POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

A Generative Algorithm for Topological Design in Facility Layout Planning via Guided Discrete Diffusion

Laurea Magistrale in Mechanical Engineering - Ingegneria Meccanica

Author: Damiano Marcon

Advisor: Prof. Andrea Matta

Academic year: 2024-2025

1. Introduction

The transition toward Industry 5.0 is redefining manufacturing, demanding systems characterized by high resilience and adaptivity. In this context, Facility Layout Planning (FLP) is the cornerstone of operational flexibility. This thesis targets the "topological design phase", establishing the logical interconnections and material flows between production resources. This stage is critical: while physical placement allows for adjustments, a flawed logical architecture creates permanent bottlenecks that no degree of geometric optimization can resolve.

Traditionally, topological design has relied on combinatorial optimization, utilizing Exact Methods (e.g., MIP) or Meta-heuristics (e.g., Genetic Algorithms). While effective for small-scale static problems, these approaches struggle with the high-dimensional, discrete complexity of modern flexible systems and often demand manual, time-consuming objective definitions. Recently, Deep Generative Models, such as VAEs and GANs, have emerged as alternatives. However, when applied to industrial graphs, these models frequently fail to respect strict engineering constraints, producing layouts that are technically non-functional despite appearing structurally valid.

To bridge this gap, a Discrete Denoising Diffusion Probabilistic Model (D3PM) for industrial graph generation is introduced [3]. Unlike continuous diffusion variants, this approach operates directly on categorical data, leveraging a backbone centered on the TransformerConv operator to capture intricate relational dependencies between workstations. The core innovation lies in the integration of Classifier-Free Guidance (CFG), which steers the generation toward high-performance configurations, and a Logit-Guided Repair mechanism. This dual strategy ensures that the generated layouts are not only optimized for performance but also strictly compliant with industrial logic. By shifting from selection to synthesis, this framework provides a robust tool for the automated design of next-generation manufacturing backbones.

2. Problem Formulation

The topological design is modeled as a directed graph generation task $G(V, E)$, where nodes V represent functional resources and edges E define material flow paths.

Graph-Based Representation. Resources are classified into four types: Machines (0), Buffers (1), Assembly (2), and Disassembly (3), each defined by specific operational attributes.

The layout is encoded as a categorical node vector $V \in \{0,1,2,3\}^N$ and a binary adjacency matrix $E \in \{0,1\}^{N\times N}$, where N is the number of nodes and the binary values represent the presence or absence of a connection. To guarantee coherent material flow, the model enforces local constraints by explicitly prohibiting self-loops and dead-ends, ensuring each resource adheres to mandatory connectivity rules, such as the required interfacing between machines and buffers.

System Typologies. The framework supports four distinct configurations: Open, Closed, Input-Only, and Output-Only. To ensure operational feasibility, the global topology must satisfy specific reachability requirements: Open Systems require a continuous path from source to sink, while Closed Systems must form Strongly Connected Components (SCCs) to enable the indefinite recirculation of materials.

Optimization Objectives. The generation is steered toward high-performing configurations by balancing two competing KPIs: System Throughput (X) and Total Energy Consumption (E). Evaluated through Discrete Event Simulation (DES), these indicators are synthesized into a scalar score S using user-defined weights, w_X and w_E. This weighting allows the framework to navigate industrial trade-offs, prioritizing either productivity or sustainability to optimize the target performance manifold.

3. Methodology: Guided Discrete Diffusion

This framework employs a Discrete Denoising Diffusion Probabilistic Model (D3PM) to synthesize facility layouts. It operates by learning to invert a Markovian forward process $p(x_t|x_{t-1})$ that gradually degrades data into maximum entropy. During training, the model approximates the reverse transitions $p_\theta(x_{t-1}|x_t)$ to extract underlying structural rules. Once trained, the Generative Pipeline leverages these learned transitions to reconstruct functional architectures starting from pure random noise.

3.1. Training Framework

The training phase aims to master the reconstruction of manufacturing graphs through a hierarchical process organized into epochs, batches, and items. While individual loss signals are calculated at the item level, they are aggregated within each batch to compute a stable gradient. A complete pass (epoch) concludes with a dataset reshuffle, ensuring the model learns underlying generative rules rather than memorizing fixed sequences. Consequently, the total number of parameter updates is defined by the product of the number of batches processed per epoch and the number of training epochs.

The pipeline consists of six fundamental stages (Fig. 1):

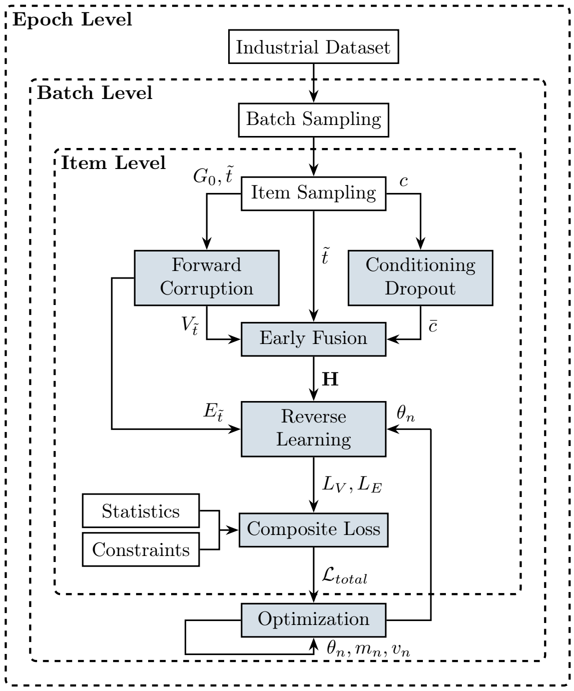

Figure 1: Procedural architecture of the Training Framework.

Forward Corruption Process This entropic phase degrades the initial graph $G_0^{(k)}(V_0^{(k)}, E_0^{(k)})$ into a noisy state $G_{\tilde{t}}^{(k)}(V_{\tilde{t}}^{(k)}, E_{\tilde{t}}^{(k)})$ via a Cosine Noise Schedule according to a timestep which is sampled uniformly across the diffusion horizon ($\tilde{t}^{(k)} \sim \mathcal{U}(0,T)$) [2]. A retention probability $\alpha_{\tilde{t}}^{(k)}$ dictates if an element maintains its original state or is replaced by a value sampled uniformly from its categorical space ($\mathcal{C} \in \{0,\ldots,3\}$ for nodes; $\mathcal{C} \in \{0,1\}$ for edges). This mechanism generates the corrupted components $V_{\tilde{t}}^{(k)}$ and $E_{\tilde{t}}^{(k)}$ that serve as the noisy input for the reverse learning phase.

Conditioning Dropout Strategy The model operates in both conditional and unconditional regimes to support Classifier-Free Guidance (CFG) [1], allowing it to "steer" the generation by interpolating between the two learned distributions. This is implemented via Conditioning Dropout: during training, the performance score $c^{(k)}$ is replaced by a null sentinel $\emptyset$ with a probability p_{uncond} or retained. The resulting vector $\bar{c}^{(k)}$ decouples baseline structural learning from performance-driven synthesis, providing the model with the flexibility to function as a goal-oriented design tool that amplifies features aligned with specific targets.

Early Fusion Strategy This stage integrates the noisy node states $V_{\tilde{t}}^{(k)}$ with the global parameters $\tilde{t}^{(k)}$ and $\bar{c}^{(k)}$, projecting scalar information into a high-dimensional space. The resulting unified tensor $\mathbf{H}^{(k)}$ encapsulates both local workstation identities and global objectives, providing the latent depth required for the Transformer-based layers to maintain a holistic understanding of the layout's state.

Reverse Learning Process The reverse transition mapping processes the unified embedding $\mathbf{H}^{(k)}$ and the corrupted edge matrix $E_{\tilde{t}}^{(k)}$ to compute reconstruction logits $L_V^{(k)}$ and $L_E^{(k)}$. Utilizing a TransformerConv backbone, the model leverages multi-head attention governed by learnable weights θ to capture complex relational dependencies. These high-dimensional features are then mapped to obtain the output logits, which quantify the model's confidence in recovering the initial categorical states. This architecture provides the probabilistic foundation for the subsequent loss computation.

Composite Loss Function The total loss $\mathcal{L}_{total}$ is evaluated directly from the logits L_V and L_E. This metric is a weighted summation of categorical reconstruction accuracy, regularization, and semantic constraints penalizing industrial rule violations. This composite signal serves as the objective function for the subsequent optimization of the learnable weights θ, ensuring both mathematical fidelity and engineering feasibility.

Optimization Strategy In the final stage, loss signals are aggregated into mini-batches to ensure a stable gradient direction. The Adam optimizer utilizes this aggregated signal to update the current model weights from θ_n to θ_{n+1}. By computing the first and second moments of the gradients - m and v respectively — the algorithm adaptively adjusts the learning rate, leveraging both historical information and the current loss to ensure efficient convergence toward a state capable of reconstructing complex industrial topologies.

3.2. Generative Pipeline

The generative pipeline serves as the inference engine of the framework, translating high-level user requirements into functional industrial layouts. As illustrated in Figure 2, the process is an iterative refinement sequence that transforms an initial state of maximum entropy into a structured topology. Starting from a completely noisy graph $G_T(V_T, E_T)$, the pipeline systematically reverses the corruption process to yield a final output $\hat{G}(V, \hat{E})$ that is both structurally valid and performance-optimized.

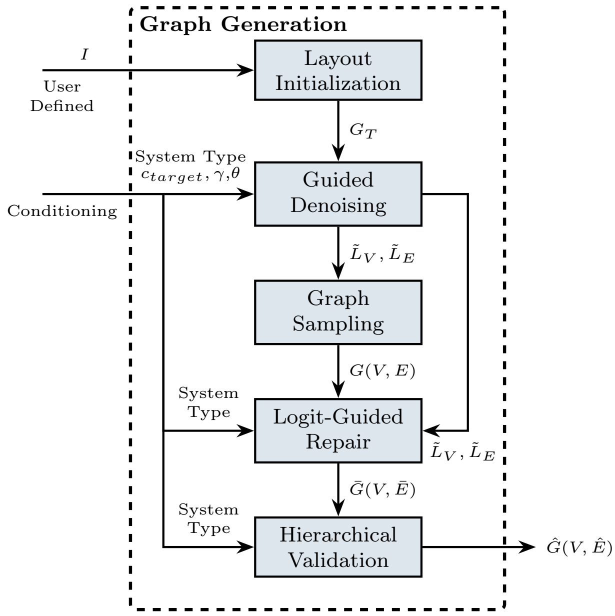

Figure 2: Procedural architecture of the Generative Pipeline.

Layout Initialization The process begins by defining the system inventory I, specifying the required node count for each category. From this, the starting graph G_T is initialized: the node vector V_T is a fixed permutation of these units, while the adjacency matrix E_T is initialized as pure categorical noise (maximum entropy). This configuration serves as the stochastic seed, ensuring that while resource quantities are predefined, their topological interconnections are synthesized from scratch.

Guided Denoising During this phase, the model processes the noisy graph $G_T(V_T, E_T)$ according to the target performance scores c_{target}, the guidance scale γ, the System Type, and the learned weights θ. Based on the defined architecture, the model generates two distinct predictions: a conditional prediction L_{cond}, guided by the input target scores, and an unconditional prediction L_{uncond}, where the Score information is masked. These are synthesized through CFG to obtain the guided logits $\hat{L}$ as follows:

$$\hat{L} = L_{uncond} + \gamma \cdot (L_{cond} - L_{uncond})$$

These raw logits $\hat{L}$ represent the model's confidence in the proposed structure. Immediately following this step, a Projector is applied to ensure that the predicted distributions do not violate the fundamental structural rules or logical constraints required by the specific industrial configuration. The final output of this stage consists of the corrected logits, denoted as $\tilde{L}_V$ and $\tilde{L}_E$, which provide the refined probabilistic foundation for the final graph generation.

Graph Sampling From the corrected logits $\tilde{L}_V$ and $\tilde{L}_E$, the final structure is derived using categorical sampling. The Softmax function converts raw confidence scores into probability distributions, from which the graph $G(V, E)$ is sampled, representing the model's primary realization of the industrial layout.

Logit Guided Repair Strategy To resolve potential topological inconsistencies, this mechanism leverages the confidence levels in $\tilde{L}_V$ and $\tilde{L}_E$. The algorithm first performs topological pruning, removing connections that violate flow logic or conflict with the predicted categorical structure. Furthermore, for closed-loop systems, a connectivity restoration routine follows, strategically creating edges to ensure Strongly Connected Components (SCCs). This transforms the probabilistic output into a robust structural skeleton $\bar{G}(V, \bar{E})$.

Hierarchical Validation In the final stage of the pipeline, the structure undergoes a rigorous auditing sequence. This phase consists of a multi-level validation hierarchy designed to ensure full compliance with all imposed constraints, spanning from basic structural integrity to complex industrial semantic rules. By verifying that the generated layout respects both topological requirements and domain-specific logic, this process guarantees that the final output is not merely a mathematically valid graph, but a technically sound and implementable manufacturing backbone, denoted as $\hat{G}(V, \hat{E})$.

3.3. Experimental Setup

Technical Metrics The algorithm is evaluated through technical metrics designed to quantify generative fidelity and adherence to constraints:

- Validity Rate: the percentage of generated graphs satisfying all topological and operational constraints.
- Uniqueness: the ratio of topologically distinct layouts (verified via Weisfeiler-Lehman hashing) to detect mode collapse.
- Novelty: the proportion of valid graphs not present in the training set, distinguishing structural synthesis from memorization.
- MAD Score: the Mean Absolute Deviation between the requested target c_{target} and the actual score, measuring the precision of the CFG steering.

Hyperparameter Configuration The model hyperparameters were defined following a dual approach. Most architectural settings were established based on theoretical principles and literature benchmarks. In contrast, the optimal values for the number of epochs and batch size were identified through an empirical study using a 3^k factorial design, ensuring a systematic analysis of the trade-offs between convergence stability and training efficiency.

4. Experimental Results

The experimental evaluation tests the framework's ability to synthesize structurally valid and operationally efficient industrial layouts. By comparing guided generation against unconditioned baselines, the analysis shows how the model internalizes topological rules and navigates engineering trade-offs. Technical metrics are integrated to quantify the algorithm's internal logic, ensuring it generalizes beyond the training data to explore the feasible design manifold rather than merely memorizing it.

4.1. Results per System-Type

The framework was tested on four topologies: Open, Closed, Input-Only, and Output-Only. Evaluation focused predominantly on Throughput ($w_X = 0.75$), with a secondary weight on Energy Consumption ($w_E = 0.25$).

The Open System serves as the primary benchmark due to its computational efficiency. As shown in Figure 3, the guidance mechanism shifts the distribution from the low-intensity baseline toward the high-performance frontier. The analysis of convex hulls and iso-score lines reveals that the model acts as a selective filter, suppressing "inefficient complexity"—where high energy consumption yields low throughput—to prioritize optimal layouts. This confirms that the framework has successfully internalized the engineering trade-offs defined by the objective function S.

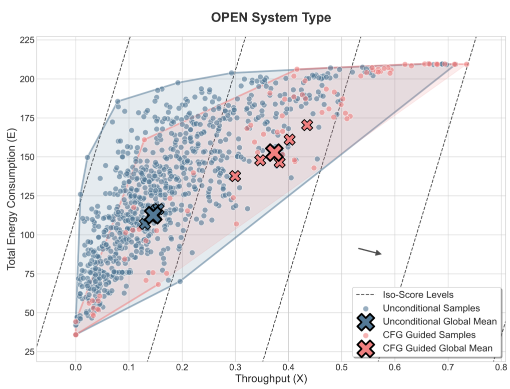

Figure 3: Performance distribution in Throughput-Energy space.

Statistical and Technical Validation Quantitative results confirm a significant performance optimization:

- Throughput (X): the CFG achieved a 156.10% mean increase over the baseline, rising from 0.146 to 0.373 units/time.
- Energy Consumption (E): a mean variation of 35.63% (152.74 vs 112.62 units) reflects the increased operational intensity required to reach high-performance targets.
- Generative Quality: the Validity Rate of 65.32% ensures structural integrity, while the Uniqueness of 33.60% reflects the model's focus on a specific set of high-performing topological patterns. Additionally, a Novelty score of 58.80% confirms the model is synthesizing original layouts rather than memorizing the training set.

A representative example of the synthesized architectures is illustrated in Figure 4, showcasing how the model translates these statistical gains into a functional layout.

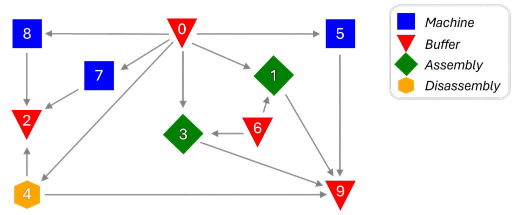

Figure 4: Representative example of a high-performance layout generated for the Open System topology.

4.2. Ablation Study

Ablation experiments isolate the individual contributions of architectural components to system performance, structural validity, and generative fidelity.

Projector & Repairer Ablation The impact of constraint-handling modules depends heavily on the underlying topology, shifting the Repairer's role from passive pruning to active generation (in Closed systems). In linear flows (Open, Input/Output-Only), the Strict Projector acts as the primary driver of validity; its deactivation causes the Validity Rate to collapse from 70.4% to a negligible 0.7%, rendering the Logit-Guided Repairer redundant as it cannot synthesize global flow logic from scratch.

Conversely, cyclic systems reveal a fundamental synergy: the Projector establishes the global circular skeleton while the Repairer resolves local degree conflicts. This collaboration in the Full Pipeline yields a 12.6% Validity Rate, more than doubling the Projector-only configuration (4.8%). Without the Projector, the Repairer fails almost entirely (0.6% validity), proving that while local fine-tuning is essential for cycles, it cannot compensate for the absence of global structural guidance during the diffusion process.

Guidance Scale Impact(γ) As shown in Fig. 5, the Guidance Scale γ controls the steering intensity toward performance targets.

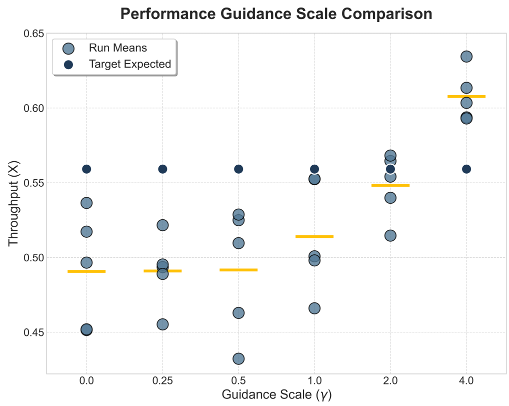

Figure 5: Performance across tested Guidance Scale (γ) levels.

At low values ($\gamma \leq 0.5$), high variance and inefficient outliers persist. Increasing γ toward 2.0 tightens the distribution and aligns the mean throughput (yellow line) with the target. At $\gamma = 4.0$, a saturation effect occurs: while MAD drops due to outlier suppression, a systematic bias toward over-performance emerges. This gain in precision and Validity Rate (exceeding 80%) triggers a trade-off, where Novelty and Uniqueness drop sharply as the model converges on "safe", repetitive patterns.
The analysis is based on the means of 5 independently generated sets.

Target Score Impact (c_{target}) As illustrated in Fig.6, the Target Score c_{target} acts as the primary control signal for productivity standards.

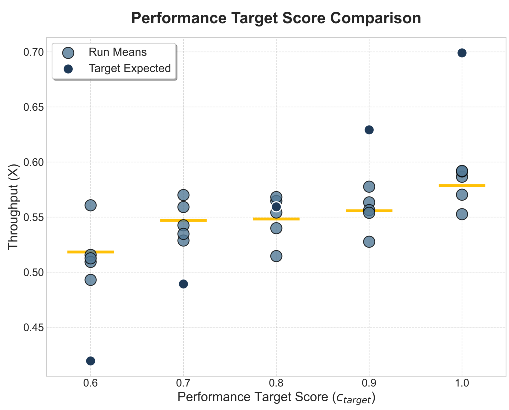

Figure 6: Performance across tested Target Score (c_{target}) levels.

The output follows the requested trend between 0.6 and 1.0, despite a "lag" relative to the theoretical maximum. While accuracy (MAD) improves up to $c_{target} = 0.9$, it degrades at the 1.0 threshold as the system reaches its functional limit. This performance push maintains structural validity but significantly reduces Novelty and Uniqueness; to satisfy stringent constraints, the model sacrifices combinatorial variety by concentrating on narrow regions of the probability space.
The analysis is based on the means of 5 independently generated sets.

5. Conclusions

By integrating D3PM with engineering-guided strategies, this research enables the synthesis of production layouts that are both structurally valid and operationally optimized.
The core achievement is a generative engine creating functional industrial backbones rather than abstract patterns. Synergy between Logit-Guided Repair and topological projectors ensures node compatibility and flow connectivity without manual post-processing. Through CFG, the model achieved a 156.10% mean throughput increase over baselines while maintaining structural integrity. High-fidelity labeling and validation are further guaranteed by a dedicated DES module.
Despite successes, computational intensity and graph abstraction remain limitations. Future work will focus on Bayesian hyperparameter tuning and scalability. Ultimately, integrating Topological and Geometric Design is essential to co-optimize material flow and spatial placement, evolving this prototype into a comprehensive industrial design tool.

References

[1] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, July 2022.

[2] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models, February 2021.

[3] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete Denoising diffusion for graph generation, May 2023.