uncc.edu (M.N.A.S.)

Abstract

Generative Artificial Intelligence (AI) is a quickly evolving sphere that has allowed generating real data and making progress in the area of producing high-quality images, text-to-image translation, and predicting time series. Nevertheless, the growing variety of model architectures has turned the choice of suitable generative methods, the tailoring of these approaches, and their assessment into a significant concern of the researchers. The presented literature review provides a task-based and systematic taxonomy of generative modeling, which provides a comparative summary of the most common paradigms such as the Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), the Normalizing Flows, and the Diffusion Models. The survey is structured in the context of the main research activities: high-fidelity data synthesis, controllable and conditional generation, sequential and structured data modeling, representation learning and disentanglement, data augmentation and simulation, and probabilistic density estimation. In every task, we discuss technical goals, essential practices, measures of evaluation, and best practices. The paper also includes cross-cutting techniques such as utilizing foundation models, data curation, large-scale data processing, computational scaling, reproducibility and responsible AI practices in addition to model selection. Lastly, we present the modern technical issues and areas of future research pointing to outstanding questions concerning controllability, evaluation, efficiency, and ethical applications.This paper serves as a comprehensive resource for researchers and practitioners in the field of generative modeling.

Keywords: Generative AI; Generative Adversarial Network; Diffuision Model; Variational Autoencoder

1. Introduction

Generative AI models, categorized as unsupervised learning, adapt the technique of creating synthesized data by mimicking data existing in the real world. The rapid evolution of the model has notable enhancements in tasks like high quality image generation, text-to-image conversion, time series prediction etc., through extensive research and development. However, selecting an appropriate model, adapting and evaluating the optimal approach for specific tasks have become a significant challenge for researchers due to the diverse architecture. It is crucial to decide the appropriate model selection based on the nature of the problem and the type of data involved.

This literature review provides a task-centric comparative analysis of major generative paradigms. It focuses on the technical trade-offs, adaptation best practices, and evaluation strategies relevant to common AI/ML research problems, serving as a practical guide for informed model selection and effective research application. The analysis in this paper is structured around key technical problem types, including high-fidelity data synthesis, controllable and conditional generation, sequential and structured data modeling, representation learning, data augmentation, and probabilistic density

 doi:10.20944/preprints202511.0421.v1

estimation. For each task, we delve into the leading generative models, discussing their underlying mathematical principles, training dynamics, practical applications, and inherent trade-offs.

In this work we aim to serve as guide for researchers and practitioner by focusing on the technical insights, best practices and evaluation strategies. Our goal is to facilitate model selection strategies makinf generative AI applications more effective in diverse research contexts.

2. A Pragmatic Taxonomy of Generative Tasks

2.1. Key Generative Research Tasks

The scope of key generative task has been represented in Figure 1. Formally define the technical characteristics and objectives of core tasks:

Figure 1. Scope of Core Generative Tasks

High-Fidelity Data Synthesis: High-fidelity data synthesis refers to the process of generating artificial data that closely replicates the perceptual qualities and realism of real-world datasets. The objective is to produce synthetic outputs that are virtually indistinguishable from authentic data when evaluated by human observers or advanced computational models. This process typically involves leveraging sophisticated machine learning techniques such as generative adversarial networks (GANs) [1], diffusion models [2,3], or variational autoencoders (VAEs) [4] to capture fine-grained details, structural patterns, and statistical properties inherent in real data distributions. High-fidelity synthesis not only enhances the realism of synthetic data but also ensures its applicability in tasks such as training machine learning models, simulation, and augmentation, where high perceptual quality is paramount. Achieving such fidelity requires optimizing for both low-level features (e.g., texture, color consistency) and high-level semantic coherence (e.g., object integrity, scene realism) [5].

Controllable/Conditional Generation: Controllable or conditional generation refers to the process of producing synthetic data—such as images, text, or audio—that adheres to specific inputs, attributes, or constraints defined prior to or during the generation process. This paradigm ensures that the generated outputs are not purely random but are systematically aligned with user-provided conditions, including class labels, semantic descriptions, structural layouts, or stylistic features. Achieving effective conditional generation typically involves conditioning mechanisms integrated into generative models, such as Conditional Generative Adversarial Networks (cGANs) [6], controllable Variational Autoencoders (VAEs) [7], or guided diffusion models [8–10]. These models are trained to not only capture the underlying data distribution but also to respect and incorporate conditioning information throughout the synthesis process. The ability to control generation is critical in applications such as personalized content creation, targeted data augmentation, and fine-grained editing, where adherence to specified attributes significantly enhances usability and realism.

Sequential & Structured Data Modeling: Sequential and structured data modeling refers to the development of machine learning systems capable of capturing complex dependencies inherent in ordered or relational data formats, such as time series, natural language, biological sequences, and graphs. Sequential models, such as Recurrent Neural Networks (RNNs) [11], Long Short-Term Memory networks (LSTMs) [12], and Transformer architectures [13], are designed to learn from temporal or

 doi:10.20944/preprints202511.0421.v1

ordered patterns, where the current state depends on historical information. In parallel, structured models address data with intrinsic relational structures, including social networks, molecular graphs, and knowledge graphs, using frameworks like Graph Neural Networks (GNNs) [14] and Graph Attention Networks (GATs) [15]. Capturing dependencies effectively enables models to perform tasks such as sequence prediction, graph classification, link prediction, and spatiotemporal forecasting with greater fidelity and generalization ability. Mastering sequential and structured data modeling is critical for advancing fields ranging from natural language processing and recommender systems to drug discovery and traffic prediction.In Figure 2 we have shown Sequential vs Structured Data Modeling Architectures.

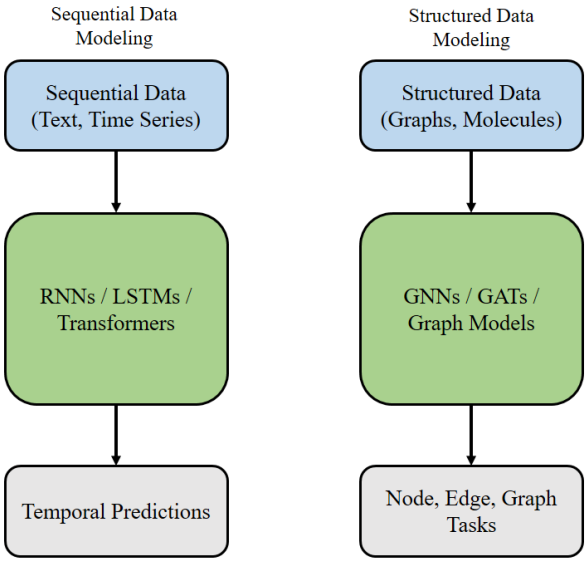

Figure 2. Sequential vs Structured Data Modeling Architectures

Representation Learning & Disentanglement: Representation learning refers to the development of models that automatically discover useful features or representations from raw data, facilitating downstream tasks such as classification, generation, or reasoning. A particularly important aspect of advanced representation learning is disentanglement, which aims to separate the underlying generative factors of variation within data into distinct, interpretable components [16], [17]. Learning structured and meaningful latent spaces enables models not only to generalize better across tasks but also to achieve greater robustness and interpretability. Techniques such as Variational Autoencoders (VAEs) [4], β-VAEs [18], and InfoGANs [19] have been widely explored to promote disentangled representations, often leveraging constraints on the latent space or maximizing mutual information between latent codes and observations. Achieving high-quality disentangled representations is vital for controllable generation, transfer learning, fairness, and scientific discovery, where understanding the latent causal structure behind the data is essential. Illustration of latent variable flow in a disentangled representation learning framework is shown in Figure 3.

Data Augmentation & Simulation: Data augmentation and simulation techniques have become essential strategies in machine learning to enhance model performance, especially when the availability of labeled data is limited. Data augmentation involves generating modified versions of existing data samples through transformations such as rotation, flipping, scaling, cropping, or more sophisticated synthetic techniques like adversarial training [20]. These methods aim to increase dataset diversity without altering the underlying label semantics, thereby improving model robustness and generalization. In parallel, simulation approaches create synthetic datasets that mimic real-world processes, enabling the training and evaluation of models in controlled or otherwise inaccessible scenarios [21]. Techniques like domain randomization, procedural generation, and physics-based simulations have been widely employed in applications such as robotics, autonomous driving, and healthcare [22].

 doi:10.20944/preprints202511.0421.v1

Effective use of data augmentation and simulation not only mitigates overfitting but also supports the development of more reliable, generalizable, and fair machine learning systems.

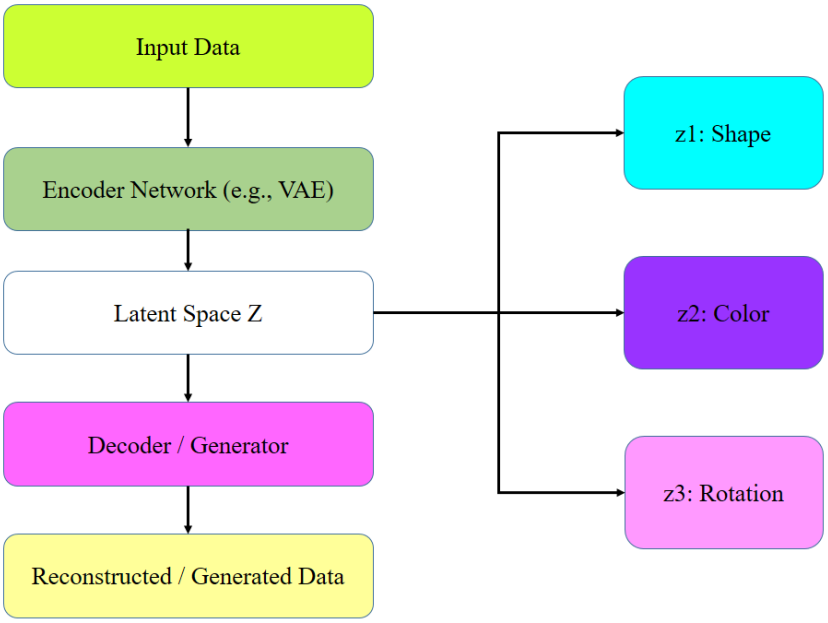

Figure 3. Illustration of latent variable flow in a disentangled representation learning framework.

Probabilistic Density Estimation: Probabilistic density estimation focuses on learning an accurate model of the underlying probability distribution $P(x)$ from which observed data are drawn. Mastering density estimation is fundamental for a range of tasks, including generative modeling, anomaly detection, and novelty generation. Classical approaches include techniques such as Kernel Density Estimation (KDE) [23] and Gaussian Mixture Models (GMMs) [24], which estimate the probability density function directly from data. Recent advances in deep learning have led to powerful neural density estimators, including Variational Autoencoders (VAEs) [4], Normalizing Flows [25], and autoregressive models like PixelCNN and PixelRNN [26]. Accurate density estimation enables the identification of out-of-distribution samples, supports uncertainty quantification, and forms the backbone of robust generative systems. In applications such as healthcare, finance, and security, where detecting anomalous patterns is crucial, probabilistic models provide the statistical rigor and flexibility necessary for reliable decision-making.

2.2. Core Generative Paradigms

Generative Adversarial Networks (GANs) [1] model data via an adversarial min-max game between a generator and a discriminator. The generator learns to map noise to data samples by optimizing a non-likelihood-based adversarial loss. Sampling is highly efficient, but mode collapse and unstable training are common challenges. GANs do not explicitly learn an interpretable latent space.

Variational Autoencoders (VAEs) [4] optimize a variational lower bound on the data likelihood, enforcing structured latent representations through regularization via Kullback-Leibler (KL) divergence. They allow easy sampling and interpolation but tend to generate blurrier outputs due to the reliance on approximate inference.

Normalizing Flows [25,27] directly model the data likelihood by learning invertible transformations between simple base distributions and complex data distributions. They offer exact likelihood evaluation and latent-variable inference but can be computationally expensive due to the constraint of invertibility and Jacobian computations.

Diffusion Models [2] learn to reverse a gradual noising process via denoising score matching. They produce high-quality samples but require many inference steps, leading to slower generation

 doi:10.20944/preprints202511.0421.v1

compared to GANs or VAEs. In standard diffusion models, the latent space is implicitly modeled through the denoising process. Recent work, such as Latent Diffusion Models (LDMs) [28], introduces explicit latent spaces to improve efficiency and scalability. Finally, Table 1 conveys comparison of core generative paradigms.

Table 1. Comparison of Core Generative Paradigms

Model Type	Objective Function	Sampling Efficiency	Latent Space	Notable Strengths	Common Challenges
GANs	Adversarial loss	Fast	Implicit	High-quality samples	Mode collapse, instability
VAEs	Variational lower bound (ELBO)	Fast	Structured, interpretable	Easy interpolation, latent utility	Blurry outputs due to inference
Normalizing Flows	Exact likelihood maximization	Moderate	Invertible, explicit	Precise likelihood, anomaly detection	High compute cost, Jacobian constraint
Diffusion Models	Denoising score matching	Slow	Implicit (LDM: explicit)	SOTA fidelity, stable training	Slow generation, compute-heavy

2.3. Evaluation Criteria from a Researcher's Perspective

To enable systematic comparison of generative models, it is essential to define a set of key technical axes:

- Sample Quality: Assesses the fidelity, coherence, and realism of generated outputs using metrics such as Fréchet Inception Distance (FID) [29], Inception Score (IS), and domain-specific evaluations.
- Sample Diversity: Measures the model's ability to capture a wide range of data modes and intra-class variations, indicating robustness against mode collapse.
- Controllability: Evaluates how effectively a model responds to conditioning inputs, supports fine-grained control, and generalizes to zero-shot scenarios.
- Computational Budget: Accounts for training and inference efficiency, including FLOPs, GPU memory usage, latency, and overall computational cost.
- Likelihood Estimation: Considers the availability, tractability, and accuracy of estimating the data likelihood $P(x)$, which is crucial for probabilistic interpretability [30].
- Latent Space Utility: Examines the structure and semantics of the latent space, focusing on interpretability, smoothness, and potential for disentangled representations.
- Training Dynamics: Reviews the stability and scalability of the training process, including convergence speed, sensitivity to hyperparameters, and adaptability to various scales. Table 2 provides information about Evaluation Criteria for Generative Models.

Table 2. Evaluation Criteria for Generative Models

Criterion	Focus	Metrics / Notes
Sample Quality	Realism and coherence of generated data	FID, IS, human evaluation
Sample Diversity	Range of variation and mode coverage	Mode count, intra-class spread
Controllability	Accuracy and granularity of conditional outputs	Prompt fidelity, editability, zero-shot ability
Computational Budget	Efficiency in training/inference	FLOPs, GPU hours, latency

 doi:10.20944/preprints202511.0421.v1

Table 2. *Cont.*

Criterion	Focus	Metrics / Notes
Likelihood Estimation	Ability to compute or estimate $P(x)$	Log-likelihood, bits/dim
Latent Space Utility	Smoothness and disentanglement of representations	Traversals, MIG, β-VAE score
Training Dynamics	Stability, convergence, scalability	Loss curves, hyperparameter robustness

3. Task-Driven Analysis

3.1. High-Fidelity Data Synthesis

Technical Goal & Challenges: High-fidelity data synthesis aims to generate synthetic data that is nearly indistinguishable from real-world data, achieving state-of-the-art (SOTA) perceptual quality. The primary technical goal is to replicate realistic textures, intricate shapes, precise lighting conditions, and dynamic interactions in a way that both human observers and machine perception systems perceive as authentic . A critical aspect involves capturing fine details, such as subtle micro-textures, hair strands, or skin pores, which are often the most challenging to synthesize without revealing artifacts [31]. Additionally, high-fidelity synthesis must handle the high dimensionality of complex data types—such as 3D environments, multimodal datasets, or spatiotemporal data—by accurately modeling their joint distributions while maintaining coherence and realism across all dimensions [32,33]. Another key goal is to ensure diversity and generalization, avoiding pitfalls like mode collapse that can limit the variety and richness of the generated samples However, achieving these goals comes with significant technical challenges. Even advanced generative models, including GANs and diffusion-based approaches, often struggle to maintain perceptual consistency, especially under close scrutiny, where minor artifacts or blurriness can betray the synthetic origin . The high dimensionality of target data spaces exacerbates the problem, requiring enormous computational resources and complex architectures to model large-scale structures while preserving fine-grained details [34]. Small-scale features are particularly difficult to reproduce, as models may lose fidelity during upscaling or introduce artifacts when attempting super-resolution enhancements [35]. Moreover, evaluating perceptual quality remains a challenge because existing metrics like FID [36], IS [37], or LPIPS [38] do not always align with human subjective judgment, complicating model optimization and benchmarking. Computational demands are another barrier, as training high-fidelity synthesis models typically involves intensive GPU or TPU usage, large datasets, and extended training times [39]. For sequential data such as video, ensuring both temporal coherence (to avoid flickering or inconsistencies) and spatial quality further amplifies complexity. Overall, high-fidelity data synthesis is a highly ambitious field that pushes the limits of current machine learning and computational capabilities. Finally, the overall working flow procedure has been presented in Figure 4.

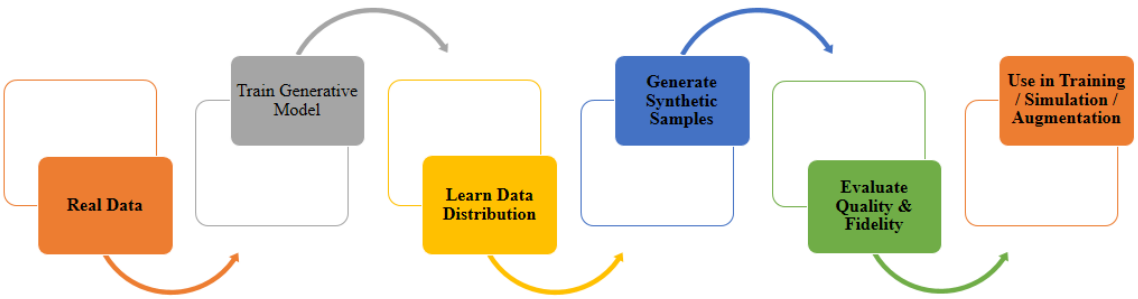

Figure 4. Working Flow of High-Fidelity Data Synthesis

 doi:10.20944/preprints202511.0421.v1

Leading Paradigms & Technical Rationale: High-fidelity data synthesis has become a central challenge in generative modeling, with the goal of generating data that closely mirrors real-world distributions while retaining complexity and variability. Two leading paradigms for achieving high-quality data generation are Diffusion Models and Generative Adversarial Networks (GANs). Diffusion models operate by gradually transforming random noise into structured data through an iterative denoising process. The model learns to reverse the diffusion process, which injects noise into a signal over time, and refines the data step-by-step. The denoising process is guided by score matching, where the model predicts the gradient of the data distribution at each step, effectively learning how to reverse the noise and generate high-fidelity outputs [40]. These models are valued for their stability during training, as they are less prone to issues such as mode collapse or vanishing gradients, which are often seen in GANs. The iterative refinement process results in sharper and more detailed samples, contributing to the generation of high-fidelity data [41].In contrast, GANs consist of two networks—the generator and the discriminator—which engage in an adversarial game. The generator's task is to produce realistic data samples, while the discriminator attempts to distinguish between real and generated data. The generator is trained to deceive the discriminator by receiving feedback on its outputs, which drives the continuous improvement of the generated samples [42]. A major innovation within GANs is the development of StyleGAN, which introduces a style-based generator that modulates image features at various levels, allowing fine control over the generation process. StyleGAN's use of adaptive normalization techniques, such as Adaptive Instance Normalization (AdaIN), enables better control over both the style and content of generated images, contributing to their high realism [43,44]. Additionally, the progressive growing technique, where training starts at low resolutions and progressively increases to higher resolutions, allows GANs to generate high-resolution images with remarkable realism.Key components like U-Nets, attention mechanisms, and adaptive normalization play crucial roles in enhancing data synthesis in both diffusion models and GANs. U-Nets, which employ an encoder-decoder architecture with skip connections, preserve both high-level semantic information and fine-grained details. In diffusion models, U-Nets help model the denoising process, while in GANs, they help refine generated images by merging coarse and detailed features [45]. Attention mechanisms, which enable models to focus on relevant parts of the data, are also crucial in improving the quality of generated samples, especially for tasks requiring a focus on specific image regions [46]. Finally, adaptive normalization techniques help stabilize training and improve sample quality by dynamically adjusting the generation process to match data characteristics [44]. Together, these innovations contribute to the high-fidelity generation of diverse data types, from images to audio, and make both diffusion models and GANs powerful tools for data synthesis.

Comparative Technical Analysis: High-fidelity data synthesis has seen significant advances through the use of both Diffusion Models and Generative Adversarial Networks (GANs), each with distinct trade-offs in terms of stability, diversity, speed, and sharpness. Diffusion models, which work by progressively refining noisy data back to its original form, exhibit a high degree of stability during training, making them less prone to issues such as mode collapse and posterior collapse that are commonly encountered in GANs. These models generate more diverse samples due to the iterative denoising process, where the data distribution is learned in a stepwise manner through score matching. However, the main drawback of diffusion models lies in their computational cost: they are typically slower both during training and inference. The process requires multiple steps to generate a single sample, resulting in a longer time to produce high-quality outputs, especially when compared to GANs, which generate a sample in a single forward pass [40].In contrast, GANs are renowned for their ability to generate high-sharpness outputs quickly. By employing an adversarial framework, GANs are capable of producing visually appealing, high-resolution data samples in a fraction of the time required by diffusion models. However, GANs suffer from issues such as mode collapse, where the generator produces a limited variety of outputs, and posterior collapse, where the generator fails to learn the full complexity of the data distribution, resulting in repetitive or low-quality samples [42]. While GANs excel in generating sharp images with high detail, these failure modes can lead to a loss of diversity

 doi:10.20944/preprints202511.0421.v1

in the generated samples, undermining the model's ability to capture the richness of real-world data distributions.The known failure modes of both models are crucial to understanding their limitations. In GANs, mode collapse occurs when the generator produces a narrow range of outputs that fool the discriminator but fail to represent the full diversity of the target distribution. This can be mitigated through techniques like feature matching and improved training strategies [42]. Posterior collapse, another challenge for GANs, happens when the generator's output becomes overly simplistic and fails to adapt to the discriminator's feedback, often leading to near-identical outputs. On the other hand, diffusion models, while more stable, can suffer from artifacts and blurriness during the denoising process. These artifacts typically arise when the model fails to perfectly reverse the noise process or when the training data is insufficient. Furthermore, because diffusion models rely on iterative refinement, they may smooth out fine details in the generated samples, leading to blurry outputs, especially when generating high-resolution images [40].In summary, the trade-off between diffusion models and GANs depends on the specific requirements of the application at hand. Diffusion models offer superior stability and diversity but at the cost of slower generation times. GANs, while faster and capable of generating sharper images, are prone to issues such as mode collapse and posterior collapse, which can limit the diversity and quality of the generated data. The selection between the two depends on whether speed and sharpness or stability and diversity are prioritized for the task.

Best Practices for Researchers: Selecting the appropriate generative model for high-fidelity data synthesis depends on key factors such as data modality, resolution, and computational resources. For image synthesis, GANs like StyleGAN [43] are ideal for high-resolution, sharp images, while diffusion models [40] offer better diversity at a higher computational cost. GANs are generally preferred for faster, high-quality generation, especially when computational resources are limited. However, diffusion models are gaining traction for generating more diverse outputs, albeit slower. For high-resolution tasks, StyleGAN and diffusion models are suitable, with the latter offering more stable and diverse results. Researchers with ample computational resources can leverage diffusion models, whereas those with fewer resources should consider GANs or their more efficient variants [47].

Implementation Details: In high-fidelity data synthesis, the model's performance is highly influenced by critical hyperparameters, optimizers, and stabilization techniques. For diffusion models, schedules for the noise variance, such as β schedules [40], play a crucial role in controlling the denoising process. Optimizing the learning rate and its schedule (e.g., cosine annealing or linear decay) ensures stable convergence. Similarly, in GANs, AdamW [48] is commonly used for its stability, with weight decay providing regularization. Loss weighting, particularly for balancing adversarial and auxiliary losses, is essential for training stability and quality. Data augmentation techniques like ADA (Augmented Data Augmentation) for GANs [49] and diffAug for diffusion models [40] help improve generalization by introducing variability in the data. These methods dynamically augment training data during the optimization process, increasing model robustness. Training stabilization is also critical. Techniques like gradient penalty [50] and spectral normalization [51] prevent mode collapse and improve the stability of the training process. Gradient penalty regularizes the discriminator to avoid overly sharp gradients, while spectral normalization controls the Lipschitz constant of the model, mitigating issues with unstable training dynamics.

Evaluation Protocol: In high-fidelity data synthesis, a rigorous evaluation protocol is essential for assessing the quality of generated data. Commonly used metrics include the Fréchet Inception Distance (FID) [29], Kernel Inception Distance (KID) [52], and Inception Score (IS) [53]. FID and KID measure the similarity between real and generated distributions in feature space, with FID being more sensitive to mode collapse. IS evaluates the quality of generated images based on the classification performance of an Inception network. However, these metrics have limitations, such as their reliance on pretrained networks, which may not capture domain-specific nuances.Perceptual studies, including human evaluation, provide valuable insights into the subjective quality of generated data. In domains like medical imaging or 3D object synthesis, domain-specific metrics (e.g., Dice score for segmentation tasks or Intersection over Union for object detection) should be employed for a more accurate assessment.

 doi:10.20944/preprints202511.0421.v1

A common pitfall in metric usage is over-relying on FID or IS without considering domain-specific factors, which can lead to misleading conclusions about model performance.

Key Codebases/Libraries: In high-fidelity data synthesis, leveraging well-established open-source codebases is critical for efficient model development and experimentation. For diffusion models, the `diffusers` library by Hugging Face [40] is a comprehensive toolkit that provides pre-trained models and utilities for efficient implementation, including support for Denoising Diffusion Probabilistic Models (DDPMs) and related architectures.For GAN-based image generation, the StyleGAN repositories are influential. `StyleGAN2` [54] and `StyleGAN3` [55] provide highly optimized implementations for state-of-the-art image generation, supporting features like adaptive normalization and progressive growing. These libraries offer flexible APIs, extensive documentation, and pretrained models, making them ideal for high-resolution image synthesis.In addition, the `TensorFlow-GAN` and `pytorch-GAN` repositories offer modular and scalable implementations for various GAN architectures, including DCGAN and WGAN, useful for both beginners and experts in GAN research.

3.2. Controllable & Conditional Generation

Technical Goal & Challenges: The technical goal of controllable and conditional generation is to produce coherent, fluent, and semantically meaningful outputs that respect explicit user-defined constraints. These constraints may be linguistic (e.g., sentiment, topic), structural (e.g., syntactic form, layout), stylistic (e.g., formality, author imitation), or categorical (e.g., class label in image or text generation). A core challenge lies in ensuring *strong adherence to diverse conditions* without compromising on generation quality or diversity. Traditional autoregressive language models and generative models like VAEs and GANs often struggle to balance this trade-off between *control* and *quality/diversity* [56–58].When constraints become too rigid or narrowly defined, models tend to overfit the conditioning signals, leading to degeneration in fluency or richness of content. Conversely, overly flexible models risk failing to adhere to the specified conditions. This control–quality tension is particularly evident when enforcing multiple simultaneous constraints, such as requiring output that is both emotionally positive and syntactically complex [59]. Further complicating the challenge are issues such as exposure bias, mode collapse, and representation entanglement, which may hinder disentangled learning of controllable factors. Recent approaches have explored solutions like prefix tuning, plug-and-play models, disentangled latent variable models, and variational learning to better balance these demands [60–62]. Yet, designing robust and scalable conditional generation systems remains an open and actively researched area in natural language processing and generative modeling.

Leading Paradigms & Technical Rationale: The field of controllable and conditional generation is underpinned by several major generative paradigms, each with distinct mechanisms for incorporating conditioning signals. In diffusion models, conditioning is typically implemented using techniques like classifier-free guidance (CFG), which interpolates between unconditional and conditional predictions to modulate adherence strength [63]. Cross-attention mechanisms are also employed, where conditioning embeddings (e.g., text) are injected into the denoising process, guiding each diffusion step [28]. Additional methods such as embedding injection or learned encoders translate condition signals into a form directly usable during generation. Generative Adversarial Networks (GANs) employ conditional extensions like conditional GANs (cGANs), where both generator and discriminator receive the condition as an input, often concatenated or projected into a shared latent space [64]. Projection discriminators further refine conditioning by computing inner products between condition embeddings and discriminator features, improving alignment [65]. Recent advances like CLIP-guided GANs use pretrained contrastive vision-language models to align generations with natural language conditions in a semantically meaningful way, offering zero-shot controllability and robust semantic supervision [66].

In Variational Autoencoders (VAEs), conditioning is integrated either via a conditional prior over latent variables or by injecting the condition into the decoder, allowing the model to learn condition-specific latent distributions [67]. These methods provide probabilistic control and support latent disentanglement. Transformers, especially in language models, support control through prompting,

 doi:10.20944/preprints202511.0421.v1

prefix-tuning, and fine-tuning. Prompting uses textual cues to steer generation with minimal changes to the model parameters, while prefix-tuning learns a small number of continuous vectors prepended to each layer's attention input [68]. Full fine-tuning remains effective for strong conditioning, though at the cost of computational efficiency. These approaches collectively enable scalable and flexible controllability across diverse modalities and applications.

Comparative Technical Analysis: A comparative technical analysis of controllable and conditional generation methods reveals trade-offs across several key dimensions. One prominent axis is *flexibility versus faithfulness*. Techniques such as prompt-based or zero-shot conditioning (e.g., CLIP-guided generation or language model prompting) offer high flexibility and low overhead, enabling rapid adaptation to unseen conditions [69,70]. However, they often exhibit weaker faithfulness to the conditioning signals, especially under complex or multi-attribute constraints [60]. In contrast, fine-tuned models—such as conditional VAEs or fully supervised cGANs—demonstrate stronger adherence to conditioning but at the cost of reduced generalization and flexibility [64,67].

Another dimension involves *zero-shot vs. fine-tuned control*. Zero-shot approaches (e.g., CLIP-guided generation, prompt engineering) avoid expensive retraining but often rely on pretrained models that are susceptible to misalignment and attribute entanglement. Fine-tuned control, while more accurate, requires extensive annotated data and increases computational complexity.

Computational overhead is also a significant concern. Methods like prefix-tuning or adapter-based tuning reduce the overhead compared to full fine-tuning, enabling more efficient deployment [68,71]. Diffusion-based models, though offering high-fidelity control, are generally slower during inference due to iterative denoising steps [28,63].

Moreover, many control techniques are vulnerable to *attribute leakage and bias amplification*, particularly when training data contain social or demographic imbalances. For instance, models trained to generate sentiment- or gender-conditioned text may unintentionally reinforce stereotypes or leak private attributes [72,73]. Addressing these risks requires not only technical safeguards (e.g., adversarial debiasing, disentanglement) but also careful evaluation frameworks.

Overall, no single method dominates across all metrics, and the choice of approach depends on the target application's needs for controllability, generalization, efficiency, and ethical robustness.

Best Practices for Researchers: Cross-attention is widely used for aligning conditioning signals like text in transformers [28]. AdaIN, AdaGN, and FiLM layers provide effective modulation for style and class control via normalization-based conditioning [44,74,75]. Classifier guidance in diffusion models adjusts generation using gradients from external classifiers. Prompt engineering is simple but less reliable than architectural methods like prefix-tuning and adapters, which offer stronger and more consistent control [68,71].

Data Strategies: Effective control requires aligning data with conditioning modalities. Paired datasets (e.g., image-caption) enable supervised learning but are often costly to obtain. Unpaired setups can be mitigated using powerful multimodal embeddings like CLIP or ALIGN to align modalities in a shared space [69,76]. Synthesizing pseudo-paired data through retrieval, augmentation, or back-translation is a practical strategy to scale supervision [77,78]. Researchers should balance data quality, diversity, and annotation cost when designing training pipelines.

Training & Adaptation: Instruction tuning aligns models with human intent using diverse, task-oriented prompts [79]. Reinforcement learning with human or AI feedback (RLHF/RLAIF) further refines outputs to maximize helpfulness and safety [80]. Parameter-efficient fine-tuning (PEFT) methods like LoRA [81], Adapters [71], and Prompt Tuning [82] reduce compute costs while enabling effective control adaptation in large models. Combining PEFT with instruction-tuned checkpoints often yields strong performance with minimal resources.

Evaluation: Reliable evaluation of controllable generation requires both automated and human-centered metrics. CLIP score and attribute classifiers are widely used to assess conditional consistency between inputs and outputs [83,84]. Disentanglement scores help verify that individual controls influence only their target attributes [85]. Human evaluations remain essential, especially for subjective

 doi:10.20944/preprints202511.0421.v1

aspects like style, intent alignment, and control faithfulness, typically using Likert scales or preference rankings [86]. A balanced protocol should combine multiple metrics for robust assessment.

3.3. Sequential & Structured Data Modeling:

Technical Goal & Challenges: The modeling of sequential and structured data aims to understand and predict patterns within data that follow a specific order or structure, such as time series data, text, and other domain-specific sequential data. A key technical goal in this area is to effectively capture dependencies between observations across time or other relational structures. Sequential data, often encountered in domains like speech recognition, natural language processing (NLP), and financial forecasting, typically presents temporal dependencies that require models capable of handling time-ordered data. Structured data, such as graphs and tables, involves capturing relationships between entities and the inherent structural dependencies, which can include spatial or hierarchical relationships. The primary technical objective is to build models that can represent these dependencies, enabling accurate predictions or classifications.

A major challenge in sequential and structured data modeling lies in the need to preserve long-range dependencies while minimizing the complexity of the model. Traditional approaches such as feedforward neural networks (FNNs) struggle with capturing long-term dependencies due to vanishing or exploding gradient problems [87]. Recurrent neural networks (RNNs), including their advanced versions such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), offer a solution by allowing the model to maintain memory over time, but they still face challenges in scaling to long sequences [88]. Moreover, structured data such as graphs requires specialized models, such as graph neural networks (GNNs), that can effectively process data with non-Euclidean structures and learn node and edge dependencies [89]. These models, however, struggle with computational complexity and the need for large amounts of labeled data to achieve generalization.

Another challenge lies in handling noisy or incomplete data, which is common in real-world applications. Sequential and structured data often include missing values, irrelevant features, or anomalies, which can significantly affect model performance. Techniques such as imputation and regularization are often employed to mitigate these issues, but they introduce trade-offs in terms of accuracy and computational overhead [90]. Furthermore, achieving scalability while maintaining interpretability is a significant hurdle, particularly in structured data settings, where complex dependencies must be modeled at scale. In summary, the technical goals of sequential and structured data modeling involve capturing the temporal and relational dependencies inherent in the data, while overcoming challenges related to long-range dependencies, noisy data, computational complexity, and interpretability.

Leading Paradigms & Technical Rationale: Sequential and structured data modeling involves the use of specialized paradigms and techniques that are designed to capture the dependencies and relationships inherent in such data. One of the leading paradigms in sequential data modeling is the use of Recurrent Neural Networks (RNNs), which are designed to process sequences of data by maintaining a hidden state that is updated at each time step. RNNs are well-suited for tasks such as speech recognition, machine translation, and time series prediction, where the output at each time step is dependent on previous observations [88]. However, standard RNNs suffer from the vanishing gradient problem, which limits their ability to model long-range dependencies in sequences. This challenge led to the development of Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which incorporate mechanisms for selectively remembering or forgetting information over time, allowing them to handle long-term dependencies more effectively [88].

In parallel, the advent of Transformer-based models has significantly advanced sequential data modeling. The Transformer architecture, introduced by Vaswani et al. [46], replaces the recurrence mechanism with self-attention, allowing the model to weigh the importance of each input in a sequence, irrespective of its position. This paradigm has revolutionized NLP tasks, leading to the development of models such as BERT, GPT, and T5, which excel in tasks ranging from sentiment analysis to machine translation [91,92]. Transformers overcome the limitations of RNNs by processing entire sequences

 doi:10.20944/preprints202511.0421.v1

in parallel and capturing long-range dependencies more effectively, thus reducing training time and improving performance.

For structured data, such as relational data in databases or graph-structured data, Graph Neural Networks (GNNs) have emerged as a leading paradigm. GNNs extend deep learning techniques to graph data by leveraging the relationships between nodes and edges in a graph structure [89]. This approach has proven effective for tasks like node classification, link prediction, and graph-based recommendation systems. The key technical rationale behind GNNs lies in their ability to propagate information across a network structure, allowing them to capture complex interdependencies that are often missed by traditional models. Another powerful approach for structured data modeling is the use of decision tree-based ensemble methods such as Random Forests and Gradient Boosting Machines (GBM). These models partition the feature space based on splits that maximize information gain, and their ensemble nature enables them to model complex, non-linear relationships in structured data [93,94].

The technical rationale behind these paradigms is rooted in the need to address the inherent dependencies and complexities present in sequential and structured data. In sequential data, the challenge is to model the temporal relationships and capture long-term dependencies, which is crucial for tasks like forecasting or sequential decision-making. For structured data, the goal is to learn representations that respect the underlying relational structure of the data, whether it is graph-based, tabular, or hierarchical. These paradigms have made significant strides in recent years by introducing novel mechanisms for dependency modeling, parallel computation, and generalization, all of which contribute to their effectiveness in real-world applications.

Comparative Technical Analysis: The technical analysis of sequential and structured data modeling highlights key differences in the challenges they address, the methodologies employed, and their relative advantages in various domains. Sequential data, such as time series or text data, is characterized by its temporal or ordered nature, where the order of observations matters. Modeling sequential data effectively requires techniques that can capture dependencies over time or across sequence elements. Traditional methods such as Hidden Markov Models (HMMs) and autoregressive models (e.g., ARIMA) have been widely used for time series prediction, but their inability to model long-range dependencies and their reliance on handcrafted features limit their effectiveness in more complex tasks [95]. In contrast, deep learning techniques, particularly Recurrent Neural Networks (RNNs) and their advanced variants like Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have been designed to overcome these limitations by maintaining hidden states that store information from previous time steps, thus capturing longer dependencies [88]. More recently, Transformer-based models have surpassed RNNs and LSTMs in several domains by employing self-attention mechanisms that enable the model to capture dependencies across the entire sequence simultaneously, thus improving both performance and training efficiency [46].

In structured data modeling, which deals with data that has a predefined format or schema (such as tabular data, graphs, or hierarchical data), the goal is to learn the complex relationships between features, entities, or structures. Models like decision trees, Random Forests, and Gradient Boosting Machines (GBMs) excel in tabular data analysis by partitioning the feature space and creating models that are interpretable, robust to overfitting, and capable of handling heterogeneous data [93,94]. On the other hand, Graph Neural Networks (GNNs) have become the go-to approach for modeling relational data, particularly in scenarios where the data is represented as a graph of nodes and edges [89]. GNNs operate by aggregating information from neighboring nodes, which allows them to capture the dependencies and relationships inherent in the graph structure, such as in social networks or molecular biology.

When comparing the two paradigms, one of the most striking differences is the handling of dependencies. Sequential models, particularly RNNs and Transformers, excel in capturing temporal dependencies and sequences where the order is paramount, making them ideal for time series forecasting, natural language processing (NLP), and speech recognition. However, they struggle with

 doi:10.20944/preprints202511.0421.v1

non-sequential, structured relationships, such as those found in graphs or heterogeneous tabular data. Conversely, structured data models like decision trees and GNNs are designed to handle complex, non-temporal relationships effectively, but they lack the ability to handle sequential dependencies. Another critical difference is in computational efficiency. Transformer models, while highly effective, tend to require significant computational resources due to their attention mechanisms, especially for long sequences [91]. In contrast, traditional models for structured data, such as decision trees and GNNs, can be more computationally efficient and scalable to larger datasets.

In summary, the choice of model depends largely on the nature of the data and the specific problem at hand. Sequential data modeling benefits from techniques like RNNs, LSTMs, GRUs, and Transformers that can capture temporal dependencies, while structured data modeling leverages methods like decision trees, Random Forests, GBMs, and GNNs that excel at understanding complex relationships in structured datasets. The key to successful modeling is selecting the appropriate paradigm based on the data type and the goal of the analysis.

Best Practices for Researchers: When conducting research in the field of sequential and structured data modeling, it is essential to follow best practices that not only enhance the model's performance but also ensure reliability, interpretability, and generalizability of results. Below are some best practices that researchers should adopt:

Data Preprocessing and Feature Engineering: One of the most critical steps in both sequential and structured data modeling is proper data preprocessing. For sequential data, this may include normalization, noise reduction, and handling missing values. For structured data, feature selection and extraction play a key role in improving model accuracy and preventing overfitting [96]. In sequential data tasks, it is important to ensure that the data is ordered correctly, and any temporal dependencies are preserved. Feature scaling, encoding categorical variables, and handling outliers should be done carefully to avoid introducing biases into the model. Additionally, researchers should explore domain-specific feature engineering techniques that may offer better insights into the data, which can dramatically improve performance in both sequential and structured modeling tasks.

Model Selection and Evaluation: Researchers should adopt an experimental mindset when selecting models for sequential and structured data. For sequential data, techniques like RNNs, LSTMs, GRUs, and Transformer-based models should be considered based on the nature of the dependencies present in the data [46,88]. Structured data models, on the other hand, benefit from decision tree-based methods, ensemble techniques like Random Forests and Gradient Boosting Machines (GBMs), and Graph Neural Networks (GNNs) for relational data [89,93]. Researchers should evaluate models using proper cross-validation methods, such as k-fold or time-series cross-validation, to ensure that the models generalize well to unseen data. The selection of performance metrics, such as accuracy, precision, recall, F1-score for classification, and RMSE for regression tasks, should align with the specific objectives of the research.

Hyperparameter Tuning and Regularization: Fine-tuning hyperparameters is essential for optimizing model performance. Sequential data models, such as LSTMs and Transformers, often require tuning of parameters like learning rate, batch size, and the number of layers to prevent underfitting or overfitting. Structured data models, including decision trees and gradient boosting methods, also require tuning of tree depth, learning rate, and regularization parameters to ensure optimal performance without overfitting [97]. Regularization techniques like L2 (Ridge) and L1 (Lasso) regularization are crucial for both types of models to prevent overfitting, especially when working with high-dimensional or noisy data [90]. Researchers should utilize grid search or randomized search methods to explore the hyperparameter space and determine the optimal configuration.

Handling Missing or Noisy Data: In both sequential and structured data, missing or noisy data can significantly hinder model performance. For sequential data, strategies like forward/backward filling, interpolation, or imputation techniques such as KNN imputation are common ways to handle missing values [98]. In structured data, missing data can be handled through various imputation techniques, such as using the mean, median, or mode for numerical values, or employing more sophisticated

 doi:10.20944/preprints202511.0421.v1

methods like multiple imputation or model-based imputation. Noise reduction techniques, such as smoothing and filtering, can be applied to both sequential and structured datasets to ensure that the models are trained on clean, reliable data [99].

Model Interpretability and Explainability: Especially in real-world applications, interpretability and explainability of models are critical. For sequential data models, techniques such as attention mechanisms in Transformer models or visualizing activations in LSTMs can help researchers understand the inner workings of their models and the temporal dependencies they learn [46]. For structured data, methods like feature importance analysis in decision trees and SHAP (Shapley Additive Explanations) values can provide insights into which features are most influential in predictions [100]. Researchers should prioritize methods that offer model transparency, particularly in domains like healthcare and finance, where understanding model decisions is essential for user trust and regulatory compliance.

3.4. Representation Learning & Disentanglement

Technical Goal & Challenges: Representation learning lies at the core of generative artificial intelligence (GenAI), aiming to extract compact, meaningful features from high-dimensional data such as images, text, and audio. A central technical objective is to learn latent representations that are not only compact and expressive but also disentangled, where each dimension corresponds to a distinct generative factor of variation [101]. Disentangled representations provide improved interpretability, allow for fine-grained control over generated outputs, and facilitate transfer across tasks or domains [101,102]. For instance, in image generation, a disentangled model could independently modify facial expressions, lighting conditions, or hair color without altering other attributes, thus enabling more controllable and reliable synthesis [102,103].

However, achieving disentanglement presents significant challenges. In the unsupervised setting, disentanglement is known to be fundamentally ill-posed without strong inductive biases or explicit supervision [104]. This ambiguity arises because multiple factorizations of the data can be equally valid explanations. Moreover, enforcing disentanglement often introduces trade-offs with generation quality; overly constrained latent spaces can reduce output fidelity or diversity [105]. Evaluating disentanglement is also difficult in real-world scenarios due to the lack of ground-truth generative factors. In addition, complex correlations between underlying factors in real-world data—such as the entanglement between age and wrinkles in facial datasets—further hinder clean separation in the latent space [103]. Challenges like latent variable collapse, mode dropping in adversarial training, and scalability to multi-modal or high-dimensional data continue to be active areas of research [106].

In summary, while disentangled representation learning holds promise for more interpretable, controllable, and generalizable GenAI systems, addressing its theoretical and empirical limitations remains crucial for its broader applicability.

Leading Paradigms & Technical Rationale: In the domain of generative AI, leading paradigms for representation learning and disentanglement are grounded in deep probabilistic models that encode high-dimensional data into structured latent spaces. Variational Autoencoders (VAEs) form a foundational approach, where the encoder-decoder framework learns to approximate the data distribution while regularizing the latent space via the Evidence Lower Bound (ELBO) objective [107]. Disentanglement in VAEs is commonly achieved by introducing modifications such as the β-VAE, which scales the Kullback-Leibler divergence term to enforce a trade-off between reconstruction fidelity and latent factor independence [102]. FactorVAE and β-TCVAE further refine this approach by explicitly penalizing total correlation, which measures statistical dependence between latent variables [108,109].

Another key paradigm involves adversarial learning frameworks, such as InfoGAN, which extend Generative Adversarial Networks (GANs) by maximizing the mutual information between a subset of latent codes and the generated samples, thereby promoting interpretability and disentanglement [110]. More recently, contrastive learning techniques have gained traction for learning useful representations by maximizing agreement between positive sample pairs while distinguishing negatives, even in the absence of labels [111]. These methods offer competitive performance on downstream tasks but often lack explicit disentanglement.

 doi:10.20944/preprints202511.0421.v1

The technical rationale behind these paradigms stems from the need to balance competing objectives: expressiveness, interpretability, and generalization. While over-regularization may lead to poor reconstruction or latent collapse, under-regularization risks learning entangled representations that lack semantic structure. This motivates research into principled decompositions of the ELBO, disentanglement-specific regularizers, and hybrid frameworks combining probabilistic inference with inductive priors. Despite progress, current methods still struggle to scale effectively to complex, high-dimensional data and to align learned factors with human-interpretable semantics without supervision [104].

In summary, the evolution of representation learning in GenAI has been shaped by probabilistic inference, adversarial training, and contrastive objectives. Each paradigm contributes uniquely to the goal of learning structured, disentangled latent representations, though achieving robust, unsupervised disentanglement across diverse domains remains a fundamental challenge.

Comparative Technical Analysis: Representation learning in Generative AI has evolved through several major paradigms, each with distinct technical trade-offs and suitability for disentanglement. Variational Autoencoders (VAEs) rely on probabilistic inference to model the data distribution via an encoder-decoder architecture, providing a principled approach for latent variable modeling [107]. Extensions such as β-VAE [102], FactorVAE [108], and β-TCVAE [109] introduce regularization techniques to promote disentangled representations. These methods offer strong theoretical grounding and stability in training but often suffer from blurry outputs and a trade-off between disentanglement and reconstruction quality [105].

Generative Adversarial Networks (GANs), in contrast, produce high-quality, sharp images by optimizing a two-player minimax game between a generator and a discriminator [42]. InfoGAN extends this framework by incorporating latent codes that maximize mutual information with generated outputs, enabling partial disentanglement [110]. However, GAN-based approaches generally lack a true inference mechanism and struggle with training instability, mode collapse, and less structured latent spaces compared to VAEs.

Recently, contrastive learning has emerged as a powerful alternative for unsupervised representation learning. Methods like SimCLR [111] and MoCo [112] focus on instance-level discrimination by contrasting positive and negative sample pairs, enabling the learning of semantically meaningful embeddings. Although not explicitly designed for disentanglement, contrastive methods can capture transferable and robust features across modalities. However, they typically require large batch sizes, careful augmentation strategies, and do not provide generative capabilities without additional architectures.

Comparatively, VAEs offer better control and interpretability, GANs provide superior sample quality, and contrastive learning delivers strong representations for downstream tasks but weak disentanglement. No paradigm yet achieves all desired objectives simultaneously, underscoring the need for hybrid models that integrate generative, inferential, and contrastive principles to fully realize the goals of disentangled representation learning in GenAI.

Best Practices for Researchers:

Align Model Objectives with Application Needs: Select representation learning paradigms (e.g., VAEs, GANs, contrastive models) based on whether the goal is generation quality, factor disentanglement, or downstream utility [101,111].

Choose Inductive Biases Thoughtfully: Incorporate architectural constraints, prior factorization, or compositionality assumptions to guide disentanglement, especially in unsupervised settings [102,104].

Use Strong Baselines and Controlled Comparisons: Evaluate novel models against standardized baselines such as β-VAE [102], FactorVAE [108], or InfoGAN [110], under consistent evaluation conditions.

Incorporate Disentanglement Metrics Cautiously: Utilize multiple metrics—e.g., Mutual Information Gap (MIG), DCI Disentanglement, SAP score—to assess disentanglement, acknowledging that no single metric is comprehensive [85].

 doi:10.20944/preprints202511.0421.v1

Leverage Supervision When Feasible: Use weak or semi-supervised approaches to partially guide representation learning when pure unsupervised methods are insufficient [113].

Avoid Over-Regularization: Excessive latent regularization (e.g., overly large β in β-VAE) can lead to latent variable collapse or degraded reconstructions [105].

Validate Representations on Downstream Tasks: Evaluate latent codes in terms of their transferability and utility on supervised classification, reinforcement learning, or image manipulation [104].

Use Visualization and Latent Traversals: Qualitative tools like latent space interpolations and single-dimension traversals remain essential for human evaluation of disentanglement [102].

Benchmark on Realistic Datasets: Avoid relying solely on simplistic synthetic datasets (e.g., dSprites) and validate generalization on complex, high-dimensional, or real-world data [104].

Document Experimental Settings Thoroughly: Ensure reproducibility by clearly reporting architecture, hyperparameters, datasets, training schedules, and evaluation code [114].

3.5. Data Augmentation & Simulation

Technical Goal & Challenges: Data augmentation and simulation in Generative AI aim to improve model robustness, generalization, and performance in low-data regimes by synthetically expanding the diversity of training samples. A key technical goal is to generate high-fidelity, semantically consistent variations that capture real-world distributional nuances while preserving critical attributes for downstream tasks [115]. Simulation-driven augmentation further seeks to create photorealistic, physics-consistent, or domain-transferrable synthetic data, especially useful in safety-critical domains like autonomous driving or healthcare [116,117]. However, a core challenge lies in maintaining distributional alignment—models often struggle with the "reality gap," where synthetic and real data differ in subtle but significant ways, impairing generalization [118]. Additionally, designing augmentations that are task-relevant yet label-preserving is non-trivial and often domain-specific, requiring expert knowledge or adaptive augmentation policies [119].

Table 3. Comparative Technical Analysis: Data Augmentation vs. Simulation in Generative AI

Aspect	Data Augmentation	Simulation
Core Idea	Modify existing real data using transformations	Generate entirely synthetic data mimicking real-world scenarios
Use Case	When real data exists but needs variation	When real data is scarce, expensive, or risky to collect
Examples	Image flipping, rotation, scaling; text paraphrasing	Virtual environments for driving, healthcare, robotics
Efficiency	✓ Computationally efficient	✗ Computationally intensive
Control & Diversity	✗ Limited to realistic transformations	✓ High control, includes rare scenarios
Realism	✓ Operates on real data	✗ Depends on simulation fidelity
Scalability	✓ Easily scalable with low cost	✗ Requires high resources and effort
Generalization Risk	✓ Low risk if transformations are valid	✗ Risk if synthetic data is not representative
Training Domain	Image classification, NLP, speech recognition	Robotics, autonomous systems, healthcare, RL
Main Limitation	Limited by the variability of original data	May introduce unrealistic or biased data

Simulation frameworks may also introduce biases or artifacts due to incomplete modeling of environmental dynamics or sensor noise, leading to spurious correlations [120]. Moreover, integrating generative augmentation into training pipelines at scale poses computational and architectural challenges, including data efficiency, augmentation diversity control, and dynamic policy learning [121].

 doi:10.20944/preprints202511.0421.v1

Addressing these issues is critical for building robust, generalizable, and sample-efficient GenAI systems that can perform reliably in open-world conditions.

Leading Paradigms & Technical Rationale: In the rapidly evolving field of Generative AI (GenAI), data augmentation and simulation have emerged as key strategies for addressing data limitations and improving model performance. These methods play a vital role in enhancing the generalization capabilities of AI models, particularly when faced with sparse, imbalanced, or biased datasets. Data augmentation techniques are centered on artificially expanding the training dataset by introducing small transformations, such as rotations, translations, or color adjustments, to the original data [42]. In contrast, simulation involves generating entirely synthetic datasets that mimic the characteristics of real-world data, often through computational models or virtual environments. These synthetic datasets allow for the scaling of training data without the need for costly manual data collection or reliance on real-world data that may be difficult to obtain.

Data augmentation methods have become a cornerstone in domains such as image recognition, natural language processing (NLP), and speech recognition. For instance, in computer vision, common augmentations such as flipping, cropping, or adding noise can significantly improve the robustness of models against variations in input data [115]. In NLP, techniques like paraphrasing or back-translation are widely used to create diverse linguistic variations of the same underlying content [122]. These augmentations help prevent overfitting, as the model is trained on a more diverse set of examples, thereby improving its ability to generalize to unseen data.

On the other hand, simulation techniques often use complex models or virtual environments to create synthetic data. In reinforcement learning (RL), for instance, agents are trained in simulated environments to learn optimal policies before being deployed in real-world scenarios [123]. Similarly, in autonomous driving, simulation environments generate vast amounts of data on driving scenarios, enabling the training of self-driving car models without the risk of accidents or the need for real-world driving data [124]. Such approaches are not only cost-effective but also allow for the creation of data under controlled conditions that may be difficult or impossible to capture in reality.

The rationale behind both data augmentation and simulation lies in their ability to improve model robustness and performance. By increasing the variety of data the model encounters, these techniques help prevent overfitting to the limited samples available and make the model more resilient to real-world variations. Furthermore, simulation allows for the generation of rare or edge-case scenarios, enabling the model to learn how to handle situations that are underrepresented in the available real-world data. As AI models continue to grow in complexity and application, these paradigms are expected to remain essential in building more powerful and generalized systems.

Comparative Technical Analysis:

Data augmentation and simulation are two central techniques in the field of Generative AI (GenAI), both aiming to improve model performance by expanding the available data for training. While both methods share the goal of enhancing the generalization of models, they differ significantly in their execution, benefits, and use cases.

Data augmentation involves generating new training data by applying transformations to existing data. This approach includes operations like flipping, rotating, scaling, cropping images, or paraphrasing and back-translating text in natural language processing (NLP) tasks. The main advantage of data augmentation is its computational efficiency; it allows for the expansion of the training dataset without the need to acquire new data. This technique is particularly useful in domains like image classification and NLP, where small modifications can provide the model with a wider range of input variations. Moreover, because data augmentation operates directly on the available real data, it is generally less resource-intensive compared to simulation. However, its effectiveness is limited by the transformations that can meaningfully alter the data without changing its core characteristics. For example, flipping an image might work well for object recognition tasks, but this approach is less effective for time-series data, where the sequential order is critical.

 doi:10.20944/preprints202511.0421.v1

Simulation, in contrast, involves generating entirely synthetic datasets that mimic real-world data. This approach is commonly used when acquiring real-world data is difficult, expensive, or impractical. In applications like autonomous driving, simulation can create virtual environments where agents can be trained without the need for real-world data or the risks associated with physical testing. Simulation allows for the creation of highly controlled, diverse, and rare scenarios that might not exist in the available real-world data. For instance, in reinforcement learning, synthetic environments enable agents to interact and learn from a variety of conditions, including rare edge cases that are crucial for ensuring robust model performance. The primary strength of simulation lies in its ability to generate large volumes of data in a controlled environment, without the logistical or ethical concerns of real-world data collection. However, simulation often requires significant computational resources to develop and maintain realistic virtual environments. Furthermore, the quality of the synthetic data depends heavily on the accuracy of the simulation model. If the simulation does not fully capture the complexities of the real world, the model may struggle to generalize when applied to real-world scenarios.

When comparing data augmentation and simulation, the key distinction lies in the balance between computational efficiency and data diversity. Data augmentation is ideal when there is sufficient real-world data available, and the goal is to expand this data by introducing slight variations. It is well-suited for applications like image recognition, speech recognition, and NLP, where simple transformations can improve the model's ability to generalize. On the other hand, simulation is more appropriate in scenarios where real-world data is scarce or difficult to obtain, such as in robotics, autonomous systems, and certain healthcare applications. Simulation can create a broader range of data, including rare or dangerous scenarios that would be hard to capture through real-world data collection.

Both techniques have their limitations, with data augmentation being constrained by the types of transformations that can be applied and the inherent diversity of the original dataset. Simulation, while more flexible and capable of generating entirely new data, often involves high computational costs and may not always reflect real-world conditions accurately. In some cases, the data generated through simulation can introduce biases or unrealistic scenarios, leading to overfitting to synthetic data.

In conclusion, data augmentation and simulation both play crucial roles in the development of GenAI systems. Data augmentation is a powerful, efficient tool when there is ample real-world data, providing a cost-effective way to diversify the training set. Simulation, though more resource-intensive, offers the ability to generate synthetic data when real-world data is inaccessible or insufficient. A careful understanding of the problem at hand and the availability of resources will help determine which approach is more suitable for a given application. Here Table 3 represents the Comparative Technical Analysis: Data Augmentation vs. Simulation in Generative AI.

Best Practices for Researchers: Prior to applying augmentation or simulation techniques, thoroughly understand the characteristics of the data. Different data types (e.g., images, text, time-series) may require distinct augmentation techniques. For instance, rotating an image may be useful, but flipping a time-series dataset could lead to unintended consequences [115].

Use diverse augmentation techniques to ensure that the model is exposed to a wide variety of transformations. This may include spatial transformations (e.g., rotation, scaling, cropping), color variations (e.g., brightness, contrast adjustments), and, for NLP tasks, syntactic variations such as paraphrasing or back-translation [125].

Augmentation techniques should preserve the underlying structure and labels of the data. For example, when augmenting images, make sure the transformations do not distort the object in a way that changes its label or classification [126].

Augmentations should be applied in moderation. Too many augmentations or overly aggressive transformations can lead to the model overfitting to artificial data patterns. Carefully monitor validation performance to avoid this risk [127].

 doi:10.20944/preprints202511.0421.v1

For simulation, ensure that the synthetic data reflects the complexities and nuances of the real-world application. In domains like autonomous driving or healthcare, the simulation must incorporate realistic environmental conditions, edge cases, and rare events that are essential for model robustness [123].

When using simulation-generated data, it's critical to continuously validate its quality by comparing it with real-world data. The closer the simulation is to reality, the better the model will generalize to actual deployments [128].

In many cases, a hybrid approach using both real-world and synthetic data can be beneficial. This allows models to learn from authentic data while also gaining exposure to rare or difficult-to-capture scenarios created via simulation or augmentation [42].

Both data augmentation and simulation can unintentionally introduce biases, especially when the transformations or synthetic data generation process is not carefully controlled. It is important to analyze the data for any emerging bias or imbalanced representations, ensuring that the model does not become overly biased toward specific features [70].

As real-world conditions evolve, regularly update and refine simulation environments to reflect new data, trends, or scenarios that may affect the application. This is especially important in fields such as robotics or autonomous driving, where new technological advances could affect the operational environment [124].

When possible, use transfer learning techniques to benefit from pre-trained models on a diverse set of real-world data, and then fine-tune the model with augmented or simulated data. This allows leveraging the benefits of large, diverse datasets while still incorporating specific domain-focused augmentation or simulation [129].

After training a model with augmented or simulated data, evaluate its performance on a separate real-world dataset (if available) to ensure that the model can generalize beyond the specific transformations or synthetic scenarios used during training [130].

Active learning can complement both augmentation and simulation. Researchers can use active learning strategies to identify the most informative samples for training, and then apply augmentation or simulation techniques to generate additional variations of those samples for further improvement [131].

Always ensure that edge cases are considered, especially when simulating environments. This includes testing rare or extreme scenarios that might not appear often in the real data but are crucial for the robustness and reliability of the model [132].

Both data augmentation and simulation can be computationally intensive. Use optimized algorithms and techniques, such as parallel processing, to handle the heavy computational load. In simulation, consider simplifying models or reducing simulation time while still maintaining the quality of generated data [133].

Document the augmentation and simulation techniques used thoroughly. This includes specifying the parameters, models, or algorithms involved in generating the augmented or simulated data, to ensure reproducibility and transparency in your research process [134].

3.6. Probabilistic Density Estimation:

Technical Goal & Challenges: The primary technical goal of probabilistic density estimation in the context of generative AI is to learn an accurate approximation of the underlying probability distribution that governs a given dataset. This is crucial for tasks such as sample generation, likelihood estimation, anomaly detection, and downstream decision-making processes. Generative models such as Variational Autoencoders (VAEs) [107], Normalizing Flows [27], and Diffusion Models [40] leverage density estimation to model complex, high-dimensional data distributions in a tractable and efficient way. Accurate density estimation enables the model to produce high-quality, diverse, and coherent samples that resemble real data.

 doi:10.20944/preprints202511.0421.v1

Table 4. Key Challenges and Objectives in Generative Modeling

Aspect	Description	Representative Techniques
High-Fidelity Data Synthesis	Generate data nearly indistinguishable from real samples.	GANs, Diffusion Models, VAEs
Controllable / Conditional Generation	Condition generation on labels, layouts, or prompts.	cGANs, Conditional VAEs, Guided Diffusion
Sequential & Structured Data Modeling	Model temporal or relational data like sequences and graphs.	RNNs, LSTMs, Transformers, GNNs, GATs
Representation Learning & Disentanglement	Learn interpretable, structured latent spaces.	VAEs, β-VAEs, InfoGANs
Data Augmentation & Simulation	Expand dataset or simulate environments for training.	Transformations, Procedural Simulations
Probabilistic Density Estimation	Model true data distribution $P(x)$ for generative tasks.	KDE, GMMs, VAEs, Flows, PixelCNN

However, achieving this goal involves several technical challenges. Firstly, high-dimensional data (such as images, audio, and text) often resides on low-dimensional manifolds, making the true data distribution difficult to capture accurately [101]. Secondly, balancing expressivity and tractability is non-trivial—while expressive models may approximate distributions more closely, they often require complex architectures and incur high computational costs [135]. Thirdly, ensuring stability and convergence during training is challenging, particularly when optimizing objectives that involve intractable likelihoods or require approximation techniques (e.g., variational inference) [136]. Moreover, evaluating the quality of the learned distribution is itself a complex task, as standard metrics may not fully capture generative performance. Finally, real-world data is often noisy, incomplete, or non-stationary, adding another layer of difficulty to effective density estimation. Addressing these challenges is critical to improving the reliability and performance of generative AI systems in practical applications. **Leading Paradigms & Technical Rationale:** Probabilistic density estimation is a foundational task in generative AI, enabling the modeling and sampling of data from complex, high-dimensional distributions. The leading paradigms in this area include Variational Autoencoders (VAEs) [107], Normalizing Flows [27,135], Energy-Based Models [137], and Diffusion Probabilistic Models [40]. Each of these approaches offers unique advantages and trade-offs with respect to expressivity, tractability, and sampling efficiency.

VAEs adopt a latent-variable framework, approximating the intractable posterior using variational inference. This enables efficient learning of a lower-dimensional latent representation while maintaining a principled probabilistic foundation [107]. Normalizing Flows transform simple base distributions into complex ones through a sequence of invertible mappings, offering exact likelihood computation and flexible density modeling [135]. Diffusion Models, in contrast, model data as the reversal of a diffusion (noise injection) process, achieving state-of-the-art results in image synthesis [40] while bypassing the need for tractable likelihoods. Energy-Based Models (EBMs) define unnormalized probability densities via energy functions and offer flexibility in modeling multi-modal distributions [137], although they are typically more challenging to train.

The technical rationale behind these paradigms is to strike a balance between expressive power and computational feasibility. Likelihood-based models allow for principled training and evaluation, but often face challenges in scalability or inference. Generative AI systems rely on these paradigms to generate realistic, diverse data samples, understand uncertainty, and enable applications like image generation, text synthesis, and data augmentation. The choice of paradigm is often governed by the application's tolerance for approximation, computational resources, and the desired balance between sampling quality and interpretability.

 doi:10.20944/preprints202511.0421.v1

Comparative Technical Analysis: A comparative technical analysis of probabilistic density estimation paradigms in generative AI reveals distinct trade-offs among the leading approaches: Variational Autoencoders (VAEs), Normalizing Flows, Diffusion Models, and Energy-Based Models (EBMs). VAEs [107] are computationally efficient and offer fast sampling and latent space interpretability, but often produce blurrier outputs due to the use of a lower bound on the log-likelihood and a factorized Gaussian posterior. Normalizing Flows [135] enable exact likelihood computation and support both inference and sampling, but they require invertible transformations and Jacobian determinants, which constrain model architecture and increase computational complexity.

Diffusion Models [40] surpass both VAEs and GANs in generative quality, especially for image synthesis, by modeling the data distribution as a reversal of a stochastic noise process. However, they are computationally expensive at sampling time due to the iterative denoising steps, which can involve hundreds or thousands of forward passes. EBMs [137], on the other hand, are highly flexible and can represent multi-modal distributions without requiring normalized probability densities. Nevertheless, they suffer from difficulties in sampling (e.g., requiring Langevin dynamics or contrastive divergence) and training instability due to the absence of tractable likelihoods.

Overall, VAEs are preferred for applications requiring latent structure and fast sampling; Normalizing Flows are ideal for tasks demanding exact likelihoods and continuous density control; Diffusion Models are optimal for high-fidelity generation at the cost of speed; and EBMs offer the most flexible framework for unnormalized modeling, albeit with significant training and sampling hurdles. The choice of model depends on application-specific requirements regarding generation quality, interpretability, and computational budget.

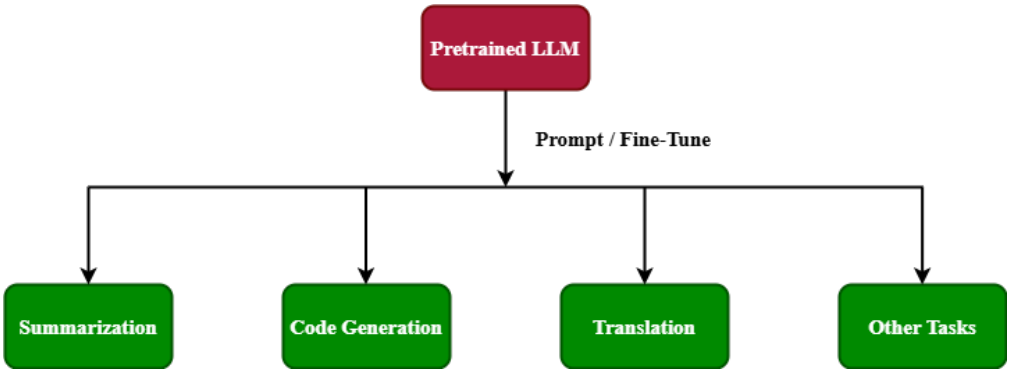

Figure 5. A pretrained LLM adapted via prompting or fine-tuning for multiple downstream tasks.

Best Practices for Researchers:

Understand the Data Distribution: Analyze the nature and dimensionality of the dataset to choose an appropriate model that can capture its complexity effectively.

Select Models Based on Application Needs: Balance trade-offs between model expressivity, computational cost, and inference speed depending on whether the goal is high-fidelity generation, fast sampling, or interpretable latent structure.

Use Proper Evaluation Metrics: Employ a combination of likelihood-based measures, sample quality scores (e.g., FID, IS), and domain-specific metrics to comprehensively assess model performance.

Regularize and Monitor Training: Apply suitable regularization techniques and monitor training stability closely, especially for complex models like EBMs and diffusion models, to prevent mode collapse or overfitting.

Leverage Variational Inference and Approximate Methods: When exact inference is intractable, use principled approximations such as variational inference or Monte Carlo methods while carefully analyzing approximation errors.

Incorporate Domain Knowledge: Embed structural or physical constraints into the model architecture or priors to improve learning efficiency and interpretability.

 doi:10.20944/preprints202511.0421.v1

Experiment with Hybrid Approaches: Combine multiple paradigms (e.g., flows with VAEs or EBMs with diffusion models) to leverage complementary strengths.

Ensure Reproducibility: Share code, hyperparameters, and training protocols to enable replication and extension of research findings.

Stay Updated with Recent Advances: Follow the latest literature and open-source developments to incorporate improvements in architecture, optimization, and evaluation.

Be Mindful of Ethical Considerations: Consider biases, data privacy, and the societal impact of generative models during research and deployment.

The key challenges and objectives in generative modeling is described in Table 4.

4. Advancing & Cross-cutting Best Practices

Leveraging Foundation Models: Large Language Models (LLMs) are widely used in many AI tasks due to their strong pretraining on diverse datasets. This training helps them generalize well, even with little task-specific data. Instead of training from scratch, researchers can prompt or fine-tune these models for specific tasks. This speeds up development, works well in low-resource settings, and allows easy adaptation across domains. [138].

Figure 5 illustrates how a pre-trained large language model can be adapted via prompting or fine-tuning to perform multiple downstream tasks, including summarization, code generation, translation, and other applications. Its key feature is the flexibility to handle diverse tasks without retraining from scratch.

4.1. Technical Strategies Beyond Basic Fine-Tuning

Modern AI systems are increasingly reliant on foundation models, prompting the development of advanced strategies to improve adaptability, efficiency, and interpretability. Techniques in prompt engineering, such as Chain-of-Thought (CoT) reasoning [139], Tree-of-Thought (ToT) planning [140], and self-consistency decoding [141], enhance the reasoning abilities of models by not altering the weights, which allows for successful zero-shot or few-shot learning.

Compared to full model fine-tuning, parameter-efficient fine-tuning methods, including LoRA [81], QLoRA [142], and Adapters [71], provide scalable and cost-effective options. LoRA involves adding trainable low-rank matrices to attention layers, QLoRA integrates quantization with LoRA for efficient fine-tuning on limited hardware, and Adapters involves adding lightweight task-specific modules between layers. The various methods differ in terms of memory requirements, training expenses, and deployment adaptability, making them suitable for diverse operational settings.

Model merging and editing techniques, such as task arithmetic [143] and knowledge injection or editing [144], enable post hoc behavioral alteration without requiring a full model retraining. The choice between fine-tuning and prompting hinges on specific use-case requirements: prompting performs well in general-purpose or low-resource situations, whereas fine-tuning is preferred for domain-specific tasks that require high accuracy and reliability.

4.2. Data Engineering & Curation

Data engineering and curation include collecting, processing, validating, storing, and managing data to ensure it is high-quality and ready for use [145]. Data comes from various sources like APIs, databases, and sensors, then goes through integration and transformation for consistency and standard formatting [146].

Preprocessing tasks involve removing duplicates, fixing missing values, and applying standard formats [147]. Validation checks for completeness, consistency, and correctness to ensure data reliability [148]. Long-term quality is maintained through ongoing assurance processes [149].

 doi:10.20944/preprints202511.0421.v1

Validated data is stored in databases or data lakes with proper indexing and metadata tagging to support easy access. Curation involves organizing, documenting, and preserving data for future use, while also addressing governance, privacy, and compliance [150,151].

Efficient engineering and curation pipelines create clean, reusable datasets that are essential for accurate analytics and machine learning [152].

Figure 6 highlights the stages of a data pipeline, including collection, cleaning, storage, and curation. These steps ensure data quality, proper organization, and usability for downstream machine learning and analytical tasks.

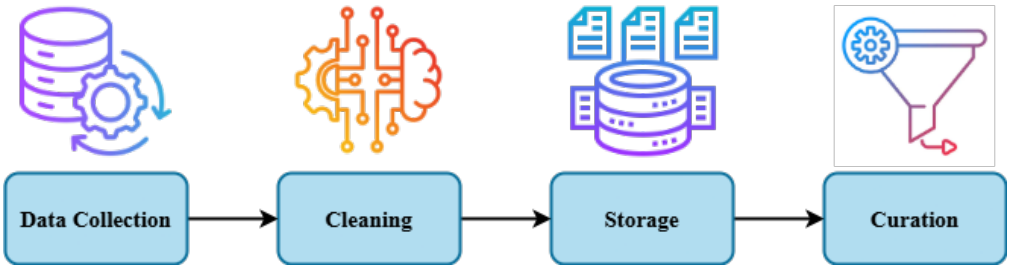

Figure 6. Stages of a data pipeline from collection to curation.

4.3. Technical Aspects of Large-Scale Data Handling

Managing large-scale data requires addressing challenges in quality assurance, duplication, data generation, and class balance. This section outlines key technical methods to support scalable and effective data pipelines.

Quality Filtering Heuristics: High-quality data is essential for effective learning. *Aesthetic scoring* evaluates structural or visual appeal—especially in image and text datasets—filtering samples based on attributes such as resolution or clarity. *CLIP filtering* [153], leveraging pretrained contrastive models, scores image-text alignment and removes mismatched or noisy samples, significantly improving dataset quality.

Deduplication Algorithms: Duplicate data skews model learning and increases computational waste. *MinHash* [154], a probabilistic technique, estimates set similarity by comparing hashed signatures, enabling scalable and efficient detection of duplicates in large datasets.

Synthetic Data Generation: When data is scarce or privacy-restricted, synthetic data generation using Generative Adversarial Networks (GANs) [1] or variational autoencoders (VAEs) can augment datasets. These methods preserve statistical characteristics of the original data and are especially useful for rare or imbalanced scenarios, while also aiding in privacy preservation.

Pre-training Data Mixology: The composition of pretraining data—referred to as *data mixology*—directly impacts downstream performance. Diverse, high-quality mixtures improve generalization, while poorly curated blends can introduce biases or degrade task-specific results [153]. Curating balanced, representative pretraining datasets is essential for model robustness.

Handling Data Imbalance: Imbalanced datasets lead to biased predictions. Techniques such as *resampling* (oversampling minority or undersampling majority classes), *cost-sensitive learning*, and *synthetic minority over-sampling* address this issue. Advanced strategies include *focal loss* [155], which shifts training focus toward harder, underrepresented examples.

4.4. Computational Efficiency & Scaling

Training large models requires optimized use of memory, compute, and distributed systems. This section highlights foundational and advanced techniques for efficient model training and scaling.

Distributed Training Frameworks: Techniques such as *DeepSpeed* with ZeRO stages [156] and *Fully Sharded Data Parallel* (FSDP) [157] enable training of massive models by partitioning parameters, gradients, and optimizer states across GPUs. This dramatically reduces memory usage and enables scalability.

 doi:10.20944/preprints202511.0421.v1

Gradient Checkpointing & Mixed Precision: *Gradient checkpointing* [158] stores only select activations during the forward pass and recomputes them as needed in the backward pass, significantly conserving memory. *Mixed-precision training*—using formats like `float16` or `bfloat16`—accelerates computation and reduces memory usage with minimal accuracy trade-off [159,160].

Efficient Attention and Model Compression: Attention mechanisms such as *FlashAttention* [161] and *linear attention* [162] reduce the quadratic time complexity of standard transformers. Model compression methods, including *pruning* [163] and *quantization-aware training* [164], reduce model size and accelerate inference, enabling deployment on edge devices.

Scaling Laws: Empirical *scaling laws* [165] offer predictive frameworks linking model size, dataset size, and compute to performance, helping practitioners optimize resource allocation and plan future model scaling efforts.

4.5. Debugging, Reproducibility & Evaluation Toolkits

Ensuring reliable and reproducible model development requires effective debugging and standard evaluation practices.

Debugging Techniques: Common issues such as *mode collapse*, *posterior collapse*, and *exposure bias* are prevalent in generative and sequence models [1,18]. Tools like *Weights & Biases* [166] and *TensorBoard* [167] allow monitoring of training dynamics, including loss curves, gradient flows, and activation distributions. Gradient/activation tracking helps identify vanishing gradients, dead neurons, or unstable updates, facilitating faster debugging and iteration [4].

Reproducibility Practices: *Containerization* with Docker ensures environment consistency across experiments, while tools like *Conda* manage dependencies and versions. These practices enable reproducible experimentation across development pipelines.

Standardized Evaluation Harnesses: Tools like *HELM* (Harnessing Evaluations with Language Models) [168] and the *EleutherAI Eval Harness* [169] support consistent multi-metric evaluations across tasks, facilitating fair comparisons and benchmarking.

4.6. Responsible AI (Technical Implementations)

Responsible AI ensures fairness, safety, transparency, and privacy in machine learning systems. This section highlights practical technical approaches to operationalizing these principles.

Bias Detection: Embedding analysis using *Principal Component Analysis (PCA)* and *clustering* helps reveal demographic or representational biases in learned representations [170].

Fairness-Aware Learning Objectives: Techniques such as *adversarial debiasing* and *re-weighting* modify training dynamics to prevent discrimination based on sensitive attributes.

Content Moderation and Safety Filtering: Generative systems require guardrails to prevent harmful outputs. *Perplexity-based filters* and *classifier-based moderation* [171] remove toxic, unsafe, or non-compliant content before deployment.

Watermarking Techniques: *Invisible watermarking* [172] embeds imperceptible signatures in generated outputs, enabling traceability, authenticity verification, and deepfake detection.

Differential Privacy: Privacy-preserving learning with *Differentially Private Stochastic Gradient Descent (DP-SGD)* [167] injects noise into gradients during training, ensuring individual data samples cannot be reverse-engineered, thereby preserving user privacy.

By integrating these responsible AI techniques, developers can build systems that are both technologically powerful and aligned with societal and ethical standards.

5. Technical Challenges & Future Directions

5.1. Key Open Problems

Despite significant progress in AI research, several technically challenging gaps remain that need to be addressed to further advance generative models and their applications. Some of these open problems include:

 doi:10.20944/preprints202511.0421.v1

Robust Compositional Generalization: Achieving robust compositional generalization, where models can correctly interpret and generate novel combinations of learned concepts, remains a significant challenge. Current models often struggle with compositional tasks, particularly when generalizing to new compositions that were not seen during training [173].

Long-range Coherence: Ensuring long-range coherence in generative models, such as in long text generation or video synthesis, is a persistent problem. Generating coherent outputs over long sequences without resorting to excessive computational resources or losing semantic consistency is a critical area of research [13].

Sample-efficient Learning of Complex Multimodal Distributions: Learning complex multimodal distributions with fewer samples is a key challenge. Current methods often require large amounts of labeled data, which is not always available. Developing more sample-efficient learning methods could lead to better generalization with less data [4].

Unifying Generative Paradigms: While multiple generative models exist (e.g., GANs, VAEs, normalizing flows), there is no unified approach that seamlessly integrates the strengths of these paradigms. A unified framework could allow for more versatile and efficient generative models [1].

Theoretically Grounded Metrics for Creativity and Novelty: Defining and quantifying metrics for creativity, novelty, and out-of-distribution detection remains an open problem. Existing evaluation methods are often subjective and inconsistent, making it difficult to compare generative models in terms of their ability to generate truly novel content [174].

These problems represent key research directions that will drive the next generation of AI advancements, particularly in generative modeling and multimodal learning.

5.2. Focus on Technical Methods

The technical methods employed in Responsible AI are essential for ensuring fairness, privacy, and safety in AI systems. These methods focus on detecting and mitigating bias, ensuring ethical content generation, and preserving privacy during training. Below are the key technical approaches that have been widely adopted:

Algorithmic Bias Detection in Embeddings: Bias detection in embeddings plays a crucial role in ensuring fairness in AI models. **Principal Component Analysis (PCA)-based techniques** are commonly used to analyze the structure of embeddings and identify any skewed representations of different demographic groups. PCA helps to detect and highlight biased associations in the model's learned embeddings. Furthermore, clustering methods can group embeddings to reveal disparities in the representation of various groups, such as gender or race, within the model's learned features. These detection methods are vital for identifying and addressing potential algorithmic biases that could negatively affect certain populations [170].

Fairness-Aware Learning Objectives: To enhance fairness in AI models, various fairness-aware learning objectives have been developed. One prominent technique is adversarial debiasing, where models are trained to make predictions while minimizing the influence of sensitive attributes (e.g., race or gender) on those predictions. This is achieved by integrating an adversarial network that introduces a regularization term that penalizes biased relationships in the model's predictions. Another key method is re-weighting, which assigns different weights to training examples in order to address class imbalances and improve the model's sensitivity to underrepresented groups. These fairness-aware objectives help ensure that AI models produce equitable outcomes.

Content Moderation and Safety Filtering: Content moderation is a critical concern for generative models, especially in the context of AI-generated text, images, and videos. **Perplexity filters** are commonly used to measure the unpredictability of generated text, identifying potentially harmful or nonsensical content. A higher perplexity often indicates problematic content that may not align with safety or ethical guidelines. Additionally, **classifier-based methods** are deployed to flag and filter inappropriate content. These classifiers are trained on labeled datasets to recognize and remove offensive or harmful content before it is generated or released, ensuring the outputs comply with safety standards [171].

 doi:10.20944/preprints202511.0421.v1

Watermarking Techniques for Generated Content: With the increasing prevalence of deepfakes and synthetic media, ensuring the authenticity and traceability of AI-generated content is of paramount importance. **Watermarking techniques** are used to embed invisible markers within generated media such as images, audio, or video. These watermarks allow the origin of the content to be tracked, ensuring accountability and helping to combat misuse, such as in disinformation campaigns. Watermarking techniques play a key role in preventing malicious exploitation of AI-generated content [172].

Differential Privacy Mechanisms: Differential privacy is a crucial technique for safeguarding individual privacy during model training, particularly when working with sensitive data. **Differentially Private Stochastic Gradient Descent (DP-SGD)** is one approach that introduces noise to the gradients during training, preventing the model from memorizing specific data points. By adding this noise, DP-SGD ensures that individual information cannot be inferred from the model, maintaining privacy while still allowing the model to learn effectively. This mechanism is particularly important in generative training, where protecting user data is vital [167].

6. Conclusions

This review has provided a clear overview of generative tasks by organizing them into a structured taxonomy. We discussed important areas such as task-driven analysis, high-quality data synthesis, controllable generation, sequential and structured modeling, representation learning, data augmentation, and probabilistic density estimation. These represent the core directions of generative AI research.

We also highlighted best practices that go beyond model design, including effective fine-tuning methods, careful data engineering, handling large-scale datasets, improving computational efficiency, ensuring reproducibility, and applying responsible AI principles. Together, these practices build the foundation for creating more reliable and practical generative models.

A key message of this review is that progress in generative AI depends on making informed technical choices. Researchers need to understand the strengths and limits of different models, their mathematical foundations, training dynamics, and computational costs. This technical understanding helps guide better decisions in applying generative models to real problems.

Finally, we outlined open challenges and future directions. Questions around controllability, evaluation, efficiency, and responsibility remain critical for the next stage of development. Addressing these issues will be essential to unlock the full potential of generative AI across different applications.

In summary, this review offers a structured and accessible framework for studying generative AI. By mapping key tasks, methods, and challenges, it aims to support researchers in making meaningful and impactful contributions in this fast-growing field.

References

1. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; et al.. Generative Adversarial Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2014.
2. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020, Vol. 33.
3. Dhariwal, P.; Nichol, A. Diffusion Models Beat GANs on Image Synthesis. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2021, Vol. 34.
4. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *International Conference on Learning Representations (ICLR)* **2014**. arXiv:1312.6114.
5. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 8110–8119. https://doi.org/10.1109/CVPR42600.2020.00813.
6. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784* **2014**.
7. Sohn, K.; Lee, H.; Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2015, Vol. 28.
8. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Zero-Shot Text-to-Image Generation. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.

 doi:10.20944/preprints202511.0421.v1

9. Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; Fleet, D.J. Video Diffusion Models. In Proceedings of the Advances in Neural Information Processing Systems; Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; Oh, A., Eds. Curran Associates, Inc., 2022, Vol. 35, pp. 8633–8646.
10. Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Poole, B.; McGrew, B.; Sutskever, I.; Chen, M. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In Proceedings of the Proceedings of the 38th International Conference on Machine Learning (ICML), 2021.
11. Elman, J.L. Finding Structure in Time. *Cognitive Science* **1990**, *14*, 179–211.
12. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Łukasz Kaiser.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017.
14. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
15. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the International Conference on Learning Representations (ICLR), 2018.
16. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 1798–1828.
17. Bengio, Y. Deep Learning of Representations for Unsupervised and Transfer Learning. In Proceedings of the Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012.
18. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
19. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2016.
20. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* **2019**, *6*, 60.
21. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 23–30.
22. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv preprint arXiv:1712.04621* **2017**.
23. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Chapman and Hall, 1986.
24. McLachlan, G.J.; Peel, D. *Finite Mixture Models*; John Wiley & Sons, 2000.
25. Papamakarios, G.; Nalisnick, E.; Rezende, D.J.; Mohamed, S.; Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research* **2021**, *22*, 1–64.
26. van den Oord, A.; Kalchbrenner, N.; Kavukcuoglu, K. Pixel Recurrent Neural Networks. In Proceedings of the International Conference on Machine Learning (ICML), 2016.
27. Rezende, D.; Mohamed, S. Variational inference with normalizing flows. In Proceedings of the International conference on machine learning. PMLR, 2015, pp. 1530–1538.
28. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.
29. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 6626–6637.
30. Dinh, L.; Sohl-Dickstein, J.; Bengio, S. Density Estimation using Real NVP. In Proceedings of the International Conference on Learning Representations (ICLR), 2017.
31. Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; Aila, T. Alias-Free Generative Adversarial Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2021.
32. Chan, E.R.; et al. Efficient Geometry-Aware 3D Generative Adversarial Networks. *arXiv preprint arXiv:2206.05185* **2022**.
33. Tewari, A.; et al. State of the Art on Neural Rendering. *Computer Graphics Forum* **2020**.
34. Dosovitskiy, A.; Brox, T. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2016.

 doi:10.20944/preprints202511.0421.v1

35. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In Proceedings of the International Conference on Computer Vision Workshops (ICCVW), 2021.
36. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2017.
37. Salimans, T.; et al. Improved Techniques for Training GANs. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2016.
38. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
39. Ramesh, A.; et al. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv preprint arXiv:2204.06125* **2022**.
40. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. *arXiv:2006.11239* **2020**.
41. Song, Y.; Ermon, S. Score-Based Generative Modeling through Stochastic Differential Equations. *NeurIPS* **2020**.
42. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems (NeurIPS)* **2014**, *27*, 2672–2680.
43. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *CVPR* **2019**.
44. Huang, X.; Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017, pp. 1501–1510.
45. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *MICCAI* **2015**.
46. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. *NeurIPS* **2017**.
47. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv:1511.06434* **2015**.
48. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. *ICLR* **2018**.
49. Kubicki, K.; Michalski, M. ADA: Augmented Data Augmentation for GANs. *ICLR* **2020**.
50. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *Advances in Neural Information Processing Systems* **2017**, *30*.
51. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. *ICLR* **2018**.
52. Binkowski, M.; Lucic, M.; Seitzer, M.; Gelly, S.; Schultz, T.; Pietquin, O. Demystifying MMD GANs. *ICLR* **2018**.
53. Salimans, T.; Goodfellow, I.; Zaremba, W.; Chen, X.; He, X.; Zhu, Y. Improved Techniques for Training GANs. *NeurIPS* **2016**.
54. Karras, T.; Laine, S.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. *CVPR* **2020**.
55. Karras, T.; Aittala, M.; Laine, S.; Karras, E.H.A.; Lehtinen, J.; Aila, T. Alias-Free Generative Adversarial Networks. *NeurIPS* **2021**.
56. Keskar, N.S.; McCann, B.; Varshney, L.R.; Xiong, C.; Socher, R. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858* **2019**.
57. Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; Xing, E.P. Toward controlled generation of text. In Proceedings of the International Conference on Machine Learning. PMLR, 2017, pp. 1587–1596.
58. Prabhumoye, S.; Tsvetkov, Y.; Salakhutdinov, R.; Black, A.W. Exploring controllable text generation techniques. In Proceedings of the Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 105–120.
59. Du, J.; Wang, C.; Huang, M.; Zhang, J. Variational discrete representation for controllable text generation. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022.
60. Liu, X.; Sheldon, D.; McCallum, A. DExperts: Decoding-time controlled text generation with experts and anti-experts. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3124–3136.
61. Lin, T.; Hilton, J.; Niekum, S. Plug and play language models: A simple approach to controlled text generation. In Proceedings of the International Conference on Learning Representations, 2020.
62. Qian, Q.; Zhang, X.; Zhao, W.X.; Hu, W.; Li, J. Controllable text generation with disentangled latent variables. *Neurocomputing* **2022**, *481*, 193–203.

 doi:10.20944/preprints202511.0421.v1

63. Ho, J.; Salimans, T. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598* **2022**.
64. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784* **2014**.
65. Miyato, T.; Koyama, M. cGANs with projection discriminator. In Proceedings of the International Conference on Learning Representations, 2018.
66. Gal, R.; Alaluf, Y.; Atzmon, M.; Patashnik, O.; Bermano, A.H.; Chechik, G.; Cohen-Or, D. StyleGAN-NADA: CLIP-guided Domain Adaptation of Image Generators. In Proceedings of the ACM Transactions on Graphics (TOG), 2022, Vol. 41, pp. 1–13.
67. Sohn, K.; Lee, H.; Yan, X. Learning Structured Output Representation using Deep Conditional Generative Models. *Advances in Neural Information Processing Systems* **2015**, *28*.
68. Li, X.L.; Liang, P. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL), 2021, pp. 4582–4597.
69. Radford, A.; Kim, J.W.; Hallacy, J.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, M.; et al. Learning Transferable Visual Models From Natural Language Supervision. *Proceedings of the International Conference on Machine Learning* **2021**.
70. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *NeurIPS* **2020**.
71. Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S. Parameter-efficient transfer learning for NLP. In Proceedings of the International Conference on Machine Learning. PMLR, 2019, pp. 2790–2799.
72. Sheng, E.; Chang, K.W.; Natarajan, P.; Peng, N. The Woman Worked as a Babysitter: On Biases in Language Generation. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, pp. 3407–3412.
73. Zhao, J.; Wang, T.; Yatskar, M.; Ordonez, V.; Chang, K.W. Gender bias in coreference resolution: Evaluation and debiasing methods. In Proceedings of the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 15–20.
74. Xu, L.; Zhang, Y.; Wu, X. AdaGN: Adaptive group normalization for conditional image generation. *IEEE TPAMI* **2022**.
75. Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; Courville, A. FiLM: Visual reasoning with a general conditioning layer. In Proceedings of the AAAI, 2018.
76. Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.T.; Parekh, Z.L.; Pham, H.; Le, Q.V.; Sung, Y.; Li, Z.; Duerig, T. Scaling up Visual and Vision-Language Representation Learning With Noisy Text Supervision. *ICML* **2021**.
77. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.
78. Moryossef, A.; Shuster, K.; Ju, D.; Weston, J. Filling the Gap: Learning Natural Language Image Search with Synthetic Datasets. In Proceedings of the EMNLP, 2021.
79. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155* **2022**.
80. Bai, Y.; Kadavath, S.; Kundu, S.; et al. Training a Helpful and Harmless Assistant with RLHF. *arXiv preprint arXiv:2204.05862* **2022**.
81. Hu, E.J.; Shen, Y.; Wallis, P.; et al. LoRA: Low-Rank Adaptation of Large Language Models. In Proceedings of the ICLR, 2021.
82. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the EMNLP, 2021.
83. Hessel, J.; Holtzman, A.; Forbes, M.; Choi, Y. CLIPScore: A Reference-Free Evaluation Metric for Image Captioning. In Proceedings of the EMNLP, 2021.
84. Gal, R.; Kadian, A.; Gat, Y.; Chechik, G.; Bermano, A.H. Image Generation With Multimodal Priors. In Proceedings of the ICLR, 2021.
85. Eastwood, C.; Williams, C.K.I. A framework for the quantitative evaluation of disentangled representations. In Proceedings of the ICLR, 2018.
86. Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; et al. OPT: Open Pre-trained Transformer Language Models. *arXiv preprint arXiv:2205.01068* **2022**.
87. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks* **1994**, *5*, 157–166. https://doi.org/10.1109/72.279181.

 doi:10.20944/preprints202511.0421.v1

88. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Computation* **1997**, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
89. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the Proceedings of ICLR, 2017.
90. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016.
91. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL* **2018**.
92. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving language understanding by generative pre-training. In Proceedings of the OpenAI Blog, 2018.
93. Breiman, L. *Random forests*; Vol. 45, 2001; pp. 5–32. https://doi.org/10.1023/A:1010933404324.
94. Freund, Y.; Schapire, R.E. Adaptive boosting algorithms for combining classifiers. In Proceedings of the Proceedings of ICML, 1999, pp. 160–167.
95. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, 2006.
96. Guyon, I.; Elisseeff, A. *Introduction to Feature Selection*; Vol. 3, 2003; pp. 1157–1182.
97. Friedman, J. *Greedy function approximation: A gradient boosting machine*; Vol. 29, 2001; pp. 1189–1232. https://doi.org/10.1214/aos/1013203451.
98. Dong, E.; Wei, W.; Zhang, W.X. *A new method for missing data imputation*; Vol. 5, 2013; pp. 263–276.
99. Little, R.J.; Rubin, D.B. *Statistical Analysis with Missing Data*; Wiley, 2018.
100. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems (NeurIPS)* **2017**, *30*.
101. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE TPAMI* **2013**.
102. Higgins, I.; Matthey, L.; Pal, A.; Burgess, C.; Glorot, X.; Botvinick, M.; Mohamed, S.; Lerchner, A. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *ICLR* **2017**.
103. Mathieu, M.; Zhao, J.J.; LeCun, Y. Disentangling disentanglement in variational autoencoders. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2019.
104. Locatello, F.; Bauer, S.; Lucic, M.; Raetsch, G.; Gelly, S.; Schölkopf, B.; Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2019, pp. 4114–4124.
105. Burgess, C.P.; Higgins, I.; Pal, A.; Matthey, L.; Watters, N.; Desjardins, G.; Lerchner, A. Understanding disentangling in beta-VAE. In Proceedings of the NeurIPS Workshop on Learning Disentangled Representations, 2018.
106. Lucas, J.; Tucker, G.; Grosse, R.B.; Norouzi, M. Don't blame the ELBO! A linear VAE perspective on posterior collapse. In Proceedings of the NeurIPS, 2019.
107. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114* **2013**.
108. Kim, H.; Mnih, A. Disentangling by factorising. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2018, pp. 2649–2658.
109. Chen, T.Q.; Li, X.; Grosse, R.B.; Duvenaud, D.K. Isolating sources of disentanglement in variational autoencoders. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2018, Vol. 31.
110. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2016, Vol. 29.
111. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning (ICML). PMLR, 2020.
112. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020, pp. 9729–9738.
113. Bouchacourt, D.; Tomioka, R.; Nowozin, S. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
114. Pineau, J.; Vincent-Lamarre, P.; Sinha, K.; Lariviere, V.; Beygelzimer, A.; d'Alché Buc, F.; Fox, E.; Larochelle, H. Improving reproducibility in machine learning research (A report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research* **2021**, *22*, 1–20.
115. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of Big Data* **2019**, *6*, 1–48.

 doi:10.20944/preprints202511.0421.v1

116. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, 2018, pp. 1082–10828.
117. Qi, X.; Li, J.; Cai, Z.; Ma, W.; Yu, W. Image synthesis with semantic segmentation and instance segmentation for automated driving datasets. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020, pp. 1141–1146.
118. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 3803–3810.
119. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019, pp. 113–123.
120. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for data: Ground truth from computer games. In Proceedings of the European Conference on Computer Vision (ECCV). Springer, 2016, pp. 102–118.
121. Zhao, C.; Jia, Q.; Wang, G.; Wang, Y.; Zhang, S.; Tian, J. Data augmentation with learned transformations for one-shot medical image segmentation. *Neurocomputing* **2020**, *388*, 34–43.
122. Wei, J.; Zou, K.; Yu, Z. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In Proceedings of the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.
123. Tamar, A.; Schwing, A.; Tenenbaum, J.; Zettlemoyer, L. Value iteration networks. In Proceedings of the Neural Information Processing Systems (NeurIPS), 2016.
124. Pomerleau, D. ALVINN: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems (NeurIPS)* **1989**, *1*, 305–313.
125. Alshammari, R. A survey on natural language processing tasks and approaches in data augmentation. *Journal of King Saud University-Computer and Information Sciences* **2019**, *31*, 720–734.
126. He, K.; Zhang, X.; Ren, S.; Sun, J. Bag of Tricks for Image Classification with Convolutional Neural Networks. *arXiv preprint arXiv:1904.06422* **2019**.
127. Pang, C.; Zhang, L.; Xu, W.; Zhang, K.; Yu, Y. Learning data augmentation strategies for deep learning. *arXiv preprint arXiv:2007.03956* **2020**.
128. Xu, X.; Yu, J.; Zhang, H. Deep reinforcement learning for autonomous driving: A survey. *IEEE Access* **2017**, *5*, 20757–20772.
129. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H.; et al. Transferable visual models: A study on a large collection of class-labeled images from the web. *Neural Information Processing Systems (NeurIPS)* **2014**, *27*.
130. Zoph, B.; Le, Q.V. Neural architecture search with reinforcement learning. *arXiv arXiv:1611.01578* **2016**.
131. Settles, B. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* **2012**, *6*, 1–114.
132. Zhang, T.; Liu, Z.; He, J. Edge computing for autonomous vehicles: A survey. *IEEE Access* **2020**, *8*, 213532–213547.
133. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O.; Moritz, P.; Ibarz, J.; Abbeel, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning (ICML), 2015.
134. Bengio, Y. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning* **2013**, *2*, 1–127.
135. Papamakarios, G.; Nalisnick, E.; Rezende, D.; Mohamed, S.; Lakshminarayanan, B. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research* **2021**, *22*, 1–64.
136. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association* **2017**, *112*, 859–877.
137. LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F.J. A Tutorial on Energy-Based Learning. *Predicting Structured Data* **2006**.
138. Bommasani, R.; Hudson, D.A.; Adeli, E.; et al.. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258* **2021**.
139. Wei, J.; Wang, X.; Schuurmans, D.; et al.. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2022.
140. Yao, S.; Zhao, D.; Zhang, I.G.; et al.. Tree-of-Thought Deliberate Reasoning. *arXiv arXiv:2305.10601* **2023**.
141. Wang, X.; Wei, J.; Schuurmans, D.; et al.. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *arXiv preprint arXiv:2203.11171* **2022**.

 doi:10.20944/preprints202511.0421.v1

142. Dettmers, T.; Pagnoni, A.; Holtzman, A.; et al.. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314* **2023**.
143. Ilharco, G.; Dodds, Z.; Wei, J.; et al.. Editing Models with Task Arithmetic. *arXiv arXiv:2301.01704* **2023**.
144. De Cao, N.; Aziz, W.; Titov, I. Inserting Knowledge into Pre-trained Models via Knowledge Fact Injection. In Proceedings of the International Conference on Learning Representations (ICLR), 2021.
145. Hernandez, M.; Tan, L. *Principles of Data Engineering*; TechPress, 2018.
146. Zikopoulos, P. *Data Integration and Governance: Best Practices*; IBM Press, 2017.
147. Batini, C.; Cappiello, C.; Francalanci, C.; Maurino, A. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)* **2009**, *41*, 1–52.
148. Olson, J. Data Quality: The Accuracy Dimension. *Morgan Kaufmann* **2016**.
149. Chen, L.; Chen, X. Data Quality Monitoring: A Comprehensive Survey. *ACM Computing Surveys* **2012**, *45*.
150. Kitchin, R. *The Data Revolution*; Sage, 2014.
151. Vines, T.; Andrew, R. The Value and Impact of Data Curation. *BioScience* **2014**, *64*, 6–12.
152. Santos, M.; Pereira, J. Building Scalable Data Pipelines. *Journal of Big Data* **2016**, *3*.
153. Radford, A.; Kim, J.W.; Hallacy, C.; et al.. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020* **2021**.
154. Broder, A. On the Resemblance and Containment of Documents. In Proceedings of the Compression and Complexity of Sequences, 1997.
155. Lin, T.Y.; Goyal, P.; Girshick, R.; et al.. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.
156. Rasam, S.; Narayanan, D.; Shoeybi, M.; et al.. DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters. *arXiv preprint arXiv:2007.03068* **2020**.
157. Zhang, P.; Shoeybi, M.; Patwary, M.; et al.. Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM. *arXiv preprint arXiv:2104.04473* **2021**.
158. Griewank, A.; Walther, A. Algorithm 799: Revolve: An Implementation of Checkpointing for the Reverse or Adjoint Mode of Computational Differentiation. *ACM Transactions on Mathematical Software* **2000**, *26*, 19–45.
159. Micikevicius, P.; Narang, S.; Alben, J.; et al.. Mixed Precision Training. In Proceedings of the International Conference on Learning Representations (ICLR), 2018.
160. Nangia, N.; Gupta, R.; He, W.; et al.. Training Transformers with 8-bit Floating-Point. *arXiv preprint arXiv:2009.04013* **2020**.
161. Dao, T.; Fu, D.; Ermon, S.; Ré, C. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. *arXiv preprint arXiv:2205.14135* **2022**.
162. Katharopoulos, A.; Vyas, A.; Patil, N.; Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In Proceedings of the International Conference on Machine Learning (ICML), 2020.
163. Han, S.; Mao, H.; Dally, W. Deep Compression: Compressing Deep Neural Networks With Pruning, Trained Quantization and Huffman Coding. *arXiv preprint arXiv:1510.00149* **2015**.
164. Jacob, B.; Kligys, S.; Chen, B.; et al.. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
165. Kaplan, J.; McCandlish, S.; Henighan, T.; et al.. Scaling Laws for Neural Language Models. *arXiv preprint arXiv:2001.08361* **2020**.
166. Biewald, L. Weights & Biases. *Software* **2020**. https://wandb.ai.
167. Abadi, M.; Chu, A.; Goodfellow, I.; et al.. Deep Learning with Differential Privacy. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), 2016.
168. Helström, L.; Li, T.; Sorensen, A.; et al.. HELM: A Holistic Evaluation of Language Models. *arXiv preprint arXiv:2210.15900* **2022**.
169. EleutherAI. EleutherAI Language Model Evaluation Harness. https://github.com/EleutherAI/lm-evaluation-harness, 2021.
170. Barocas, S.; Hardt, M.; Narayanan, A. Fairness and Machine Learning. *Book manuscript* **2019**.
171. Merity, S.; Xiong, C.; Bradbury, J.; Socher, R. Pointer Sentinel Mixture Models. *arXiv arXiv:1611.01726* **2017**.
172. Maimour, M.; Siahbani, M. Invisible Watermarking for Deepfake Detection. *arXiv arXiv:1909.01746* **2019**.

 doi:10.20944/preprints202511.0421.v1

173. Lake, B.; Ullman, T.; Tenenbaum, J.; Gershman, S. Building Machines That Learn and Think Like People. In Proceedings of the Behavioral and Brain Sciences, 2017.
174. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In Proceedings of the International Conference on Learning Representations (ICLR), 2019.