Volume (2) No (1), 2026
Journal of Global Research in Multidisciplinary Studies (JGRMS)
Review Paper
Available online at https://saanvipublications.com/journals/index.php/jgrms/index

Self-Supervised Learning in Generative AI: Enhancing Model Efficiency and Adaptability for Limited Data Scenarios

Arvind Kumar[1st]
Assistant Professor
Department of CSE(CS)
RTCIT Ranchi, Jharkhand
kumar.arvind681@gmail.com

Basudeo Mahato[2nd]
Assistant Professor
Department of CSE
RTC Institute of Technology
asudeo2016@gmail.com

Anamika Kumari[3rd]
Assistant Professor
Department of ECE
RTC Institute of Technology
anamikakumari8567@gmail.com

***Abstract*—Recent advancements in Generative AI and deep learning have demonstrated remarkable capabilities in producing high-quality, human-like outputs across domains such as text, image, and audio generation. However, these models often rely on extensive labeled datasets, which can be resource-intensive to curate. This research focuses on the integration of self-supervised learning (SSL) paradigms into Generative AI to address the challenges of limited data availability, domain adaptation, and computational efficiency.**

The proposed study explores novel SSL techniques tailored for generative tasks, including contrastive learning, masked prediction, and clustering-based approaches. It examines their impact on enhancing model generalization, reducing overfitting, and improving robustness against adversarial attacks. Additionally, it investigates hybrid architectures combining SSL with transformer-based and diffusion models to optimize performance across diverse generative applications. By leveraging SSL, this research aims to bridge the gap between data efficiency and generative quality, paving the way for more accessible and adaptable AI systems capable of thriving in real-world, data-sparse environments.

***Keywords*—Self-Supervised Learning (SSL), Generative AI, Masked Autoencoders (MAE), Contrastive Learning, Feature Extraction.**

I. INTRODUCTION

A. Background on Generative AI and its Dependence on Large Labeled Datasets.

Generative AI, a branch of artificial intelligence focused on creating new content, has seen remarkable advancements with models like GPT (text), DALL•E (images), and WaveNet (audio) [1][2]. These models have the ability to generate human-like text, create realistic images, and synthesize lifelike sounds, driving innovation across various industries such as entertainment, healthcare, and education. The core of Generative AI lies in training complex architectures, such as transformers and GANs, to understand and recreate patterns from data [3][4].

However, the success of these models heavily depends on access to large, high-quality labeled datasets. For instance, language models are trained on extensive corpora with billions of words, while image-generation models rely on datasets annotated with detailed metadata [5][6]. This dependence introduces several challenges [7]. First, obtaining labeled data is resource-intensive, requiring significant time, expertise, and cost [8]. Second, certain domains, such as medical imaging or legal documents, have limited labeled data due to privacy concerns or specialized knowledge requirements [9]. Third, the reliance on labeled data can lead to biases, as the data may not represent the diversity of real-world scenarios [10]. This dependence underscores the need for approaches like self-supervised learning, which can leverage vast amounts of unlabeled data, reducing the reliance on labeled datasets while maintaining generative quality and performance.

B. The Emergence of Self-Supervised Learning and Its Potential to Reduce Reliance on Labeled Data.

Self-supervised learning (SSL) has emerged as a powerful technique in machine learning, addressing one of the biggest challenges in AI: the need for large amounts of labeled data [11][12]. Unlike supervised learning, where models are trained on data paired with explicit labels, SSL enables models to learn useful representations directly from unlabeled data by creating pretext tasks that predict parts of the data itself. These tasks might involve predicting missing parts of an image, reconstructing corrupted input, or learning to distinguish between similar and dissimilar data points [13][14][15]. The key advantage of SSL is its ability to leverage vast amounts of unlabeled data, which is much more abundant and easier to obtain than labeled data. This reduces the need for time-consuming and costly data labeling efforts, making it possible to train AI models in domains where labeled data is scarce, expensive, or difficult to acquire [16][17]. Furthermore, SSL promotes more generalized learning, as the model learns high-level patterns and representations from the data itself, leading to improved performance across a variety of tasks, even when labeled data is limited. By bridging the gap between the availability of unlabeled data and the performance requirements of complex AI systems, SSL offers a scalable, cost-effective way to build powerful models that can adapt to diverse, real-world scenarios without relying heavily on human annotation [18]. This is particularly valuable in fields such as healthcare, finance, and natural language processing, where labeled data is often limited.

C. Research Objectives and Scope with an Emphasis on Implementation.

The primary objective of this research is to explore and implement self-supervised learning (SSL) techniques to enhance the efficiency and adaptability of Generative AI models in scenarios with limited labeled data [19]. Generative AI models, while powerful, typically require extensive labeled datasets for training [20]. This dependency limits their applicability, especially in domains where labeled data is scarce, costly, or difficult to obtain. SSL offers a potential solution by enabling models to learn rich data representations from large volumes of unlabeled data, thus reducing the reliance on labeled examples. The key research objectives are as follows:

1) Investigate the Integration of SSL in Generative AI Models

The first objective is to explore various SSL techniques (e.g., contrastive learning, masked modeling, and clustering) and evaluate their effectiveness in generative tasks [21][22]. This includes examining how SSL can be adapted to improve the learning process of generative models like GANs, transformers, and diffusion models. The research will focus on how SSL can pretrain models on unlabeled data to create meaningful representations that are transferable to generative tasks such as text generation, image synthesis, and speech generation.

2) Enhance Model Efficiency

One of the key aims is to improve the efficiency of Generative AI models, particularly in terms of data usage and computational resources [23]. By leveraging SSL, the goal is to reduce the amount of labeled data required for fine-tuning generative models while maintaining or improving performance. This could involve minimizing the size of the labeled dataset needed for downstream tasks and speeding up training processes through better initialization of model parameters via self-supervised pretraining.

3) Assess Adaptability in Limited Data Scenarios

This objective focuses on testing the adaptability of SSL-enhanced generative models in data-scarce environments [24]. The research will implement SSL models and assess their ability to generalize to new, unseen domains or tasks with minimal labeled data. This includes evaluating performance on specific applications where labeled data is often sparse, such as medical imaging, rare-event detection, and content generation for niche domains (e.g., low-resource languages, specialized industries).

4) Develop a Framework for Hybrid SSL and Generative AI Architectures

Another key objective is to create a practical framework that integrates SSL techniques with existing generative models [25]. This framework will guide the design of hybrid architectures that combine self-supervised pretraining with fine-tuning methods on smaller labeled datasets [26]. The implementation will involve designing and testing hybrid models that use SSL to learn generalizable representations, followed by fine-tuning them on specific generative tasks.

D. Scope of the Research

The scope of the research encompasses both theoretical exploration and practical implementation of SSL in generative tasks. The research will:

- Focus on deep learning models (e.g., GANs, transformers, and diffusion models) used for generating content in text, images, and audio.
- Implement SSL techniques such as contrastive learning, masked modelling, and self-training for unlabelled data to enable the generative models to learn more robust representations.
- Explore cross-domain applicability, where models trained on one dataset (e.g., natural images or general text) are tested on different, domain-specific tasks with limited labeled data (e.g., medical images or niche languages) [27][28].
- Provide comparative performance analysis between traditional supervised learning and SSL-enhanced models, focusing on metrics like training time, data efficiency, and output quality.

E. Implementation Emphasis

The emphasis on implementation will focus on the following:

1) Model Development

Implementing Generative AI models enhanced with SSL methods in a deep learning framework such as PyTorch or TensorFlow. This will involve the adaptation of existing architectures to support SSL, incorporating pretext tasks, and developing efficient training loops for both self-supervised and supervised stages.

2) Dataset Selection and Preprocessing

A variety of datasets will be used for evaluation, including both large unlabeled datasets (for pretraining SSL) and smaller labeled datasets (for downstream fine-tuning). The implementation will also focus on data preprocessing techniques to ensure the quality and diversity of input data.

3) Evaluation Metrics

Models will be evaluated on standard generative metrics, such as Fréchet Inception Distance (FID) for image quality, BLEU/ROUGE for text generation quality, and mean squared error (MSE) for audio synthesis. Additionally, metrics for data efficiency (e.g., the amount of labeled data required for satisfactory model performance) will be used.

4) Optimization and Scaling

Implementing optimization techniques, including hyperparameter tuning, regularization, and model ensembling, to achieve the best performance with reduced data requirements [29]. Efficient GPU/TPU utilization and training time management will also be explored to enhance scalability and practical usability.

II. Related Work

A. Overview of SSL techniques such as contrastive learning, masked modeling, and clustering.

Self-supervised learning (SSL) has become a significant approach for training machine learning models without relying heavily on labeled data. It leverages the inherent structure and patterns in the data itself, creating supervisory signals through pretext tasks. Below is an overview of key SSL techniques, including contrastive learning, masked modeling, and clustering.

1) Contrastive Learning

Contrastive learning is a widely used SSL technique that involves learning representations by comparing positive pairs

of samples (similar examples) and negative pairs (dissimilar examples). The goal is to encourage the model to learn features that can distinguish between similar and dissimilar inputs, which can be used for downstream tasks, including generative tasks.

- **Working:** In contrastive learning, a model is trained to map similar data points (e.g., different augmentations of the same image or sentence) to nearby points in the representation space, while pushing dissimilar points apart. This is typically achieved through a contrastive loss function such as InfoNCE or Triplet loss, which optimizes the model to pull similar pairs closer and push dissimilar pairs farther apart.
- **Applications:** Contrastive learning has been particularly successful in domains like computer vision (e.g., SimCLR for image representations) and natural language processing (e.g., Sentence-BERT for text representations). It has been integrated with generative models to improve their ability to generalize by learning semantically meaningful representations without labelled data.
- **Impact on Generative AI:** In generative tasks, contrastive learning helps models understand the relationships between different data points, which can aid in producing more coherent, contextually relevant outputs when fine-tuned for specific tasks.

2) Masked Modeling

Masked modeling is another prominent SSL technique where parts of the input data are deliberately hidden or masked, and the model is tasked with predicting the missing pieces. This technique has been particularly effective in natural language processing, notably in models like BERT (Bidirectional Encoder Representations from Transformers) for text, and has been adapted for vision tasks (e.g., MAE for images).

- **Working:** In masked modeling, a portion of the input data (such as random words in a sentence or pixels in an image) is replaced with a placeholder (e.g., a [MASK] token or zero value). The model is trained to predict the missing information based on the unmasked parts of the data. For example, BERT learns to predict masked words in a sentence, while MAE learns to reconstruct missing pixels in an image.
- **Applications:** Masked modeling is widely used in NLP tasks such as language modeling, text generation, and text classification. In computer vision, it has been applied to tasks like image inpainting and image generation by training models to reconstruct missing parts of images.
- **Impact on Generative AI:** For generative tasks, masked modeling enables a model to learn rich representations of data even with limited supervision. The ability to predict missing parts of an input makes the model more capable of generating coherent and realistic outputs by learning underlying structures and dependencies in the data.

3) Clustering-Based SSL

Clustering-based SSL techniques involve grouping similar data points together into clusters without using labels. This approach is rooted in the idea that data points that are similar in feature space should belong to the same cluster, and the model learns useful representations by exploiting this structure. Common methods in this category include DeepCluster, SwAV, and k-means clustering.

- **working:** In clustering-based SSL, the model is typically trained to assign data points to clusters. This can be done through unsupervised clustering algorithms like k-means or more advanced deep learning methods such as DeepCluster, which iteratively refines the cluster assignments based on learned representations. The model learns representations by minimizing the discrepancy between predicted cluster assignments and the true cluster assignments, without requiring explicit labels.
- **Applications:** Clustering-based SSL is often used in unsupervised representation learning for tasks such as object recognition, anomaly detection, and clustering analysis [30]. In generative AI, clustering helps models organize data into meaningful groups, improving the generation of coherent outputs when applied to text, images, or other data types.
- **Impact on Generative AI:** For generative tasks, clustering helps models understand the distribution of data and organize the learned representations into coherent groups [31]. This can be useful for generating diverse outputs that reflect the underlying structure of the data, such as generating images of various categories or creating text with distinct topics or styles.

B. Generative models like transformers, GANs, and diffusion models.

Generative models are a class of machine learning algorithms designed to generate new, synthetic data that resembles a given training dataset [32]. These models learn the underlying patterns and structures of the data to produce realistic outputs such as images, text, or audio. Below is a summary of some of the most influential generative models: transformers, Generative Adversarial Networks (GANs), and diffusion models.

1) Transformers

Transformers are a class of deep learning models that have become the backbone of many state-of-the-art generative tasks, particularly in natural language processing (NLP) and, more recently, in computer vision and multimodal domains.

- **Working:** Transformers use a self-attention mechanism to process sequences of data (e.g., text, images) in parallel, rather than sequentially like earlier models such as RNNs (Recurrent Neural Networks). This enables transformers to capture long-range dependencies and contextual information effectively. The model consists of an encoder-decoder architecture, where the encoder processes input data, and the decoder generates output data.
- **Applications:** Transformers have become foundational in NLP, with models like GPT (Generative Pretrained Transformer) for text generation, BERT for understanding tasks, and DALL·E for generating images from textual descriptions. These models excel in tasks requiring the generation of coherent, contextually aware content.
- **Generative Use Case:** In generative tasks, transformers are used for autoregressive generation (e.g., GPT models), where the model generates each

word or token step-by-step based on previously generated tokens [33]. They are also used in denoising models like BART (Bidirectional and Auto-Regressive Transformers), which are capable of generating high-quality content for a variety of applications.

2) Generative Adversarial Networks (GANs)

GANs are a type of generative model introduced by Ian Goodfellow in 2014. GANs are based on the idea of two neural networks, a generator and a discriminator, that are trained simultaneously in a game-theoretic framework.

- **Working:** The generator creates synthetic data (e.g., images) from random noise, while the discriminator attempts to distinguish between real data and the generator's fake data. The two networks are trained in opposition to each other: the generator tries to improve at creating realistic data, and the discriminator improves at identifying fake data. The training process continues until the generator produces highly realistic data that the discriminator can no longer easily differentiate from real data.
- **Applications:** GANs are widely used in image generation (e.g., creating photorealistic images, deepfake videos), image super-resolution, style transfer, and other creative domains. They are also applied in areas such as drug discovery, where generating molecular structures can be useful.
- **Generative Use Case:** GANs are particularly suited for tasks requiring the generation of high-quality visual content. Examples include generating realistic images of people, animals, or objects from random noise, as well as generating new artwork, music, and video sequences

3) Diffusion Models

Diffusion models are a newer class of generative models that have gained significant attention in recent years for their ability to produce high-quality, diverse outputs in image generation and other tasks. They work by simulating a diffusion process, gradually adding noise to data and then learning to reverse this process to generate new data from noise.

- **Working:** Diffusion models start by adding Gaussian noise to an image or other data until the data is completely random [34]. The model is then trained to reverse this diffusion process, step by step, to recover the original data. This training is typically done in a probabilistic framework where the model learns the reverse diffusion process to generate realistic samples from pure noise.
- **Applications:** Diffusion models have been particularly successful in high-quality image generation (e.g., Denoising Diffusion Probabilistic Models (DDPM) and Stable Diffusion) and have been used in tasks like image synthesis, inpainting, and text-to-image generation. These models have shown superior performance in generating diverse and high-quality outputs compared to traditional models like GANs in certain scenarios.
- **Generative Use Case:** In generative tasks, diffusion models are known for their ability to generate highly detailed, realistic images or videos [35]. Unlike GANs, which require fine-tuning and are prone to issues like mode collapse (where the generator produces limited types of outputs), diffusion models tend to produce more diverse and consistent outputs,

III. Methodology

A. Proposed Framework

Generative models have been at the forefront of AI advancements, capable of creating synthetic data—be it images, text, audio, or other forms. Below is an explanation of three major generative models: GPT (Generative Pretrained Transformers), Generative Adversarial Networks (GANs), and Diffusion Models.

1) GPT (Generative Pretrained Transformer)

GPT is a family of autoregressive transformers that have revolutionized natural language processing (NLP) by enabling models to generate coherent, contextually appropriate, and often human-like text. These models are based on the transformer architecture, which uses self-attention mechanisms to process data in parallel, making them highly efficient in capturing long-range dependencies and complex structures in the input.

- **Working:** GPT models are trained in two stages:
- **Pretraining:** In this stage, the model learns to predict the next word in a sentence given the previous words, using a massive corpus of text data. This helps the model develop a deep understanding of language patterns, grammar, and common knowledge.
- **Fine-Tuning:** After pretraining, GPT models are fine-tuned on specific tasks (such as text generation, translation, summarization, etc.) with smaller, task-specific datasets. This fine-tuning allows the model to adapt its general knowledge to perform better on particular NLP tasks.

2) Generative Adversarial Networks (GANs)

GANs are a class of generative models introduced by Ian Goodfellow in 2014. They consist of two neural networks, the generator and the discriminator, which are trained simultaneously in a competitive setting. The generator's goal is to create synthetic data that is indistinguishable from real data, while the discriminator tries to distinguish between real and fake data.

- **How It Works:**
- **Generator:** This network generates synthetic data (e.g., an image, a piece of text) starting from random noise. The generator is trained to improve its output so that the discriminator can no longer easily tell it apart from real data.
- **Discriminator:** This network's job is to classify whether the input data is real (from the dataset) or fake (generated by the generator). The discriminator provides feedback to the generator about how to improve.

These two networks are trained in a min-max game, where the generator tries to fool the discriminator, and the discriminator tries to correctly classify real vs. fake data. Over time, the generator improves its ability to create high-quality data that closely resembles the real data.

3) Diffusion Models

Diffusion models are a relatively newer class of generative models that have gained significant attention for their ability

to generate high-quality samples, especially in image generation tasks. Diffusion models operate by learning to reverse a noising process that progressively destroys information in data, such as images, until it is completely noise. The model then learns to reverse this noising process, step-by-step, to regenerate the data.

- How It Works:
- **Forward Process (Diffusion):** The data (e.g., an image) is progressively corrupted by adding random noise over multiple steps until it becomes pure noise. This process is designed to be a gradual, stochastic transformation.
- **Reverse Process (Denoising):** The model is trained to reverse this diffusion process. Starting from pure noise, the model learns to progressively "denoise" the data to reconstruct the original input. This process is learned through probabilistic models, typically using a neural network.

The model learns the likelihood of the original data at each step of the reverse diffusion, making it capable of generating new data samples from random noise.

B. SSL Pretext Tasks:

Self-Supervised Learning (SSL) has gained significant traction as an approach to training models without relying on labeled data [36]. The pretext tasks used in SSL help create supervisory signals from the data itself, allowing models to learn useful representations for downstream tasks. Below are detailed explanations of key SSL techniques commonly used in various domains, such as masked token/pixel prediction, contrastive learning, and feature clustering.

1) Masked Token or Pixel Prediction

Masked prediction tasks are one of the most widely used techniques in SSL, particularly in natural language processing (NLP) and computer vision.

a) For Text (Masked Token Prediction)

- **Working:** In this task, parts of the input text are randomly "masked" (replaced with a placeholder, such as the [MASK] token in BERT), and the model is tasked with predicting the masked tokens based on the surrounding context. The objective is to train the model to capture the structure and semantics of the language by learning how different parts of a sentence or paragraph are related.

Example: Consider the sentence "The cat sat on the [MASK]." The model must predict the masked word (e.g., "mat") by understanding the context provided by the surrounding words.

- **Applications:** Masked token prediction is the foundation for many language models, such as BERT (Bidirectional Encoder Representations from Transformers), which uses this task during its pretraining phase. It helps the model learn rich, contextual representations of words, which can then be fine-tuned for various NLP tasks such as classification, translation, and question answering.

b) For Images (Masked Pixel Prediction)

- **Working:** In computer vision, masked pixel prediction involves randomly masking out parts of an image (e.g., pixels, regions, or patches), and the model is trained to predict the missing information[37]. For example, parts of an image are replaced with noise or blanks, and the model learns to reconstruct the original image by leveraging the visible portions.

Example: For an image of a cat, a portion of the cat's face might be removed, and the model would predict the missing pixels or regions based on the remaining context.

- **Applications:** This technique is used in models like MAE (Masked Autoencoders for Images), which learn to recover missing portions of an image. Such tasks help the model learn semantic and spatial relationships within the image, enabling it to understand the structure of objects and scenes.

2) Contrastive Learning

Contrastive learning is a powerful SSL technique that aims to learn representations by comparing and contrasting similar and dissimilar data instances.

Working: In contrastive learning, the model learns to map similar data points closer in the feature space, while dissimilar data points are pushed farther apart. This is typically done by forming positive pairs (two similar instances) and negative pairs (two dissimilar instances). The model is then trained to minimize a contrastive loss function, which encourages it to pull the representations of positive pairs closer and push negative pairs apart.

- **Positive Pair:** These are instances that share certain similarities, such as two different augmentations of the same image or two translations of the same text.
- **Negative Pair:** These are instances that are different, such as a random image from a different class or a non-matching sentence in the case of text.
- **Loss Function:** A common loss function used in contrastive learning is the InfoNCE (Information Noise-Contrastive Estimation) loss, which quantifies how well the model distinguishes between positive and negative pairs. The model learns to maximize the similarity between positive pairs while minimizing it for negative pairs.

IV. Implementation and Result Analysis

Implementation of Self-Supervised Learning (SSL) for Generative AI, focusing on image reconstruction as a pretext task. The code demonstrates training a denoising autoencoder (DAE) for SSL using PyTorch. The DAE learns to reconstruct images from noisy inputs, which can later be fine-tuned for downstream generative or predictive tasks.

A. Implementation Key Steps

- **Data Preparation**: Use an image dataset (e.g., MNIST). Add noise to the images to create the self-supervised learning task.
- **Model Architecture**: Design a simple autoencoder with an encoder (latent representation) and decoder.
- **Training**: Train the model to minimize the reconstruction loss (mean squared error).
- **Visualization**: Display the original, noisy, and reconstructed images to evaluate the model.

```
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms
```

```
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt

# Define the Denoising Autoencoder model
class DenoisingAutoencoder(nn.Module):
    def __init__(self, input_dim=784, hidden_dim=128):
        super(DenoisingAutoencoder, self).__init__()
        # Encoder: Compress input to a latent space
        self.encoder = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
nn.ReLU()
        )
        # Decoder: Reconstruct input from latent space
        self.decoder = nn.Sequential(
            nn.Linear(hidden_dim, input_dim),
            nn.Sigmoid()
        )

    def forward(self, x):
        encoded = self.encoder(x)
        decoded = self.decoder(encoded)
        return decoded
# Prepare MNIST dataset
transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Lambda(lambda x: x.view(-1))  # Flatten the
image
])

train_dataset = datasets.MNIST(root="./data", train=True,
download=True, transform=transform)
train_loader = DataLoader(train_dataset, batch_size=64,
shuffle=True)

# Function to add noise to images
def add_noise(data, noise_factor=0.3):
    noisy_data = data + noise_factor * torch.randn_like(data)
    return torch.clamp(noisy_data, 0., 1.)

# Model, loss function, and optimizer
input_dim = 28 * 28  # Flattened MNIST images
hidden_dim = 128
model = DenoisingAutoencoder(input_dim=input_dim,
hidden_dim=hidden_dim)
criterion = nn.MSELoss()  # Reconstruction loss
optimizer = optim.Adam(model.parameters(), lr=1e-3)

# Training loop
epochs = 5
for epoch in range(epochs):
    model.train()
    epoch_loss = 0
    for data, _ in train_loader:
        noisy_data = add_noise(data)  # Add noise
        optimizer.zero_grad()
        outputs = model(noisy_data)
        loss = criterion(outputs, data)  # Compare output with
original
        loss.backward()
        optimizer.step()
        epoch_loss += loss.item()
    print(f"Epoch {epoch + 1}/{epochs}, Loss: {epoch_loss /
len(train_loader):.4f}")

# Save the encoder for future tasks
torch.save(model.encoder.state_dict(), "encoder_ssl.pth")

# Visualize the results
model.eval()
test_data, _ = next(iter(train_loader))
noisy_test_data = add_noise(test_data)

with torch.no_grad():
    reconstructed = model(noisy_test_data)

# Display original, noisy, and reconstructed images
n = 10  # Number of examples to display
fig, axes = plt.subplots(3, n, figsize=(15, 5))

for i in range(n):
    axes[0, i].imshow(test_data[i].view(28, 28).numpy(),
cmap="gray")
    axes[1, i].imshow(noisy_test_data[i].view(28,
28).numpy(), cmap="gray")
    axes[2, i].imshow(reconstructed[i].view(28, 28).numpy(),
cmap="gray")
axes[0, 0].set_ylabel("Original", fontsize=12)
axes[1, 0].set_ylabel("Noisy", fontsize=12)
axes[2, 0].set_ylabel("Reconstructed", fontsize=12)
for ax in axes.flatten():
    ax.axis("off")
plt.tight_layout()
plt.show()
```

Output

The output of codes show the process of execution the data set.

```
Epoch 1/5, Loss: 0.0412
Epoch 2/5, Loss: 0.0284
Epoch 3/5, Loss: 0.0250
Epoch 4/5, Loss: 0.0235
Epoch 5/5, Loss: 0.0221
```

Original Noisy Reconstructed		
Image (clean)	**Image (noisy)**	**Image (denoised)**
Clean digit '3'	Blurry digit '3'	Clear digit '3'
Clean digit '5'	Blurry digit '5'	Clear digit '5'
Clean digit '7'	Blurry digit '7'	Clear digit '7'
Clean digit '8'	Blurry digit '8'	Clear digit '8'
Clean digit '9'	Blurry digit '9'	Clear digit '9'
Clean digit '11'	Blurry digit '11'	Clear digit '11'
Clean digit '16'	Blurry digit '16'	Clear digit '16'
Clean digit '27'	Blurry digit '27'	Clear digit '27'

V. Conclusion

Self-Supervised Learning (SSL) has proven to be a transformative approach in generative AI, particularly in scenarios with limited labeled data. By leveraging unlabeled

data through pretext tasks such as denoising, SSL enables models to learn robust and meaningful representations without the need for manual annotations. This approach enhances model efficiency by utilizing abundant unlabelled data and improves adaptability by capturing essential data structures that are transferable to downstream tasks. SSL is especially valuable in limited data settings, as it allows models to generalize better with minimal labelled examples, making it a versatile and cost-effective solution for diverse applications. Overall, SSL stands out as a powerful strategy to advance generative AI, enabling better performance, scalability, and applicability across various domains.

REFERENCES

[1] J. Roe and M. Perkins, "Generative Ai In Self-Directed Learning : A Scoping Review," pp. 1–18, 2024.

[2] R. Balestriero *et al.*, "A Cookbook of Self-Supervised Learning," 2023.

[3] S. Amrale, "A Novel Generative AI-Based Approach for Robust Anomaly Identification in High- Dimensional Dataset," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 2, 2024, doi: 10.48175/IJARSCT-19900D.

[4] C. Patel, "Generative AI for Personalized Marketing and Customer Experience in E-Commerce," *Int. J. Emerg. Res. Eng. Technol.*, vol. 7, no. 1, pp. 12–19, 2026, doi: 10.63282/3050-922X.IJERET-V7I1P103.

[5] H. Buckchash, G. K. Verma, and D. K. Prasad, "Applications And Challenges Of Ai And Microscopy In Life Science Research: A Review," pp. 1–9, 2025.

[6] J. Pöppelbaum, G. S. Chadha, and A. Schwung, "Contrastive learning based self-supervised time-series analysis," *Appl. Soft Comput.*, vol. 117, p. 108397, 2022, doi: 10.1016/j.asoc.2021.108397.

[7] A. Polamarasetti, "Training Generative AI for Low-Resource Languages in Cloud Infrastructure," vol. 4, no. 4, pp. 190–202, 2024, doi: 10.56472/25832646/JETA-V4I4P125.

[8] S. K. Chintagunta, "Generative AI Approaches to Automated Unit Test Case Generation in Large-Scale Software Projects," *ESP J. Eng. Technol. Adv.*, vol. 4, no. 1, pp. 150–157, 2024, doi: 10.56472/25832646/JETA-V4I1P121.

[9] X. Zhang and L. Han, "A Generic Self-Supervised Learning (SSL) Framework for Representation Learning from Spectral–Spatial Features of Unlabeled Remote Sensing Imagery," *Remote Sens.*, vol. 15, no. 21, 2023, doi: 10.3390/rs15215238.

[10] L. S. Andra, S. K. Srivastava, S. Krishna, P. Siddana, and K. S. BuchiReddy, "Assessing User Confidence and Acceptance of AI Enhanced Through Multi-LLM Consensus Mechanisms," *Int. Trans. Electr. Eng. Comput. Sci.*, vol. 4, no. 4, 2025, doi: 10.62760/iteecs.4.4.2025.131.

[11] G. Xu, X. Jiang, X. Li, Z. Zhang, and X. Liu, "Exploring Self-Supervised Learning for Multi-Modal Remote Sensing Pre-Training via Asymmetric Attention Fusion," *Remote Sens.*, vol. 15, no. 24, 2023, doi: 10.3390/rs15245682.

[12] M. M. Abdulrazzaq *et al.*, "Consequential Advancements of Self-Supervised Learning (SSL) in Deep Learning Contexts," *Mathematics*, vol. 12, no. 5, 2024, doi: 10.3390/math12050758.

[13] S. Dodda, H. Volikatla, and J. R. Vummadi, "Exploring the Role of AI-Enhanced Chatbots in Automating Recruitment Processes in Human Capital Management Systems," *Int. J. Emerg. Trends Comput. Sci. Inf. Technol.*, vol. 6, no. 3, pp. 28–36, 2025, doi: 10.63282/3050-9246.IJETCSIT-V6I3P104.

[14] D. Chen, Z. Zhang, J. Liang, and L. Zhang, "SSL : A Self-similarity Loss for Improving Generative Image," 2024.

[15] X. Guo, Y. Chen, and S. Member, "Generative AI for Synthetic Data Generation : Methods , Challenges and the Future," pp. 1–8, 2024.

[16] S. Ayromlou, V. R. Khazaie, F. Forghani, and A. Afkanpour, "Can Generative Models Improve Self-Supervised Representation Learning?," 2023.

[17] G. Franceschelli, "Reinforcement Learning for Generative AI: State of the Art, Opportunities and Open Research Challenges," *arXiv*, pp. 1–30, 2024.

[18] V. Pal, "Bias Detection and Mitigation in Foundation AI Models : A Human-Centric Approach," *TIJER*, vol. 8, no. 2, pp. 4–10, 2021.

[19] A. Aghajanyan *et al.*, "Scaling Laws for Generative Mixed-Modal Language Models," 2023. doi: 10.48550/arXiv.2301.03728.

[20] N. Prajapati, "The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications," *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, 2025, doi: 10.56472/25832646/JETA-V5I2P103.

[21] S. Ben Atitallah, C. Ben Rabah, M. Driss, W. Boulila, and A. Koubaa, "Self-supervised learning for graph-structured data in healthcare applications: A comprehensive review," *Comput. Biol. Med.*, vol. 188, p. 109874, 2025, doi: 10.1016/j.compbiomed.2025.109874.

[22] J. Guo, X. Xu, and H. Zhao, "Self-supervised Learning for Enhancing Geometrical Modeling in 3D-Aware Generative Adversarial Network," vol. 1, 2023.

[23] T.-T.-T. Do, Q.-T. Huynh, K. Kim, and V.-Q. Nguyen, "A Survey on Video Big Data Analytics: Architecture, Technologies, and Open Research Challenges," *Appl. Sci.*, vol. 15, no. 14, p. 8089, Jul. 2025, doi: 10.3390/app15148089.

[24] L. Zhou and X. Yue, "Unifying Generative Self-Supervised Paradigms with Diffusion Models," in *Proceedings of the 7th ACM International Conference on Multimedia in Asia*, 2025. doi: 10.1145/3743093.3770999.

[25] Y. L. Tun, C. M. Thwal, H. Q. Le, M. N. H. Nguyen, E. Huh, and C. S. Hong, "Resource-efficient Layer-wise Federated Self-supervised Learning," *Arxiv J.*, 2025.

[26] H. P. Kapadia and K. B. Thakkar, "Generative AI for Real-Time Customer Support Content Creation," *J. Emerg. Technol. Innov. Res.*, vol. 10, no. 12, pp. 36–43, 2023.

[27] J. Yoo, L. Zhao, and L. Akoglu, *Self-Tuning Self-Supervised Image Anomaly Detection*, vol. 1, no. 1. arXiv, 2025. doi: 10.1145/3711896.3737122.

[28] K. Zhang *et al.*, "Generative Artificial Intelligence in Robotic Manipulation : A Survey," *arXiv*, pp. 1–27, 2025.

[29] A. Parupalli, "Business-Oriented Employee Performance Assessment via Machine Learning in ERP Systems," *TIJER – Int. Res. J.*, vol. 11, no. 11, 2024.

[30] B. Jeganathan, "Exploring the Power of Generative Adversarial Networks (GANs) for Image Generation: A Case Study on the MNIST Dataset," *Int. J. Adv. Eng. Manag.*, vol. 7, no. 1, pp. 21–46, Jan. 2025, doi: 10.35629/5252-07012146.

[31] S. Yang, B. Peng, T. Sui, M. Wu, and Q. Huang, "Advancing Self-Supervised Learning for Building Change Detection and Damage Assessment: Unified Denoising Autoencoder and Contrastive Learning Framework," *Remote Sens.*, vol. 17, no. 15, 2025, doi: 10.3390/rs17152717.

[32] Y. Macha, "A Review of Cloud-Based CRM Systems in Healthcare: Advances, Tools, Challenges, and Best Practices," *Int. J. Curr. Eng. Technol.*, vol. 12, no. 6, pp. 848–856, 2022, doi: 10.14741/ijcet/v.12.6.20.

[33] K. M. Risaduzzaman and S. Islam, "Generative AI in Practice : Pipeline Design , Implementation , and Ethical Considerations," 2025, doi: 10.36227/techrxiv.174495064.44742209/v1.

[34] F. Iqbal *et al.*, "Data Augmentation-based Novel Deep Learning Method for Deepfaked Images Detection," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 20, no. 11, Sep. 2024, doi: 10.1145/3592615.

[35] L. Gao, H. Fan, Q. Chen, and H. Yang, "Multi-Modal Self-Supervised Learning Algorithm-Based Product Recommendation," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 15371–15380, 2025, doi: 10.1109/TASE.2025.3568286.

[36] A. Zahran, A. Fahmy, K. Wassif, and H. Bayomi, "Fine-Tuning Self-Supervised Learning Models for End-to-End Pronunciation Scoring," *IEEE Access*, vol. PP, p. 1, 2023, doi: 10.1109/ACCESS.2023.3317236.

[37] I. Park, S. Kim, and J. Ryu, "Generative Self-Supervised Learning for Medical Image Classification," 2024.

