Constraint-Guided Discrete Diffusion for Conditional Hierarchical Industrial Graph Generation ⋆

Thomas von Plessing[1,†], Andrea Matta[1,*,†], Mohsen A. Jafari[2,†] and Yulin Li[2,†]

[1]Politecnico di Milano, Via Privata Giuseppe La Masa, 1, 20156 Milano MI, Italy
[2]Rutgers University, 98 Brett Rd, Piscataway, NJ 08854, USA.

Abstract

Industrial production systems must continuously adapt to changes in product mix, machine states, and workforce availability. The ability to swiftly generate new production system layouts and process plans to respond to disruptions is essential. This work introduces a breakthrough two-stage constraint-guided diffusion model that realizes fully automatic, hierarchical industrial layout generation with strict feasibility guarantees. An automatic synthesis is developed by combining a plant-level flow graph with processing stations, buffer stations, assembly and disassembly stations, together with a station-level graph capturing the detailed behavior for each station. The framework trains two discrete diffusion models: one learns the global topology, and the other, conditioned on station type, learns internal Petri net representations for individual stations. A projector is defined to enforce a set of structural and dynamic constraints at every denoising step. The method delivers 100% validity under three progressively constrained inventory scenarios and preserves 99% uniqueness according to the Weisfeiler–Lehman hash. A complete hierarchical layout can be generated in approximately 2 seconds, and simulation-based evaluations confirm the operational competitiveness of the generated designs.

Keywords

graph generation, diffusion models, hierarchical graphs, constraint enforcement, industrial automation

1. Introduction

Smart factories reconfigure material flows and resources on-line during production to adapt to disruptions affecting the availability of resource and external signals. Yet, designing alternative layouts that always meet connectivity, capacity, and control rules is still largely manual and error-prone [1, 2].

This work focuses on the problem of automatically synthesizing alternative plant layouts that respect a set of constraints governing the use of resources and the flow of materials. Classical optimization (mixed integer programming, graph grammars, metaheuristics) scales poorly and transfers little knowledge across projects [3, 4]. Graph neural networks improve component recommendation [5] but cannot *generate* fully valid plants. Discrete diffusion models, originally devised for images and later extended to categorical data [6, 7], have now reached state-of-the-art graph synthesis by denoising noisy samples and *projecting* each intermediate step onto the feasible set [8, 9]. Yet existing work is limited to flat, homogeneous graphs with a few dozen nodes and ignores the hierarchical semantics crucial to manufacturing [10, 11].

We close this gap with a constraint-guided *hierarchical* diffusion framework. An entire plant is abstracted as a directed graph whose nodes are processing machines, buffers, assembly machines, and disassembly machines. Every non-buffer station embeds a Petri net that captures local concurrency, blocking, and resource contention [12, 13]. Diffusion models are developed to generate at plant-level the global topology of the layout, while three type-conditioned chains simultaneously generate at

PMAI25: 4th International Workshop on Process Management in the AI era, October 25, 2025, Bologna, IT.
⋆Extended abstract of a paper submitted to NeurIPS 2025 (under review).
*Corresponding author.
†These authors contributed equally.
thomasedmund.von@mail.polimi.it (T. v. Plessing); andrea.matta@polimi.it (A. Matta); jafari@soe.rutgers.edu (M. A. Jafari); yl959@soe.rutgers.edu (Y. Li)
0009-0000-7644-6859 (T. v. Plessing); 0000-0003-3902-2007 (A. Matta); 0000-0003-4776-0632 (M. A. Jafari); 0009-0001-0135-4332 (Y. Li)

lower level the details of each station using Petri net formalism. Two hard projectors act after every reverse step: the industrial projector enforces machine–buffer pairing and correct in/out degrees for assembly and disassembly machines; the Petri projector guarantees strict place–transition alternation. The sample thus remains inside the feasible set throughout the trajectory.

Experiments on 300 synthetic factories show that the model delivers 100% validity, 86–99% structural uniqueness, and realistic throughput–energy trade-offs. Omitting the projectors reduces validity below 1%, confirming their necessity. To our knowledge, this is the first diffusion approach that unifies multi-level industrial topology synthesis *and* embedded control-logic generation, paving the way for fully automatic, constraint-aware factory design in the AI era.

2. Methodology

Diffusion background. Let $\mathbf{z}_0$ be a categorical encoding of a graph (node labels and adjacency). A discrete forward kernel $q(\mathbf{z}_{t+1} \mid \mathbf{z}_t) = (1-\beta_t)\,\mathbf{z}_t + \beta_t\,\mathbf{u}$, with $\mathbf{u}$ the uniform distribution over the K categories (here K is the number of possible node/edge classes), progressively randomizes the graph for $t = 0, \dots, T-1$. The reverse process learns $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) = \mathrm{Cat}\big(\sigma_\theta(\mathbf{z}_t, t)\big)$, where σ_θ is the neural network that outputs logits for the Categorical distribution, and samples by denoising from $\mathbf{z}_T \sim \mathbf{u}$ back to $\mathbf{z}_0$. In our setting $\mathbf{z} = (X, Y)$ combines the station types X and the edge tensor Y.

Hierarchical plant representation. We formally define our modeling framework as follows. A factory layout is a two–level directed graph $G = (V, E, \gamma, \delta)$: the plant layer contains n stations labeled by $\gamma \in \{\textit{processing machine}, \textit{buffer}, \textit{assembly machine}, \textit{disassembly machine}\}$ and each non-buffer station embeds a Petri net $P_v = (V_v, E_v, \delta)$ with $\delta \in \{\textit{place}, \textit{transition}\}$. Seven hard rules (degree limits, buffer isolation, acyclic flow, place–transition alternation, etc.) define the feasibility set $\mathscr{F}$.

Dual diffusion with hard projection. First, at the plant level a lightweight graph-transformer (two TransformerConv layers, d_h=12) predicts updated logits for station labels and inter-station edges; the industrial projector Π_{pl} then clips or flips entries so that global constraints—one-to-one processing–buffer pairs, correct in/out degrees for assembly and disassembly machines, and the prohibition of buffer–buffer arcs—already hold before the next iteration. At the station level three identical denoisers, each conditioned on its parent's type, generate the internal Petri nets of processing, assembly, and disassembly machines; the Petri projector Π_{pet} removes self-loops and enforces the strict place↔transition alternation that guarantees safe dynamics. A buffer degenerates to a single place and therefore bypasses the station-level chain. Projection after every reverse step keeps the sample inside $\mathscr{F}$, so the final graph is always valid.

With a random timestep $t \in \{0, \dots, T-1\}$ we corrupt the clean graph (X, Y) into $(\hat{X}_t, \hat{Y}_t)$ and minimize

$$\mathscr{L} = \mathrm{CE}_{\mathrm{node}} + \mathrm{CE}_{\mathrm{edge}} + \lambda_{\mathrm{KL}}\,\mathrm{KL}\big(p_\theta \,\|\, p_{\mathrm{data}}\big) + \lambda_c\,\mathscr{L}_{\mathrm{proj}},$$

where $\mathrm{CE}_{\mathrm{node}}$ and $\mathrm{CE}_{\mathrm{edge}}$ are cross-entropy terms that force the network to reproduce the true node labels X and adjacency matrix Y from the noisy inputs; the KL term $\mathrm{KL}(p_\theta \,\|\, p_{\mathrm{data}})$ keeps the one-step reverse marginals aligned with the empirical class frequencies, curbing early-step mode collapse; and $\mathscr{L}_{\mathrm{proj}}$ sums the probabilities assigned to edges that are forbidden by the industrial or Petri-net grammar, thereby penalizing structural violations. The positive scalars λ_{KL} and λ_c control the strength of the KL regularizer and the structural penalty, respectively.

Conditional sampling. Masks let us *(i) free* generate, *(ii) pin* the exact inventory, or *(iii) partially pin* any subset of stations. Pinned labels remain unchanged while the stochastic chain and projectors drive the rest toward a consistent layout. Generating 1 000 hierarchical graphs (15–40 stations each) costs 21 min on commercial GPU.

3. Results

We produced three batches of three hundred layouts each: free generation, full inventory pinning, and partial pinning of the stations. Every layout satisfied the seven structural rules and the Petri-net typing checks, confirming that the projectors keep the diffusion trajectory inside the feasible set. Uniqueness, measured with the Weisfeiler-Lehman hash, remained high: approximately 86% for the free run and 89% when the inventory was fixed, showing that the model still explores a broad solution space even when strong priors are imposed. Maximum-mean discrepancy stayed below 1.3 in all cases, indicating that node-type frequencies and edge density match the training distribution (Table 1).

Table 1
Structural quality of the 300 generated layouts per setting.

Mode	Valid (%)	WL-Uniq (%)	MMD ↓
E1: free	100	86.3	0.95
E2: all-pinned	100	99.3	1.21
E3: partial 30 %	100	93.0	1.33

Operational realism was gauged with a discrete-event simulator that ran each plant for ten thousand time steps with stochastic, type-specific processing times. Throughput and energy use were recorded at every tick. As Figure 1 shows, the generated layouts cover the same throughput-versus-energy frontier as the training set: fully pinned samples cluster on the energy-efficient side, whereas free samples explore a broader range, including high-throughput, high-energy outliers. These results indicate that the model produces layouts that are not only structurally valid but also operationally plausible.

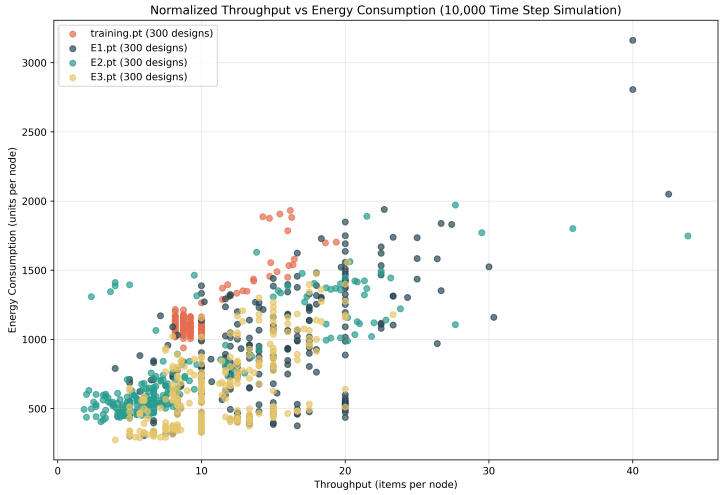

Figure 1: Throughput–energy trade-off for the 300 layouts in each mode compared with the training baseline. The training samples are represented in red color. The figure shows diversity in terms of energy and throughput of the generated layouts compared to the training samples.

4. Conclusion

We presented a hierarchical diffusion framework that combines plant-level and station-level generators with hard projectors, producing industrial layouts that are valid and semantically consistent at every step. Systematic ablations confirmed that the structural projector is essential for handling the combinatorial constraints, while a modest network width of six to twelve hidden units already delivers high structural fidelity and realism. Complementary simulation-based evaluations further show that the generated layouts achieve favorable throughput and energy efficiency, even without explicit performance supervision.

Future research will scale the approach to real factory data and investigate adaptive, learnable constraint sets. In particular, recent large-language-model reasoning could be leveraged to express context-sensitive semantic rules on demand, giving designers even greater flexibility and precision when generating industrial plant layouts.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o in order to: Grammar and spelling check. After using these tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] Y. Lu, K. C. Morris, S. Frechette, Current Standards Landscape for Smart Manufacturing Systems, Technical Report NISTIR 8107, National Institute of Standards and Technology (NIST), 2016. Defines smart manufacturing as fully-integrated, real-time adaptive systems.

[2] S. Antony Jose, A. Tonner, M. Feliciano, T. Roy, A. Shackleford, P. L. Menezes, Smart manufacturing for high-performance materials: Advances, challenges, and future directions, Materials 18 (2025). URL: https://www.mdpi.com/1996-1944/18/10/2255. doi:10.3390/ma18102255.

[3] A. Konak, S. Kulturel-Konak, B. A. Norman, A. E. Smith, A new mixed integer programming formulation for facility layout design using flexible bays, Operations Research Letters 34 (2006) 660–672. doi:10.1016/j.orl.2005.09.009.

[4] R. Brandenburg, A. Borrmann, M. Nagl, Graph transformations in engineering design: A survey of applications, in: Proceedings of the 5th International Workshop on Graph-Based Modeling in Engineering, 2014, pp. 1–8.

[5] Z. Niu, M. Li, Y. Chen, J. Zheng, Graph neural network-based component recommendation for cad assembly design, in: 2022 IEEE International Conference on Industrial Informatics (INDIN), 2022, pp. 357–364. doi:10.1109/INDIN51773.2022.9975880.

[6] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: Proceedings of the 32nd International Conference on Machine Learning (ICML), volume 37 of *JMLR Workshop and Conference Proceedings*, 2015, pp. 2256–2265.

[7] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020, pp. 6840–6851.

[8] E. Hoogeboom, V. G. Satorras, C. Vignac, M. Welling, Equivariant diffusion for molecule generation in 3d, in: Proceedings of the 39th International Conference on Machine Learning (ICML), volume 162 of *Proceedings of Machine Learning Research*, 2022, pp. 8867–8887.

[9] M. Madeira, C. Vignac, D. Thanou, P. Frossard, Generative modelling of structurally constrained graphs, arXiv preprint arXiv:2406.17341 (2023).

[10] M. Karami, HiGen: Hierarchical graph generative networks, arXiv preprint arXiv:2305.19337 (2023).

[11] L. Wang, C. Song, Z. Liu, Y. Rong, Q. Liu, S. Wu, Diffusion models for molecules: A survey of methods and tasks, arXiv preprint arXiv:2502.09511 (2025).

[12] Y. T. Lee, S. Kumaraguru, S. Jain, S. Kumara, A classification scheme for smart manufacturing systems' performance metrics, in: Procedia CIRP, volume 61, 2017, pp. 17–22. doi:10.1016/j.procir.2016.11.251, uses a hierarchy (enterprise → factory → line → cell/machine → process) to model manufacturing systems, aligning with the idea of stations composed of machines and buffers.

[13] J. A. Buzacott, J. G. Shanthikumar, Stochastic Models of Manufacturing Systems, Prentice Hall, 1993. Develops hierarchical queueing network models of production lines and job shops, showing how stations with multiple machines and buffers can be analyzed under stochastic variability.