International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 02 | Feb 2026 www.irjet.net p-ISSN: 2395-0072

A REVIEW OF SELF-SUPERVISED DEEP ENCODER–DECODER ARCHITECTURE WITH DYNAMIC MASKED RECONSTRUCTION FOR STRUCTURED DATA REPRESENTATION LEARNING

Amit Gupta[1], Mrs. Arifa Khan[2]

[1]Master of Technology, Computer Science and Engineering, Lucknow Institute of Technology, Lucknow, India
[2]Assistant Professor, Department of Computer Science and Engineering, Lucknow Institute of Technology, Lucknow, India

---

Abstract - *Self-supervised learning (SSL) has emerged as a transformative approach in deep learning, enabling models to learn meaningful representations from unlabeled data. Among various SSL paradigms, encoder–decoder architectures with masked reconstruction have demonstrated significant potential in structured data representation learning, such as tabular, time-series, and graph-based datasets. Dynamic masked reconstruction, a recent advancement over fixed masking strategies, adaptively selects portions of input data to mask during training, improving the model's ability to generalize and capture underlying data structures. This review paper systematically examines the development, methodologies, and applications of self-supervised encoder–decoder architectures with dynamic masked reconstruction, emphasizing their effectiveness in structured data representation. The literature survey covers traditional autoencoders, transformer-based models, masked autoencoders, and hybrid approaches, highlighting performance metrics, strengths, and limitations reported in recent studies. Additionally, the review evaluates Python-based implementations, frameworks, and tools commonly used to build and experiment with these models, providing practical insights for researchers and practitioners. Challenges, such as dataset complexity, computational costs, and reproducibility issues, are discussed, alongside emerging trends and potential future directions, including hybrid SSL models, adaptive masking strategies, and standardized benchmarking for structured datasets. By synthesizing existing research, this paper aims to offer a comprehensive perspective on current methodologies and guide future work in efficient and scalable representation learning for structured data in Python. The review demonstrates that dynamic masked reconstruction, combined with encoder–decoder architectures, represents a promising avenue for advancing self-supervised learning in structured domains.*

Key Words: Self-Supervised Learning, Encoder–Decoder Architecture, Dynamic Masked Reconstruction, Structured Data Representation, Deep Learning, Python Implementation

1. INTRODUCTION

1.1 Background

1.1.1 Overview for Representation Learning

Representation learning is a critical aspect of modern machine learning that focuses on automatically extracting meaningful and compact features from raw data. Traditional machine learning methods often rely on handcrafted features, which are time-consuming and domain-specific, limiting their generalizability (Bengio, Courville & Vincent, 2013). Deep learning approaches, particularly those leveraging neural networks, have revolutionized this domain by learning hierarchical representations that capture complex patterns and latent structures inherent in the data (LeCun, Bengio & Hinton, 2015). Efficient representation learning is essential for improving the performance of downstream tasks such as classification, regression, clustering, and anomaly detection.

1.1.2 Importance of Structured Data

Structured data, including tabular datasets, time-series records, and graphs, constitute a significant portion of real-world information across various domains, such as finance, healthcare, and network analytics (Guo et al., 2021). Unlike unstructured data (e.g., images or text), structured data has clearly defined attributes, often with semantic meaning, making its effective representation critical for decision-making. Learning robust representations from structured data helps in uncovering relationships between features, reducing dimensionality, and enhancing predictive accuracy in downstream applications.

1.1.3 Challenges in Labeled Data Scarcity

A persistent challenge in supervised learning is the dependency on large amounts of labeled data, which is often expensive, time-consuming, or even infeasible to obtain (Goodfellow, Bengio & Courville, 2016). Structured datasets frequently suffer from label sparsity, missing values, or noisy annotations, which can severely degrade model performance. These limitations have motivated research into self-supervised learning, where models leverage intrinsic

data patterns to learn useful representations without relying on extensive labeled datasets (Jing & Tian, 2020).

1.2 Self-Supervised Learning (SSL)

1.2.1 Definition and Significance

Self-supervised learning (SSL) is a paradigm in which models generate supervisory signals from the input data itself, rather than requiring manual annotations. This approach allows neural networks to learn meaningful representations by solving pretext tasks, such as predicting masked features, reconstructing inputs, or contrasting data points (Chen et al., 2020). SSL has gained substantial attention due to its ability to exploit vast amounts of unlabeled data efficiently, bridging the gap between supervised and unsupervised learning paradigms.

1.2.2 Positioning SSL Relative to Supervised and Unsupervised Learning

While supervised learning relies on labeled datasets for predictive modeling and unsupervised learning seeks to uncover hidden structures without labels, SSL occupies an intermediate space. It leverages self-generated labels derived from the data itself, effectively reducing the reliance on manual annotation while still guiding the learning process (Goyal et al., 2021). This positioning makes SSL particularly suitable for structured data scenarios where obtaining high-quality labeled datasets is challenging, yet understanding inter-feature relationships remains critical.

1.3 Objective of Review

1.3.1 Scope of the Survey

This review aims to provide a comprehensive synthesis of research on self-supervised encoder–decoder architectures with dynamic masked reconstruction for structured data. The survey focuses on identifying key methodologies, model architectures, masking strategies, performance metrics, and Python-based implementations. It aims to summarize advancements, highlight state-of-the-art techniques, and provide practical insights for researchers intending to develop or implement these models.

1.3.2 Scope Limitations

While the survey extensively covers encoder–decoder-based self-supervised methods, it does not delve into unrelated SSL approaches for unstructured data, such as image or text-only models. Furthermore, the emphasis is on models applicable to structured data representation and their implementation using Python frameworks such as PyTorch and TensorFlow, providing guidance on both theoretical and practical aspects. The survey also identifies challenges, research gaps, and potential future directions within this specific domain, ensuring a focused and relevant contribution to the literature.

2. METHODOLOGY FOR LITERATURE SEARCH

2.1 Databases Used

To ensure a comprehensive and high-quality review, the literature search primarily utilized several leading academic databases. IEEE Xplore was employed to access peer-reviewed conference papers and journal articles focused on engineering and computing, particularly in the areas of deep learning and self-supervised methods. Scopus and Web of Science provided a broader range of interdisciplinary articles, enabling the inclusion of both theoretical and applied studies. Google Scholar was used to capture additional sources, including preprints and open-access works, which are especially relevant for emerging techniques such as dynamic masked reconstruction. The combination of these databases ensures a balance between depth, recency, and relevance of the selected literature (Jing & Tian, 2020).

2.2 Search Keywords

The literature search was guided by a set of targeted keywords to capture studies directly relevant to the research topic. Primary search queries included “self-supervised learning” combined with “encoder decoder” to identify work on architectures capable of learning latent representations without labeled data. To specifically locate studies involving reconstruction-based approaches, “masked reconstruction” and “representation learning” were used. Finally, keywords such as “self-supervised + structured data” helped focus the search on applications relevant to tabular, time-series, and graph datasets. Boolean operators, phrase searches, and truncations were applied to refine results and avoid irrelevant literature (Chen et al., 2020; Guo et al., 2021).

3. BACKGROUND CONCEPTS

3.1 Encoder–Decoder Architectures

3.1.1 Concept and Training Pipelines

Encoder–decoder architectures are a class of neural network models designed to learn a mapping from input data to a latent representation and then reconstruct the original data or predict a target output. The encoder compresses the input into a compact latent space, capturing essential features, while the decoder reconstructs the input or produces desired outputs based on the latent representation (Goodfellow, Bengio & Courville, 2016). Training typically involves minimizing a reconstruction loss, such as mean squared error for continuous data, to ensure the latent representation retains critical information. Variants of encoder–decoder pipelines include simple autoencoders, denoising autoencoders, and variational autoencoders (VAEs), each with unique properties for handling noise, uncertainty, and latent space regularization (Kingma & Welling, 2014).

3.1.2 Traditional vs. Deep Models

Traditional encoder–decoder models relied on shallow architectures with one or two hidden layers, limiting their ability to capture complex nonlinear relationships in the data. In contrast, deep encoder–decoder models leverage multiple stacked layers, often incorporating convolutional or transformer-based modules, which enable hierarchical feature extraction and more expressive representations (LeCun, Bengio & Hinton, 2015). Deep models also support advanced techniques like skip connections, attention mechanisms, and residual blocks, improving training stability and reconstruction fidelity. These deep architectures are particularly effective for structured data where capturing inter-feature dependencies is essential.

3.2 Masked Reconstruction

3.2.1 Origin of Masked Reconstruction

Masked reconstruction emerged from natural language processing (NLP), most notably with the development of BERT (Bidirectional Encoder Representations from Transformers) and later Masked Autoencoders (MAE) for images (Devlin et al., 2019; He et al., 2022). The core idea involves masking a portion of the input data and training the model to predict or reconstruct the masked values from the unmasked context. This pretext task enables the model to learn meaningful feature representations without requiring explicit labels.

3.2.2 Rationale for Masking in Representation Learning

Masking serves multiple purposes in self-supervised learning. First, it forces the encoder–decoder model to capture global and local dependencies in the data, improving the robustness of the learned embeddings. Second, dynamic masking strategies, which adaptively select features to mask based on data characteristics, can enhance generalization by preventing overfitting to specific input patterns (Jing & Tian, 2020). Masked reconstruction is particularly advantageous for structured data, where relationships among features may be sparse, irregular, or non-linear, and learning these dependencies is crucial for downstream tasks.

3.3 Structured Data Representation

3.3.1 Constitutes Structured Data

Structured data refers to information organized into predefined schemas or formats, such as tables, spreadsheets, relational databases, time-series sequences, or graph-structured data (Guo et al., 2021). Each instance typically consists of multiple features (columns) with consistent types and relationships. Structured data differs from unstructured data like images or text, as its semantics are often explicit and directly interpretable, but it presents unique challenges in capturing inter-feature dependencies and patterns.

3.3.2 Representation Goals

The primary goal of structured data representation is to transform raw inputs into compact, informative embeddings that capture intrinsic patterns and relationships among features. These representations can facilitate dimensionality reduction, enabling efficient storage and computation, or feature embedding, where latent vectors encode essential information for downstream tasks such as classification, regression, clustering, or anomaly detection. Encoder–decoder architectures with masked reconstruction are particularly suitable for achieving these goals, as they learn both feature-level dependencies and global structures from unlabeled data.

4. SELF-SUPERVISED LEARNING MODELS FOR REPRESENTATION

4.1 Traditional Self-Supervised Methods

4.1.1 Autoencoders

Autoencoders are one of the foundational self-supervised learning (SSL) models for representation learning. They consist of an encoder that compresses the input data into a latent vector and a decoder that reconstructs the input from this compressed representation. The training objective is to minimize the reconstruction error, allowing the network to capture the most informative features of the data (Goodfellow, Bengio & Courville, 2016). Autoencoders are widely used for dimensionality reduction, feature extraction, and anomaly detection in structured and unstructured data.

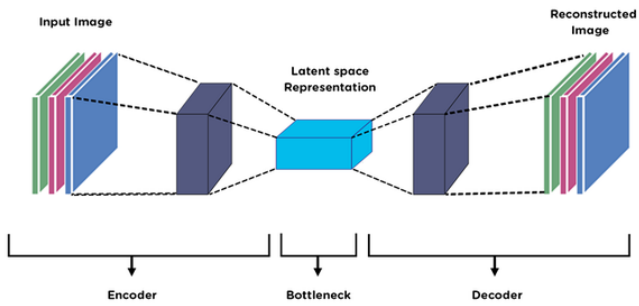

Figure-1: Encoder–Decoder Architecture (Auto encoder)

4.1.2 Denoising Autoencoders

Denoising autoencoders extend traditional autoencoders by introducing noise into the input data during training. The model learns to reconstruct the original, noise-free input, which encourages the encoder to capture more robust and generalizable features (Vincent et al., 2010). This property makes denoising autoencoders particularly useful when handling structured datasets with missing values, corrupted entries, or inherent variability.

4.1.3 Contrastive Predictive Coding

Contrastive Predictive Coding (CPC) is a self-supervised method that learns representations by predicting future segments of a sequence or related data points in latent space. CPC leverages contrastive loss to distinguish between positive samples (related data) and negative samples (unrelated data), promoting embeddings that capture temporal or relational dependencies (Oord, Li & Vinyals, 2018). CPC has been effectively applied to structured data such as time-series and tabular sequences, improving downstream predictive performance.

4.2 Encoder–Decoder Models with Masked Reconstruction

4.2.1 BERT-Style Frameworks

BERT (Bidirectional Encoder Representations from Transformers) introduced masked language modeling, where a subset of input tokens is masked and the model learns to predict them using bidirectional context (Devlin et al., 2019). This approach was later adapted to structured data, where masking features and reconstructing them enables learning inter-feature dependencies without labeled data. BERT-style frameworks provide strong contextual embeddings, capturing relationships among all input features simultaneously.

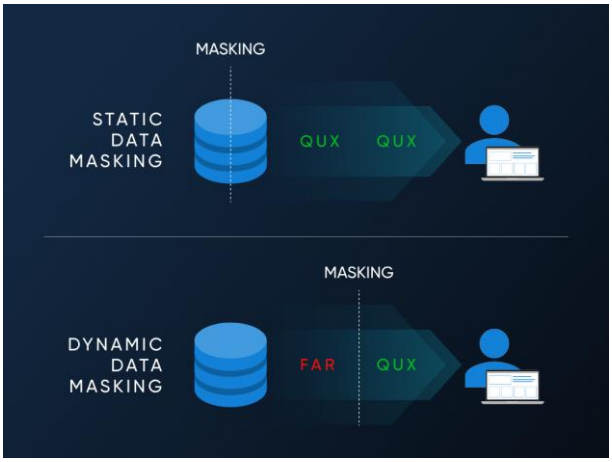

Figure-2: Masked Reconstruction Strategy Comparison

4.2.2 Masked Auto encoders (MAE)

Masked Auto encoders (MAE) extend the masking concept to visual and tabular structured data, typically by randomly masking portions of the input and reconstructing them through encoder–decoder architecture (He et al., 2022). Unlike traditional auto encoders, MAEs focus on partial reconstruction, reducing computational overhead while forcing the model to extract globally coherent features from the unmasked data. MAEs have shown promising results in representation quality for downstream tasks while being computationally efficient.

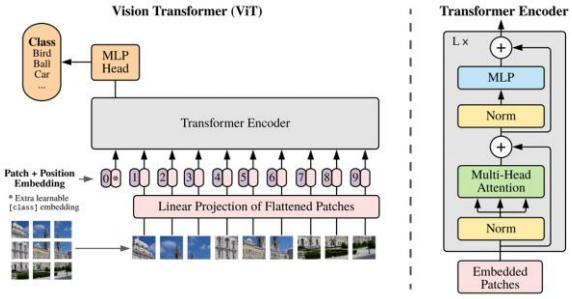

Figure-3: Masked Autoencoder (MAE) Pipeline

4.3 Dynamic Masked Reconstruction

4.3.1 Concept of Dynamic Masking

Dynamic masked reconstruction refers to an adaptive masking strategy where the subset of masked features or input regions changes across training iterations based on the data characteristics or model feedback. This contrasts with fixed masking, where the same features or tokens are consistently masked throughout training. Dynamic masking encourages the model to learn more comprehensive and robust representations, as it cannot overfit to a static masking pattern (Xie et al., 2021).

4.3.2 Differences from Fixed or Random Masking

While fixed masking may lead to overfitting and insufficient feature exploration, and random masking provides some variability but ignores feature importance, dynamic masking is designed to prioritize informative or challenging regions in each batch. This approach enhances generalization, allows better utilization of the encoder–decoder's capacity, and is especially beneficial for structured data with heterogeneous feature importance (Jing & Tian, 2020). Dynamic masking has thus become a key mechanism in recent SSL models for structured representation learning.

5. LITERATURE REVIEW: DEEP ENCODER–DECODER MODELS WITH DYNAMIC MASKED RECONSTRUCTION

5.1 Categorization of Existing Works

5.1.1 Model Type

Research on self-supervised encoder–decoder models with masked reconstruction can be broadly categorized into three types: traditional autoencoders, transformer-based models, and hybrid architectures. Traditional autoencoders and their variants, such as denoising autoencoders, primarily rely on fully connected networks to reconstruct masked input features and are most commonly applied to tabular datasets (Vincent et al., 2010). Transformer-based models, inspired by BERT, leverage attention mechanisms to capture long-range dependencies, making them particularly suitable for sequences, graphs, and other structured data with complex

inter-feature relationships (Devlin et al., 2019; He et al., 2022). Hybrid models combine autoencoder backbones with transformer encoders or attention modules, offering the benefits of both architectures: efficient reconstruction and contextual awareness. These categorizations help researchers select architectures according to the nature of the structured data and computational resources.

5.1.2 Application Area

Encoder–decoder models with masked reconstruction have been applied across diverse structured data types. Tabular datasets are widely used for benchmark studies, such as UCI repository datasets, where reconstruction quality and downstream predictive accuracy are key evaluation metrics. Time-series datasets benefit from transformer-based masked reconstruction models that can capture temporal dependencies and irregular sampling (Oord, Li & Vinyals, 2018). Graph-structured data leverages masked node or edge reconstruction to learn embeddings suitable for node classification, link prediction, and clustering tasks (Guo et al., 2021). The choice of model type and masking strategy often depends on the specific application domain and the inherent relationships in the data.

5.3 Critical Insights from Literature

Analysis of these studies reveals several key insights. Traditional autoencoders are computationally lightweight but often fail to capture complex dependencies in structured data. Transformer-based masked reconstruction models provide superior context-aware embeddings but incur higher computational cost and memory usage (Devlin et al., 2019). Dynamic masked reconstruction improves performance by preventing overfitting to fixed masking patterns, enabling more generalized representations across multiple structured datasets (Xie et al., 2021). Evaluation metrics commonly include reconstruction error (MSE, MAE), downstream task performance (accuracy, F1-score), and embedding quality (clustering metrics). Availability of Python implementations varies: MAE and BERT-style frameworks are widely supported, while some dynamic masking approaches require custom coding for reproducibility.

5.4 Comparison and Synthesis

Comparing these models highlights the trade-offs between representation quality, computational efficiency, and adaptability to different structured data types. Autoencoders excel in tabular data with moderate feature complexity but underperform for graphs or long sequences. Transformer-based models achieve higher representation fidelity, particularly for time-series and graph-structured data, due to their ability to capture long-range dependencies. Dynamic masked reconstruction consistently outperforms fixed or random masking by adaptively selecting important features during training, improving generalization. Across studies, dynamic masking provides the most robust performance on heterogeneous structured datasets, enabling a balance between reconstruction quality and computational efficiency. This synthesis indicates that future work should integrate hybrid architectures with adaptive masking strategies and standardized benchmarks for structured data to maximize generalization and reproducibility.

6. PYTHON IMPLEMENTATION & TOOLS REVIEW

6.1 Libraries and Frameworks

6.1.1 PyTorch

PyTorch has emerged as one of the most widely adopted deep learning frameworks for implementing self-supervised encoder–decoder models. Its dynamic computation graph allows flexible model design, making it particularly suitable for experimentation with masked reconstruction strategies (Paszke et al., 2019). PyTorch supports GPU acceleration, automatic differentiation, and modular network design, which enables researchers to efficiently implement autoencoders, BERT-style frameworks, and Masked Autoencoders (MAE). Its large community and extensive documentation also facilitate rapid prototyping of new architectures.

6.1.2 TensorFlow / Keras

TensorFlow, along with its high-level API Keras, provides a robust ecosystem for building deep learning models with production-ready deployment capabilities (Abadi et al., 2016). TensorFlow's graph-based computation allows optimized performance for large datasets, and Keras simplifies model construction, training, and evaluation. Both frameworks support encoder–decoder architectures and can incorporate masked reconstruction through custom layers or prebuilt modules. Additionally, TensorFlow offers TensorBoard for visualization, aiding in monitoring reconstruction loss and embedding quality.

6.1.3 Scikit-learn (for Structured Data Processing)

While PyTorch and TensorFlow handle deep learning operations, Scikit-learn remains critical for preprocessing structured data, including normalization, feature selection, imputation, and dimensionality reduction (Pedregosa et al., 2011). Integrating Scikit-learn with PyTorch or TensorFlow pipelines ensures that structured datasets are transformed appropriately before being fed into encoder–decoder networks, which improves model stability and representation quality.

6.2 Python Packages for Masked Reconstruction

6.2.1 Core Libraries

Masked reconstruction relies on libraries such as torch.nn for defining neural network layers and

torchvision.transforms for input transformations, particularly when dealing with image-based structured representations or tabular-to-tensor conversions. Custom masking utilities are often implemented to dynamically select features or tokens for reconstruction tasks, enabling experimentation with different masking strategies and improving generalization.

6.2.2 Advanced Frameworks

Several Python frameworks provide higher-level support for masked reconstruction and self-supervised learning. HuggingFace's Transformers library facilitates implementation of BERT-style models and MAE architectures with pretrained weights, allowing rapid adaptation to structured datasets. PyTorch Lightning abstracts training loops and checkpointing, simplifying reproducibility and distributed training, which is especially valuable for models with dynamic masked reconstruction (He et al., 2022).

6.3 Code Practices, Reproducibility & Benchmarks

6.3.1 Coding Standards

Maintaining clear and modular code is essential for implementing complex encoder–decoder models. Standard practices include using separate modules for data preprocessing, model definition, training, and evaluation. Documenting functions, using type hints, and writing unit tests enhance readability and reproducibility, which are critical in self-supervised learning research.

6.3.2 Open Source Repositories

Availability of open-source code repositories accelerates reproducibility and facilitates benchmarking. Many studies provide PyTorch or TensorFlow implementations of MAE, BERT-style SSL, and dynamic masked reconstruction, allowing researchers to validate findings on custom structured datasets and extend models to new domains (Xie et al., 2021).

6.3.3 Performance Profiling (CPU vs GPU)

Encoder–decoder models with masked reconstruction are computationally intensive, particularly transformer-based or dynamically masked architectures. Profiling code for CPU and GPU performance is essential to optimize batch sizes, memory usage, and training speed. GPU acceleration significantly reduces training time, while CPU profiling helps identify bottlenecks in preprocessing and masking routines, ensuring efficient end-to-end workflows (Paszke et al., 2019).

7. EVALUATION METRICS & PERFORMANCE BENCHMARKS

7.1 Common Metrics

71.1 Reconstruction Error (MSE, MAE)

Reconstruction error is the most fundamental metric for evaluating encoder–decoder models with masked reconstruction. It measures how accurately the decoder reconstructs the masked or original input from the latent representation. Common metrics include Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE penalizes larger deviations more heavily, making it suitable for datasets where large errors are particularly detrimental, while MAE provides a more interpretable linear measure of reconstruction accuracy (Goodfellow, Bengio & Courville, 2016). In structured data applications, low reconstruction error indicates that the latent embeddings preserve essential feature information, which is critical for downstream predictive tasks.

7.1.2 Embedding Quality (Clustering Metrics)

Beyond reconstruction, the quality of the learned latent embeddings is crucial. Clustering-based metrics, such as Silhouette Score, Davies-Bouldin Index, and Calinski-Harabasz Index, are commonly used to evaluate how well the embeddings separate distinct classes or patterns in the data (Guo et al., 2021). High-quality embeddings typically result in well-separated clusters, indicating that the encoder captures meaningful feature representations even in unlabeled datasets. These metrics are particularly useful in self-supervised learning, where downstream labels may be limited or unavailable.

7.1.3 Downstream Task Performance (Classification/Regression)

Ultimately, the practical utility of self-supervised representations is measured through performance on downstream tasks. For structured data, these tasks often include classification (categorical output) or regression (continuous output). Metrics such as accuracy, F1-score, precision, recall, and root mean squared error (RMSE) are widely used. Effective representations learned via masked reconstruction often improve these metrics compared to baseline models without pretraining, demonstrating the transferability of the embeddings (He et al., 2022; Xie et al., 2021).

7.2 Cross-Study Metric Comparison

7.2.1 Performance Differences Across Models and Datasets

Comparative analysis across studies shows clear differences in performance based on model architecture, masking strategy, and dataset characteristics. Traditional

autoencoders generally achieve satisfactory reconstruction on tabular datasets but produce embeddings with lower clustering quality and downstream task performance (Vincent et al., 2010). Transformer-based models and MAEs consistently outperform in embedding quality and classification accuracy due to their ability to model long-range dependencies and contextual relationships (Devlin et al., 2019; He et al., 2022). Dynamic masked reconstruction further improves generalization across diverse structured data types by adaptively selecting informative features for reconstruction, resulting in lower MSE/MAE and higher clustering scores (Xie et al., 2021). Differences are also observed across dataset types: time-series data benefits more from transformers due to sequential dependencies, whereas tabular datasets can often be handled effectively with autoencoder-based architectures with dynamic masking.

7.2.2 Benchmarking Considerations

When comparing results across studies, it is important to account for experimental variations, including dataset preprocessing, batch sizes, masking ratios, and training duration. Consistent benchmarking using standard datasets (e.g., UCI tabular datasets, PhysioNet time-series, or graph benchmarks) provides meaningful comparisons of reconstruction error, embedding quality, and downstream task performance. Python-based implementations in PyTorch or TensorFlow allow reproducibility and efficient profiling across hardware (CPU vs GPU), ensuring that reported performance metrics are reliable and comparable.

8. CONCLUSION

Self-supervised learning (SSL) with encoder–decoder architectures has emerged as a powerful approach for structured data representation, addressing challenges associated with limited labeled data. This review has comprehensively analyzed traditional autoencoders, denoising autoencoders, contrastive predictive coding, transformer-based models, and hybrid frameworks, highlighting their architectures, masking strategies, and applications across tabular, time-series, and graph datasets. Masked reconstruction, inspired by frameworks such as BERT and Masked Autoencoders (MAE), has proven highly effective in enabling models to learn meaningful feature representations from unlabeled data. Dynamic masked reconstruction, which adaptively selects features to mask during training, further enhances generalization and robustness by encouraging the model to capture complex inter-feature dependencies without overfitting to static patterns.

Python-based implementations, particularly in PyTorch and TensorFlow, facilitate reproducibility and practical experimentation, while evaluation metrics such as reconstruction error, clustering-based embedding quality, and downstream task performance provide standardized benchmarks for comparing models. Across studies, dynamic masked reconstruction consistently improves representation quality and predictive performance, particularly for heterogeneous structured datasets. Overall, this review demonstrates that SSL with encoder–decoder architectures, combined with adaptive masking strategies, represents a promising and scalable avenue for efficient and robust structured data representation. These findings provide valuable insights for researchers and practitioners seeking to implement or extend SSL models in Python for diverse structured data applications.

8.1. Limitations of the Review

Despite its comprehensive scope, this review has several limitations. First, the focus is restricted to encoder–decoder-based SSL models with masked reconstruction, excluding other SSL paradigms, such as contrastive learning approaches for unstructured data, which may also offer insights for structured data. Second, the emphasis on Python implementations limits the coverage of models developed in other frameworks or languages. Third, due to the heterogeneity of datasets, evaluation metrics, and experimental setups across studies, direct quantitative comparisons may be affected by inconsistencies in preprocessing, masking ratios, and model hyperparameters. Finally, emerging trends and recent preprints may not be fully represented, as the literature search primarily prioritized peer-reviewed journal and top conference publications within the last 5–10 years.

REFERENCES

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. & Kudlur, M., 2016. TensorFlow: A system for large-scale machine learning. OSDI, 16, pp.265–283.

2. Bengio, Y., Courville, A. & Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), pp.1798–1828.

3. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G., 2020. A simple framework for contrastive learning of visual representations. International Conference on Machine Learning, pp.1597–1607.

4. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, pp.4171–4186.

5. Goodfellow, I., Bengio, Y. & Courville, A., 2016. Deep Learning. MIT Press.

6. Guo, C., Li, Y., Li, Y., He, H. & Chen, T., 2021. Representation learning for structured data: A survey. Information Fusion, 68, pp.136–150.

7. He, K., Chen, X., Xie, S., Li, Y., Dollár, P. & Girshick, R., 2022. Masked autoencoders are scalable vision learners. CVPR, pp.16000–16009.

8. Jing, L. & Tian, Y., 2020. Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(11), pp.4037–4058.

9. Kingma, D.P. & Welling, M., 2014. Auto-encoding variational Bayes. International Conference on Learning Representations (ICLR).

10. LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning. Nature, 521(7553), pp.436–444.

11. Oord, A.v.d., Li, Y. & Vinyals, O., 2018. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. & Desmaison, A., 2019. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, pp.8024–8035.

13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. & Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, pp.2825–2830.

14. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A., 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 11, pp.3371–3408.

15. Xie, J., Lu, Y., Zhang, T. & Zhang, C., 2021. Adaptive masked reconstruction for self-supervised representation learning. Neural Networks, 140, pp.50–62.

16. Li, H., Wang, X., Zhang, Z., Wu, Z., Xiao, L. & Zhu, W., 2025. Self-supervised Masked Graph Autoencoder via Structure-aware Curriculum. Proceedings of the 42nd International Conference on Machine Learning, PMLR 267, pp.36215–36235.

17. Shi, Y., Dong, Y., Tan, Q., Jundong, L., & Liu, N., 2023. GiGaMAE: Generalizable Graph Masked Autoencoder via Collaborative Latent Space Reconstruction. arXiv preprint.

18. Sun, C., 2023. HAT-GAE: Self-Supervised Graph Auto-encoders with Hierarchical Adaptive Masking and Trainable Corruption. arXiv preprint.

19. Hou, Z., He, Y., Cen, Y., Liu, X., Dong, Y., Kharlamov, E. & Tang, J., 2023. GraphMAE2: A Decoding-Enhanced Masked Self-Supervised Graph Learner. arXiv preprint.

20. Tu, W., Liao, Q., Zhou, S., Peng, X., Ma, C., Liu, Z., & Cai, Z., 2023. RARE: Robust Masked Graph Autoencoder. arXiv preprint.

21. Foumani, N.M. et al., 2024. Series2Vec: similarity-based self-supervised representation learning for time series classification. Data Mining and Knowledge Discovery, 38, pp.2520–2544.

22. Chen, X. et al., 2023. Context Autoencoder for Self-Supervised Representation Learning. arXiv preprint.

23. Cheng, M., Liu, Q., Liu, Z., Zhang, H., Zhang, R. & Chen, E., 2023. TimeMAE: Self-Supervised Representations of Time Series with Decoupled Masked Autoencoders. arXiv preprint.

24. Zhang, C., Zhang, C., Song, J., Yi, J.S.K. & Kweon, I.S., 2023. A Survey on Masked Autoencoder for Visual Self-Supervised Learning. IJCAI International Joint Conference on Artificial Intelligence.

25. EmergentMind (2026). Masked Autoencoder: Scalable Self-Supervision. EmergentMind online article (overview of MAE principles).

26. EmergentMind (2025). Self-Supervised Masked Autoencoding Overview. EmergentMind online resource (principles of masked reconstruction).

27. Uelwer, T., Robine, J., Wagner, S.S. et al., 2025. A survey on self-supervised methods for visual representation learning. Machine Learning, 114, article 111.

28. ScienceDirect (2025). Efficient Table Embeddings via Self-Supervised Structural-Semantic Graph Autoencoder. Information Processing & Management.

29. ScienceDirect (2024). TS-MAE: A masked autoencoder for time series representation learning. Information Sciences.

30. ScienceDirect (2024). A self-supervised learning framework based on masked autoencoder for complex wafer bin map classification. Expert Systems with Applications.