Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

TEMPORAL DIFFUSION MODELS FROM PARALLEL DEVS MODELS: A GENERATIVE-AI APPROACH FOR SEMICONDUCTOR FABRICATION MANUFACTURING SYSTEMS

Vamsi Krishna Pendyala[1], Hessam S. Sarjoughian[1], Edward J. Yellig[2]

[1]Arizona Center for Integrative Modeling & Simulation, Arizona State University, Tempe, Arizona, USA
[2]Intel Corporation, Chandler, Arizona, USA

ABSTRACT

Generative-AI models offer powerful capabilities for learning complex dynamics and generating high-fidelity synthetic data. In this work, we propose Conditional Temporal Diffusion (CTD) models for generating wafer fabrication time-series trajectories conditioned on static factory configurations. The model is trained using data from a Parallel Discrete Event System Specification (PDEVS)-based MiniFab benchmark model, which simulates different steps of a semiconductor manufacturing process and captures the wafer processing dynamics (e.g., throughput & turnaround time). These simulations incorporate multiscale, realistic behaviors such as preventive maintenance and wafer dispatching under both uniform and sinusoidal generation patterns. CTD models are conditioned on static covariates, including wafer composition, lot sizes, repair type, and wafer generator mode of the factory. Experimental evaluations demonstrate that the synthetic outputs achieve high fidelity with average errors below 15% while significantly reducing data generation time. This highlights CTD's effectiveness as a scalable and efficient surrogate for complex manufacturing simulations.

1 INTRODUCTION

Simulation-based modeling is essential for analyzing complex manufacturing systems. In semiconductor fabrication, high-fidelity discrete-event simulation (DES) models like Intel's MiniFab (Sarjoughian et al. 2023), developed using the Parallel DEVS (PDEVS) formalism (Chow and Zeigler 1994), capture wafer processing dynamics across stages such as Diffusion (not to be confused with *Conditional Temporal Diffusion (CTD) Model*), Implantation, and Lithography. While accurate, these simulations are computationally expensive and scale poorly with diverse factory configurations or long time horizons. To address these limitations, we propose a data-driven alternative based on *Conditional Temporal Diffusion* (CTD) models. In this work, we explore Conditional Temporal Diffusion (CTD) models (Meijer and Chen 2024) as an application of generative AI for semiconductor manufacturing. Rather than proposing a new diffusion algorithm, we adapt the existing conditional diffusion model framework as surrogate models to capture domain-specific temporal dependencies and process variability in MiniFab simulations. Our CTD model is inspired by denoising diffusion approaches (Ho et al. 2020), and adapts the idea of progressive denoising to the time-series domain. It conditions the generation process on static manufacturing covariates such as wafer composition, repair type, and lot size, enabling context-aware trajectory synthesis. Unlike traditional time-series forecasting approaches, where we predict future values using a historical look-back window of the same trajectory (Pendyala et al. 2024), CTD models generate entire trajectories from random noise by learning to reverse a diffusion process. The reverse process is implemented using a TCN-based denoiser that captures long-range dependencies efficiently. Our approach targets two key manufacturing metrics - throughput (TH) and turnaround time (TA). The dataset, derived from PDEVS simulations, consists of time series paired with conditional static covariates. During training, Gaussian noise is added to clean profiles following a predefined schedule, and the model learns to reconstruct the signal from noise, conditioned on the static covariates. Once trained, the CTD model generates realistic TH and TA trajectories for new configurations without re-running the simulation. Importantly, CTD offers constant-time execution across

979-8-3315-8726-0/25/$31.00 ©2025 IEEE

varying lot sizes and configurations, as shown in Figure 4, whereas PDEVS simulation time increases significantly with configuration complexity. This makes CTD highly scalable and computationally efficient for use in real-time or high-throughput scenarios. In extensive evaluations, CTD models achieve average errors below 15% for TH and 10% for TA, while offering a high acceleration of the execution time. We further assess the model's interpretability based on the influence of the input on the generated outputs, generalizability across unseen lot sizes, long-horizon drift behavior, and provide insights into the capabilities and limitations of CTD models for scalable simulation-free data generation in manufacturing systems.

2 RELATED WORKS

In smart connected semiconductor manufacturing, simulation models are important for analyzing key performance indicators such as throughput and turnaround time (Kopp et al. 2020). Throughput and turnaround time trajectories can be viewed as time series, where traditional forecasting methods attempt to predict future values from past trends using a look-back window (Pendyala et al. 2024). However, while forecasting models can predict future points, they cannot generate entirely new trajectories. In contrast, Generative-AI models can synthesize realistic time-series data that closely mirrors training data patterns (Guo and Chen 2024). Generative-AI has shown significant success across domains such as time-series (Rasul et al. 2021; Kollovieh et al. 2023)and natural language processing (Håkansson and Phillips-Wren 2024). Recent techniques like Generative Adversarial Networks (GANs) and Diffusion Models enable realistic simulation data synthesis, reducing reliance on expensive simulations. In the modeling and simulation space, Generative-AI has been used for simulating human interactions (Gao et al. 2024) and preparing for extreme climate events (McCormack and Grierson 2024). Inspired by such work, we explore Generative-AI models to synthesize throughput and turnaround time series for semiconductor manufacturing settings. Our research focuses on developing a Generative-AI model that can generate time series trajectories based on factory-specific static conditions. As per the categorization of Temporal Diffusion models highlighted in (Meijer and Chen 2024), this can be categorized as *conditional generation*. Conditional time-series data generation involves the process of generating time-series profiles based on specific conditions or inputs, like textual prompts or labels, allowing for more controlled and targeted data generation (Zhan et al. 2024). Rather than running numerous simulations, we train our Conditional Temporal Diffusion (CTD) model to generate throughput and turnaround time profiles conditioned on conditional settings including wafer lot configuration, repair type, and uniform/sinusoidal wafer generation patterns.

3 BACKGROUND

Parallel Discrete Event Simulation (PDEVS) models have been developed and used for machine learning (Sarjoughian et al. 2023; Pendyala et al. 2024), where machine learning models utilize simulation-derived data. The efficiency of the manufacturing process (e.g., factory throughput) depends on the specification of each model, such as repair mode and parameterization (e.g., transportation times). It also depends on the wafer lots that have different configurations, frequencies, and patterns (e.g., uniform), entering the factory.

3.1 PDEVS Semiconductor Fabrication Model

Single-stage and multi-stage semiconductor fabrication factory models are constructed using the Parallel Discrete Event System Specification (PDEVS) formalism (Chow and Zeigler 1994) and executed through the DEVS-Suite simulator (ACIMS 2023), following the MiniFab benchmark model (Spier and Kempf 1995). The factory is represented as a coupled system comprising Diffusion (not to be confused with *Conditional Temporal Diffusion (CTD) Model*), Implantation, and Lithography machines. Each machine processes wafer lots through a series of fixed, non-preemptive phases—loading, processing, unloading, and transportation—with configurable stochastic durations. Machines may enter a repair mode after processing a predefined number of lots or based on the mean time between failures. Coordinators manage the formation and dispatching of wafer lots/batches to the appropriate machines. The diffusion (not to be confused with

Conditional Temporal Diffusion (CTD) Model) stage features a dispatcher with a queue feeding two identical processing machines. Similarly, the implantation and lithography stages consist of machines that process wafer lots and pass them downstream, adhering to specified timing constraints. These stages are linked in a sequential cascade factory setting to model multi-stage fabrication systems. Wafer lots undergo six distinct processing steps—two each in Diffusion, Implantation, and Lithography—within a feedforward-feedback loop. Transducers collect data from the machines and coordinators without affecting system behavior. The simulation operates with a time resolution in minutes and includes 10% stochastic variability in wafer processing times. An 8-stage fabrication system is created by cascading eight single-stage factories, where each stage's output becomes the input to the next. The factory handles three wafer types—Product a (Pa), Product b (Pb), and Test wafer (Tw)—which are grouped into batches/lots of three (wafer batch/lot formation rules are applied at every stage. E.g., *at most one* Tw wafer in a batch/lot of size three) at each stage and process them chronologically in a 6-step process. For each stage, we have the diffusion (not to be confused with *Conditional Temporal Diffusion (CTD) Model*) machines A and B perform steps 1 and 5, implantation machines C and D perform steps 2 and 4, and lithography machine E handles steps 3 and 6 as shown in the Figure 1. Wafer lots are generated by atomic models at fixed intervals: Pa every 8 hours, Pb every 16 hours, and Tw every 24 hours. Each of the sinusoidal wafer generators has sequenced amplitudes 1, 2, 3, 2, and 1.

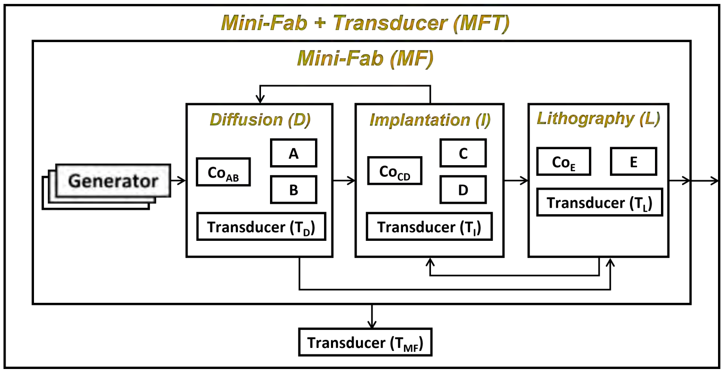


Figure 1: Component-based PDEVS model (Sarjoughian et al. 2023).

3.2 Conditional Temporal Diffusion Models

Conditional Temporal Diffusion (CTD) models work by establishing a Markov process (Kemeny and Snell 1960) that progressively introduces noise into a clean data sample in D number of forward diffusion steps, where entire trajectories are corrupted to capture both temporal dependencies and the underlying data distribution (Lin et al. 2024). A noise variance schedule, $\{\beta_d\}_{d=1}^{D}$, where $\beta_d \in [0,1]$ determines the noise addition for D steps of the forward diffusion process. The cumulative noise factor is computed as $\bar{\alpha}_d = \prod_{s=1}^{d}(1-\beta_s)$, representing the fraction of the original signal preserved up to step d. For a clean time series y, the noisy data at step d is generated by $y^d = \sqrt{\bar{\alpha}_d}\, y + \sqrt{1-\bar{\alpha}_d}\, \varepsilon$, where ε is sampled from a noise distribution $p(\varepsilon)$. During training, a neural network $\mathscr{G}_\theta$ (parameterized by θ) is trained to predict the denoised output via $\hat{y} = \mathscr{G}_\theta(y^d, c)$, where c represents conditional static covariate, by minimizing the loss - $\mathscr{L}(y,\hat{y})$. For N number of time-series trajectories $\{(y_i,c_i)\}_{i=1}^{N}$, where y_i corresponds to a time-series trajectory and c_i corresponds to the respective conditional static variables, the training procedure is as mentioned in Algorithm 1. This allows the model to denoise trajectories across varying noise levels.

Algorithm 1 Conditional Temporal Diffusion (CTD) model training procedure.

```
Input: Time-series profiles - $\{(y_i, c_i)\}_{i=1}^{N}$ where $y_i$ represents a time-series profile and $c_i$ represent its corresponding Conditional Static Covariates, Number of Diffusion Steps $D$
for each epoch do
    for each time-series profile $(y_i, c_i)$ do
        Sample a random diffusion step $d \in \{1, \ldots, D\}$
        Sample noise $\varepsilon \sim p(\varepsilon)$
        Compute $\bar{\alpha}_d = \prod_{s=1}^{d}(1 - \beta_s)$
        Generate noisy input: $y_i^d = \sqrt{\bar{\alpha}_d}\, y_i + \sqrt{1 - \bar{\alpha}_d}\, \varepsilon$
        Predict denoised output: $\hat{y}_i = \mathscr{G}_\theta(y_i^d, c_i)$
        Compute loss: $\mathscr{L}(y_i, \hat{y}_i)$
        Update model parameters $\theta$ to minimize loss
    end for
end for
```

In the generation phase, starting from a noisy signal $y^D \sim p(y^D)$, the trained model $\mathscr{G}_\theta$ model iteratively refines the signal using the reverse update. This iterative reverse process gradually removes the noise, ultimately reconstructing a clean time-series y^0, and the model applies the learned reverse diffusion process to generate new time-series trajectories as described in Algorithm 2. This general framework for training and testing temporal diffusion models provides a flexible and scalable approach for generating high-fidelity synthetic time series conditioned on static variables. This can be extended to generate time-series profiles, which can be computationally intensive to generate using simulation models.

Algorithm 2 Conditional Temporal Diffusion (CTD) model time-series profile generation procedure.

```
Input: Trained Model $\mathscr{G}_\theta$, Conditional Static Covariates $c$, Number of Diffusion Steps $D$, Noise Schedule $\{\beta_d\}_{d=1}^{D}$
Initialize $y^D \sim p(y^D)$
for $d = D \ldots, 1$ do
    Predict denoised signal: $\hat{y}^{d-1} = \mathscr{G}_\theta(y^d, c)$ Set value:
    $y^{d-1} \leftarrow \hat{y}^{d-1}$
end for
Apply smoothing: $y^0 = \texttt{GaussianFilter}(y^0, \sigma = 1)$
return $y^0$
```

4 METHODOLOGY

The process of generating throughput and turnaround time profiles of a semiconductor manufacturing factory involves the transformation of noise signals into valid time-series data for faster computation than the simulation models. Hence, we define our problem statement as -

Problem Statement: Let dataset $\mathscr{D} = \{(y_i, c_i)\}_{i=1}^{N}$, where each $y_i \in \mathbb{R}^T$ is a time-series trajectory for T time steps and $c_i \in \mathbb{R}^k$ is a vector of conditional static covariates of size k that define y_i. Define a generative model $\mathscr{G}_\theta : \mathbb{R}^T \times \mathbb{R}^k \to \mathbb{R}^T$, such that, for a noise vector $n \sim p(n)$ with $p(n)$ being a suitable prior distribution on $\mathbb{R}^T$, the synthetic time series is given by $\hat{y}_i = \mathscr{G}_\theta(n, c)$. We determine the optimal parameters θ^* for $\mathscr{G}_\theta$ by solving $\theta^* = \arg\min_\theta \mathbb{E}_{(y_i, c_i) \sim \mathscr{D}, n \sim p(n)} \left[\mathscr{L}\big(y_i, \mathscr{G}_\theta(n, c)\big)\right]$, where $\mathscr{L}$ is a loss function.

The model $\mathscr{G}_\theta$ employs a two-phase procedure: a training phase that learns to reverse a forward diffusion process and a testing phase that generates new trajectories from random noise as described in Section 3.2.

4.1 PDEVS Simulation Dataset

The PDEVS-based semiconductor fabrication model described in Section 3.1 is used to generate throughput (TH) and turnaround time (TA) trajectories. Since these are discrete-event outputs with a continuous-time base and sparse value changes, the trajectories' time bases must be pre-processed into discretized time bases for ML-based analysis. We convert each discrete-event trajectory into a discrete time series by keeping the time interval fixed to one minute. Forward-filling is applied to fill missing values, reflecting the piecewise-constant nature of DEVS atomic model outputs for throughput and turnaround time. Figure 3 illustrates the TH and TA profiles after this transformation for an 8-stage cascade MiniFab factory. Each time-series trajectory is defined by its conditional static covariates: wafer types (Pa, Pb, Tw), lot size, repair type- Mean Time Between Failure (MTBF) or Processing-steps based, and wafer generation pattern- Uniform or Sinusoidal (Pendyala et al. 2024). The simulation experiments span over six factory scenarios as summarized in Table 2, each combining a repair strategy with a wafer generator type. Wafer lots are grouped into small (60–108 wafers), medium (120–156), and large (168–204). For training, 9 lot configurations (3 from each lot size group) were used per scenario, resulting in a total of 54 TH and TA profiles. Testing was performed using 3 new configurations (1 per lot size group) for each scenario, totaling 18 test profiles. All configurations are detailed in Table 1. Each trajectory spans 30,000 time steps (minutes), corresponding to the upper bound on simulation duration across all configurations. Additionally, to assess scalability and computational efficiency (discussed in Section 5.2), we generated TH and TA profiles for 93 distinct wafer configurations, varying Pa, Pb, Tw, and lot size.

Table 1: Simulation Wafer Configurations.

	Train Configurations				**Test Configurations**			
Lot Size Category	*Pa*	*Pb*	*Tw*	*Lot Size*	*Pa*	*Pb*	*Tw*	*Lot Size*
Small	0	54	18	72	3	42	15	60
	0	72	0	72				
	6	51	3	60				
Medium	78	30	12	120	72	45	27	144
	90	18	36	144				
	93	24	39	156				
Large	120	30	30	180	150	24	18	192
	147	0	33	180				
	168	9	15	192				

Table 2: Simulation Scenarios.

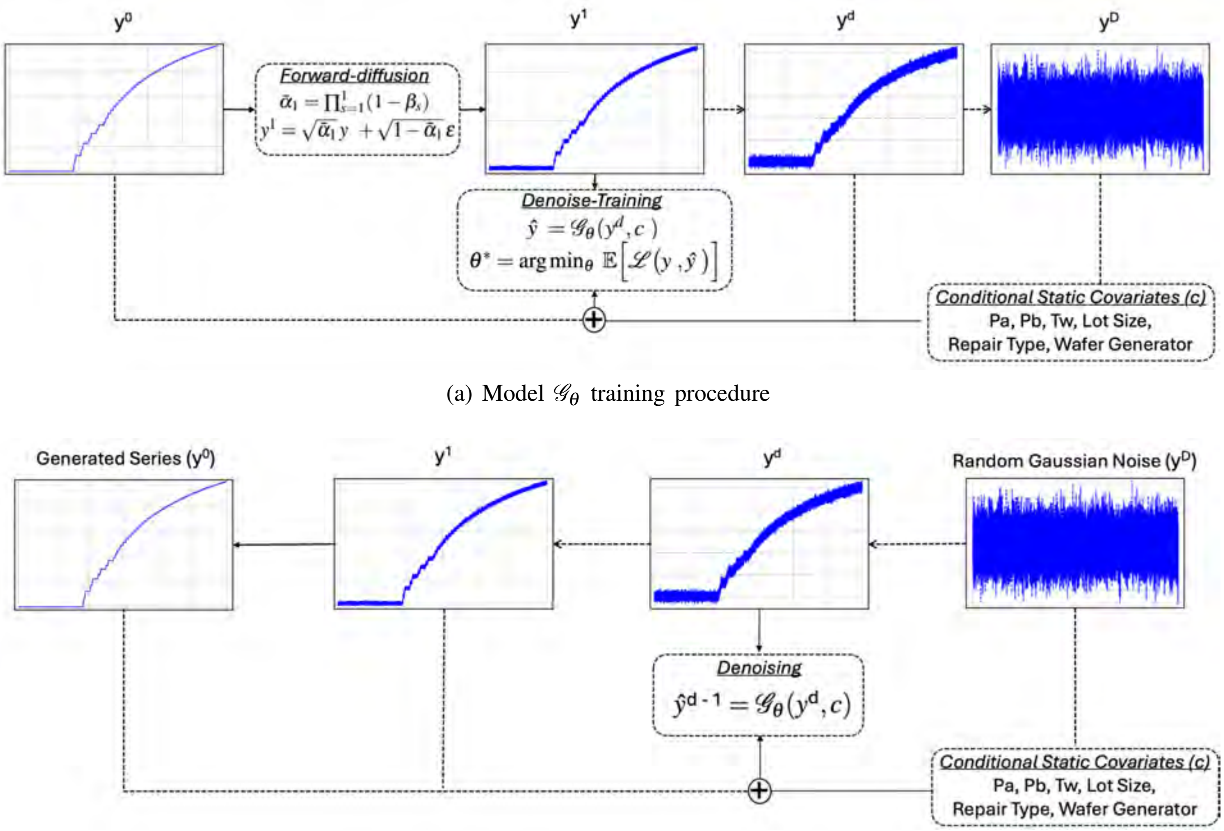
Scenario	**Repair Type**	**Wafer Generation**
A	No Repair	Sinusoidal
B	No Repair	Uniform
C	MTBF	Uniform
D	Processing Steps	Uniform
E	MTBF	Sinusoidal
F	Processing Steps	Sinusoidal

4.2 Model Architecture

The dataset described in Section 4.1 consists of time-series profile $y \in \mathbb{R}^T$ paired with static covariates $c \in \mathbb{R}^k$, which is further normalized using MinMax scaling for training stability (Pedregosa et al. 2011). As discussed in Section 3.2, the $\mathscr{G}_\theta$ predicts denoised $\hat{y}$ trajectory from a noisy trajectory conditioned on static covariates to reconstruct TH & TA profiles. The model employs a forward diffusion process to corrupt inputs and a reverse process to generate clean sequences.

4.2.1 Training Procedure

Training involves teaching the model to reverse a forward diffusion process as shown in Figure 2(a) and discussed in Algorithm 1. Given a pre-processed simulation time-series $y \in \mathbb{R}^T$ (e.g., TH or TA), noise is added based on a predefined schedule $\{\beta_d\}_{d=1}^{D}$, where we trained our model with $T = 30{,}000$, $D = 2{,}000$ and β_d linearly ranging from 0.01 to 0.1. The cumulative noise factor is computed as $\bar{\alpha}_d = \prod_{s=1}^{d}(1-\beta_s)$, representing the preserved proportion of the original signal up to diffusion step d. The noisy signal at step d is $y^d = \sqrt{\bar{\alpha}_d}y + \sqrt{1-\bar{\alpha}_d}\varepsilon$, where $\varepsilon \sim \mathcal{N}(0,1)$ is a Guassian Noise. The use of zero-mean, unit-variance Gaussian noise ensures stochasticity while preserving differentiability, hence suitable for training the reverse denoising model via gradient-based optimization. We have used a Temporal Convolutional Network (TCN)


y⁰
y¹
yᵈ
yᴰ
Forward-diffusion
$\bar{\alpha}_1 = \prod_{s=1}^{1}(1-\beta_s)$
$y^1 = \sqrt{\bar{\alpha}_1}\, y + \sqrt{1-\bar{\alpha}_1}\, \varepsilon$
Denoise-Training
$\hat{y} = \mathcal{G}_\theta(y^d, c)$
$\theta^* = \arg\min_\theta \mathbb{E}\left[\mathcal{L}(y, \hat{y})\right]$
Conditional Static Covariates (c)
Pa, Pb, Tw, Lot Size,
Repair Type, Wafer Generator
Generated Series (y⁰)
y¹
yᵈ
Random Gaussian Noise (yᴰ)
Denoising
$\hat{y}^{d-1} = \mathcal{G}_\theta(y^d, c)$
Conditional Static Covariates (c)
Pa, Pb, Tw, Lot Size,
Repair Type, Wafer Generator

(a) Model $\mathcal{G}_\theta$ training procedure

(b) Model $\mathcal{G}_\theta$ profile generation

Figure 2: Model $\mathcal{G}_\theta$ architecture for generating time-series profiles (TH/TA vs time).

(Bai et al. 2018) model as $\mathcal{G}_\theta$ since it is designed to capture long-range dependencies. The model takes the noisy time-series y^d, and static covariates c as inputs and outputs the denoised profile $\hat{y}^{d-1} = \mathcal{G}_\theta(y^d,c)$. The final output layer of the model employs a Softplus Activation to ensure non-negative values. The training objective is to minimize the composite loss function ($\mathcal{L}$), for which we have used the Mean Squared Error (MSE) and Mean Absolute Error (MAE), to reduce overall error by balancing the smoothness and sharpness of the generated trajectory. The model parameters θ are updated using the Adam Optimizer (Kingma and Ba 2014), and we trained our model for 50 epochs.

4.2.2 Time-series Profile Generation

The generation process, or reverse diffusion as shown in Figure 2(b), begins with a random noisy signal $y^D \sim p(y^D)$ (typically sampled from a standard Gaussian distribution). The model then progressively removes noise using the learned reverse update. For each denoising step d (iterating backward from D to 1), the model predicts the denoised trajectory $\hat{y}^{d-1}$ from the current noisy input y^d (conditioned on c and d). To further smooth the generated trajectory and remove any residual high-frequency artifacts, a Gaussian filter (Gonzalez and Woods 2002) is applied along the time axis to smooth high-frequency noise. The final output y^0 is the generated, denoised time-series profile.

5 EXPERIMENTAL RESULTS

Experiments were conducted for the CTD models based on the scenarios described in Table 2 for wafer configurations mentioned in Table 1. Every TH and TA profile for our experiments is defined by their conditional static variables: Pa, Pb, Tw, Lot Size, Repair Type, and Wafer Generator. The CTD models are evaluated on their capability to generate 18 different TH & TA time-series profiles as mentioned in Section 4.1, compared to the actual PDEVS simulation time-series profiles. To quantitatively assess performance, we employ standard regression evaluation metrics—Mean Squared Error (MSE), Coefficient of Determination (R^2), and Mean Absolute Percentage Error (MAPE)—which provide insight into the accuracy, fit quality, and relative deviation of the generated profiles from the ground truth (Kim et al. 2025).

5.1 Throughput & Turnaround Time Profile Generation

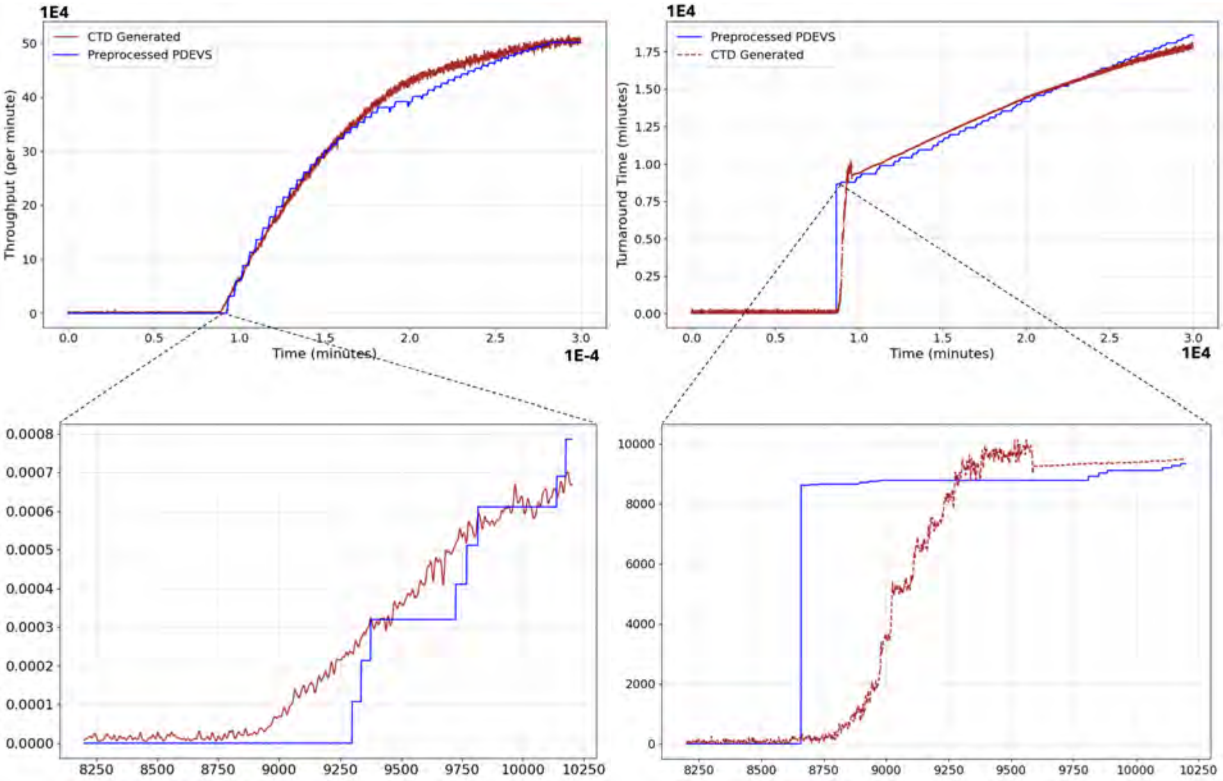
After training the Conditional Temporal Diffusion (CTD) model, we evaluated it on 18 test configurations across all simulation scenarios defined in Table 1 and Table 2. Figure 3 compares the generated throughput (TH) and turnaround time (TA) profiles against the ground truth from preprocessed PDEVS simulations. The CTD model accurately reconstructs full-length TH and TA trajectories with minimal deviations. Unlike traditional forecasting approaches (Pendyala et al. 2024), which predict one step at a time using historical data, CTD generates the entire sequence from noise, conditioned only on static covariates. Zoomed-in plots highlight the model's ability to reproduce local fluctuations based solely on static input, as shown in Figure 3 and discussed in Section 3.2. Quantitative results in Table 3 show that CTD achieves an average MAPE below 15% for TH and below 10% for TA. The MSE values for TH & TA are in the range of 10^{-8} & 10^{8} because of TH and TA values being in the range of 10^{-4} & 10^{4} respectively. Turnaround time is easier to predict due to its stable nature, while throughput, being sensitive to factory dynamics, shows more variation. Scenario B (no repair, uniform generator) yields the best TH accuracy due to its simplicity, while Scenarios E and F exhibit higher errors (14.29% for TH and 6.10% for TA) due to preventive maintenance and sinusoidal input patterns. Despite these challenges, CTD consistently captures temporal trends, offering a reliable and efficient surrogate for complex manufacturing simulations.

Table 3: Average MSE, $MAPE$, and R^2 Scores on Test Configurations for CTD Model.

	Throughput			**Turnaround Time**		
Scenario	*MSE*	R^2*Score*	*MAPE*	*MSE*	R^2*Score*	*MAPE*
A	8.07E-08	0.970	10.69%	1.25E+06	0.970	3.38%
B	3.66E-08	0.970	7.17%	1.27E+06	0.967	5.50%
C	1.26E-07	0.958	14.20%	1.40E+06	0.963	6.21%
D	9.58E-08	0.968	12.88%	1.07E+06	0.971	4.96%
E	1.43E-07	0.951	14.29%	1.20E+06	0.968	5.74%
F	1.03E-07	0.965	13.20%	1.27E+06	0.966	6.10%

5.2 Computational Efficiency

An important advantage of Conditional Temporal Diffusion (CTD) models over traditional PDEVS-based simulation is their superior computational efficiency. To empirically evaluate this, we compare execution times for both approaches under identical conditions. The experiment involves 93 unique wafer configurations (Pa, Pb, Tw, and Lot Size), using a uniform wafer generator with no repair (Scenario B), and measures the time to generate complete throughput (TH) and turnaround time (TA) profiles. As shown in Figure 4, the execution time for the simulation model (Figure 4a) scales non-linearly with lot size, increasing from approximately 100 seconds at a lot size of 60 to over 1000 seconds at a lot size of 204. This increase stems from the discrete event simulation's intrinsic step-wise execution, where complexity grows with the number of wafers, machines, repair cycles, and stochastic delays, resulting in an approximate time complexity of $\mathcal{O}(n \cdot m \cdot d)$—where n is the number of wafer events, m is the number of machines, and d is the number of delays per wafer. In contrast, CTD model generation time remains nearly constant across configurations (Figure 4b), averaging around 0.05 seconds per configuration. Since the model generates an entire time



(a) TH profile (Scenario B)

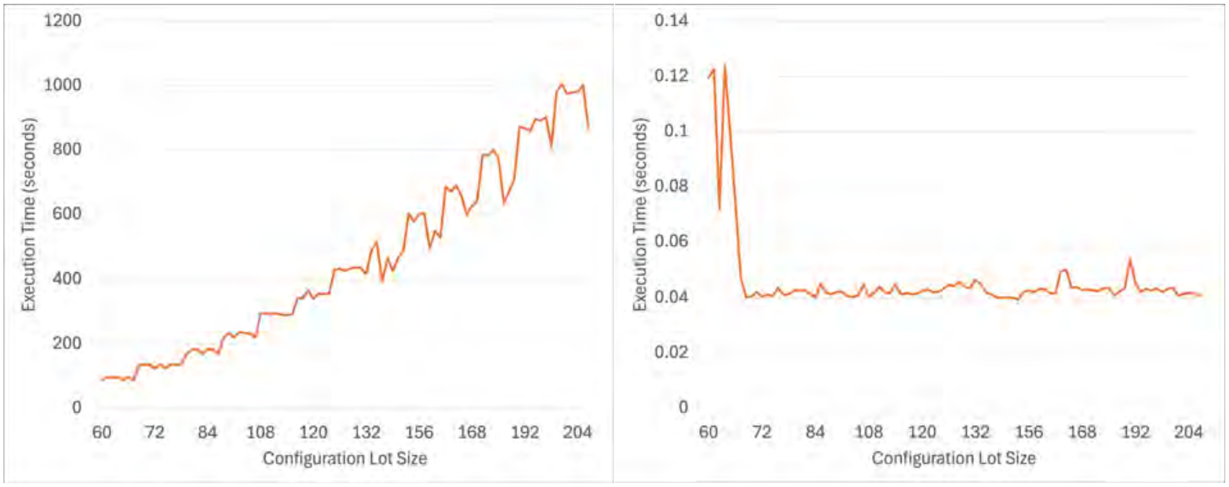
(b) TA profile (Scenario C)

Figure 3: TH and TA profile generation for 8 stage MiniFab cascade factory (Pa=72, Pb=45, Tw=27).

series in a single forward pass through a neural network, its time complexity is $\mathscr{O}(T)$, where T is the fixed length of the output sequence (e.g., 30,000 time steps). This complexity is independent of the internal logic of wafer movement or lot configuration, yielding scalable and efficient inference irrespective of simulation conditions.

5.3 Impact of Conditional Static Covariates

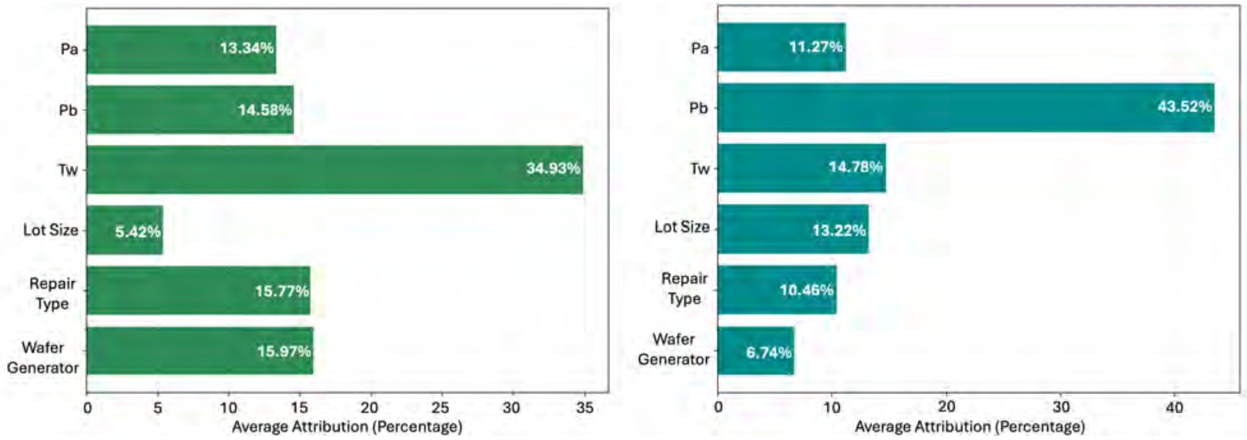
Static covariates play a crucial role in shaping throughput and turnaround time profiles and generating these profiles using CTDs. In this experiment, the CTD model was trained on various configurations involving wafer composition (Pa, Pb, Tw), Lot Size, Repair Type, and Wafer Generator, and we checked the impact of each of these static covariates on the CTD model. As shown in Figure 5, feature importance was assessed using Integrated Gradients (IG) (Sundararajan, Taly, and Yan 2017), which computes attributions by integrating output gradients with respect to input features along a baseline path. For throughput (TH), the Tw (Test wafer) variable had the largest influence, contributing 34.93%, highlighting its role in creating valid wafer lots and directly impacting throughput as mentioned in Section 3. Lot Size had smaller contributions of 5.42% as it is just the sum of total number of Pa, Pb & Tw wafers. For turnaround time (TA), Pb had the highest impact with 43.52%, followed by Tw (14.78%) and Repair Type (10.46%). These results further demonstrate the significant role of wafer types in both TH and TA profile generation, crucial for valid wafer lot formation (Pendyala et al. 2024). Static covariates were normalized using MinMax scaling to ensure stable and equal contributions during training. The importance of each feature was calculated over 50 samples, and average attribution percentages were plotted.



(a) Simulation Model

(b) CTD Model

Figure 4: Computational Efficiency.



(a) TH Profile Attribution

(b) TA Profile Attribution

Figure 5: Conditional Static Covariate Attribution.

5.4 Cross-Process Transferability

To evaluate the generalizability of the Conditional Temporal Diffusion (CTD) model, we conduct a cross-process transferability test by training separate models on small (M_{small}), medium (M_{medium}), and large (M_{large}) lot sizes, as defined in Section 4.1. These are compared to the original model trained on the full dataset (M_{original}). As shown in Figure 6(a), MAPE for TH exceeds 60% for M_{small}, indicating poor generalization. M_{medium} and M_{large} perform better, with MAPE in the range of 35–40% and 25–30%, respectively. In contrast, M_{original} achieves MAPE below 15% across all scenarios. For TA, which is less sensitive to process variations, all models generalize well, with MAPE under 6%. Since TA reflects average processing time per wafer or lot, its temporal profile is more stable. However, the overall accuracy reduces when model is trained with less data even for TA profiles. These results highlight the CTD model's sensitivity to training diversity, particularly for volatile metrics like throughput.

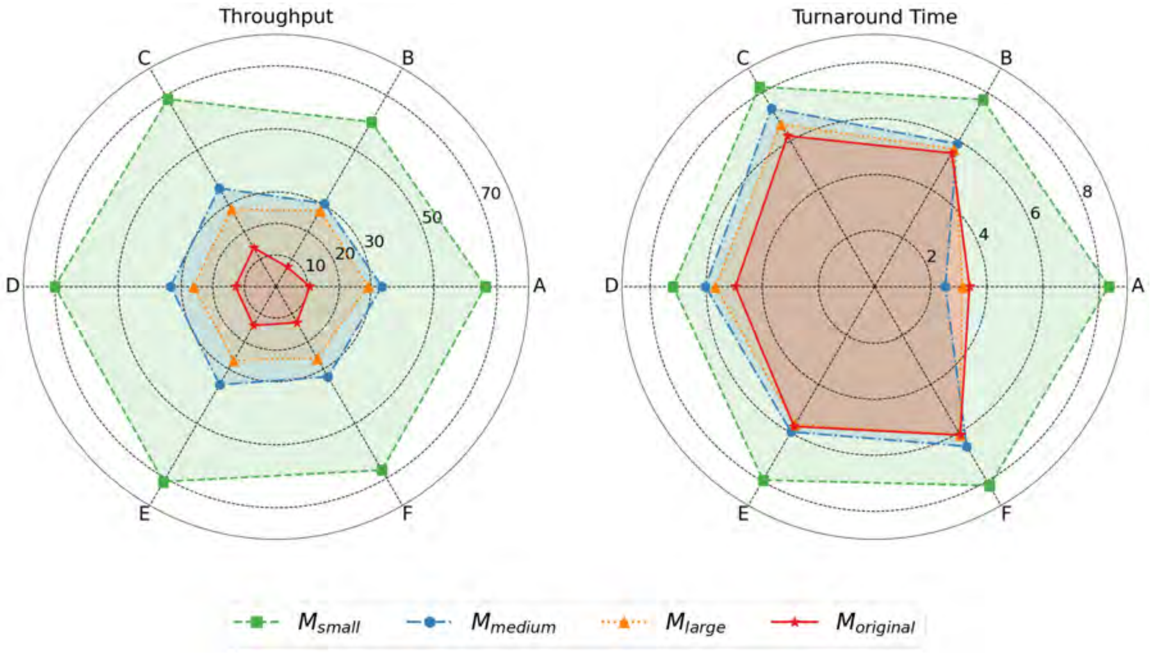
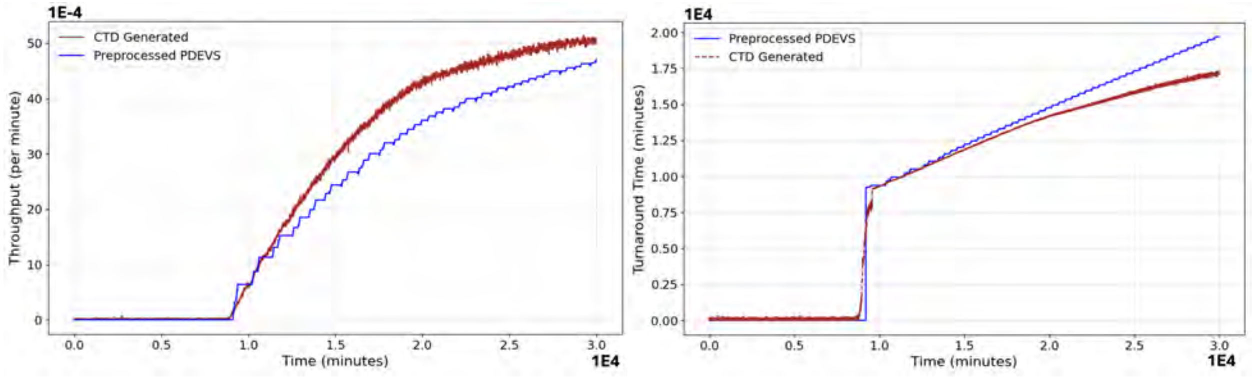


Figure 6: Average $MAPE(\%)$ of profile generation for test configurations.

5.5 Long-Term Stability & Drift Analysis

To assess long-horizon stability, we conduct a drift analysis comparing CTD-generated trajectories against PDEVS ground truth. As shown in Figure 7, CTD accurately follows initial TH and TA patterns but gradually diverges over time. This drift stems from its non-autoregressive formulation: the entire sequence is generated from a single noisy input y_D, conditioned on static covariates c, producing output $y^0 = \mathscr{G}_\theta(y^D, c)$. Without temporal correction, small deviations accumulate, especially over long durations. While trends are well captured, smoothing via Gaussian filters can suppress key transitions, especially in throughput—highlighting the need for autoregressive or corrective extensions for improved long-term fidelity. Additionally, the diffusion model is configured to generate TH and TA profiles up to a fixed horizon of 30,000 minutes. However, the actual simulation end-time varies with factory configurations, and without this contextual information, the model may overshoot or undershoot the actual length, and incorporating dynamic end-time awareness could prevent this.



(a) TH profile (Scenario A)

(b) TA profile (Scenario B)

Figure 7: TH and TA profile generation for 8 stage MiniFab cascade factory (Pa=150, Pb=24, Tw=18).

6 CONCLUSION & FUTURE SCOPE

We proposed Conditional Temporal Diffusion models as fast, data-driven surrogates to simulate semiconductor fabrication manufacturing systems. Trained on 54 configurations and evaluated on 18 test cases, CTD achieved mean absolute percentage errors of less than 15% for throughput (TH) and 10% for turnaround time (TA), while reducing execution time by a factor of 10^3 compared to the DEVS-Suite simulator. Unlike step-ahead time-series forecasting, CTD models generate the entire time-series trajectories from noise, conditioned only on finite prior simulation data sets and static covariates. While CTD produces high-fidelity trajectories, it is sensitive to training diversity — cross-process experiments. For example, it shows accuracy drops when trained on limited lot-size configurations, revealing its dependence on representative data. Nevertheless, CTD models are promising as surrogate simulation. This study provides a demonstration of the role of conditional covariates like wafer type, wafer lot size/configuration, wafer generation pattern, and repair type. Future work will explore the robustness of the models as the patterns and durations of the simulated MiniFab data trajectories are varied. Additionally, we will focus on addressing long-horizon drift and improving model robustness. This involves expanding the space of conditional variables (e.g., changing routing patterns) and data trajectories (e.g., durations vary from a few weeks to months) to better capture real-world complexity that governs manufacturing operational scenarios.

ACKNOWLEDGMENTS

This research is funded by Intel Corporation, Chandler, Arizona, USA.

REFERENCES

ACIMS 2023. “DEVS-Suite Simulator, version 7.0”. https://acims.asu.edu/devs-suite/, accessed 15th March 2024.

Bai, S., J. Z. Kolter, and V. Koltun. 2018. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*. July 10th-15th, Stockholmsmässan, Sweden, 732-740.

Chow, A. C. H., and B. P. Zeigler. 1994. “Parallel DEVS: A Parallel, Hierarchical, Modular Modeling Formalism”. In *1994 Winter Simulation Conference (WSC)*, 716–722 https://doi.org/10.1109/WSC.1994.717419.

Gao, C., X. Lan, N. Li, Y. Yuan, J. Ding, Z. Zhou, *et al*. 2024. “Large Language Models Empowered Agent-Based Modeling and Simulation: A Survey and Perspectives”. *Humanities and Social Sciences Communications* 11(1):1–24.

Gonzalez, R. C., and R. E. Woods. 2002. *Digital Image Processing*. New Jersey, USA: Prentice-Hall, Inc.

Guo, X., and Y. Chen. 2024. “Generative AI for Synthetic Data Generation: Methods, Challenges and The Future”. *arXiv preprint arXiv:2403.04190*.

Håkansson, A., and G. Phillips-Wren. 2024. “Generative AI and Large Language Models - Benefits, Drawbacks, Future and Recommendations”. *Procedia Computer Science* 246:5458–5468.

Ho, J., A. Jain, and P. Abbeel. 2020. “Denoising Diffusion Probabilistic Models”. *Advances in Neural Information Processing Systems* 33:6840–6851.

Kemeny, J. G., and J. L. Snell. 1960. *Finite Markov Chains*. New Jersey, USA: D. Van Nostrand Company.

Kim, J., H. Kim, H. Kim, D. Lee, and S. Yoon. 2025. “A Comprehensive Survey of Time Series Forecasting: Architectural Diversity and Open Challenges”. *arXiv preprint arXiv:2411.05793*.

Kingma, D. P., and J. Ba. 2014. “Adam: A Method for Stochastic Optimization”. *arXiv preprint arXiv:1412.6980*.

Kollovieh, M., A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. Wang. 2023. “Predict, Refine, Synthesize: Self-Guiding Diffusion Models for Probabilistic Time Series Forecasting”. *Advances in Neural Information Processing Systems* 36:28341–28364.

Kopp, D., M. Hassoun, A. Kalir, and L. Mönch. 2020. “SMT2020—A Semiconductor Manufacturing Testbed”. *IEEE Transactions on Semiconductor Manufacturing* 33(4):522–531.

Lin, L., Z. Li, R. Li, X. Li, and J. Gao. 2024. “Diffusion Models For Time-series Applications: A Survey”. *Frontiers of Information Technology & Electronic Engineering* 25(1):19–41.

McCormack, J., and M. Grierson. 2024. *Building Simulations with Generative Artificial Intelligence*, 137–150. Switzerland: Springer Nature.

Meijer, C., and L. Y. Chen. 2024. “The Rise of Diffusion Models in Time-Series Forecasting”. *arXiv preprint arXiv:2401.03006*.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al*. 2011. “Scikit-learn: Machine Learning in Python”. *The Journal of Machine Learning Research* 12:2825–2830.

Pendyala, V. K., H. Sarjoughian, B. S. Potineni, and E. Yellig. 2024. “A Benchmark Time Series Dataset for Semiconductor Fabrication Manufacturing Constructed using Component-based Discrete-Event Simulation Models”. *arXiv preprint arXiv:2408.09307*.

Pendyala, V. K., H. S. Sarjoughian, and E. J. Yellig. 2024. “Generating TCN Models from Parallel DEVS Models: Semiconductor Manufacturing Systems”. In *2024 Winter Simulation Conference (WSC)*, 2265–2276 https://doi.org/10.1109/WSC63780.2024.10838759.

Rasul, K., C. Seward, I. Schuster, and R. Vollgraf. 2021. “Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting”. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 8857–8868. PMLR.

Sarjoughian, H. S., F. Fallah, S. Saeidi, and E. J. Yellig. 2023. “Transforming Discrete Event Models To Machine Learning Models”. In *2023 Winter Simulation Conference (WSC)*, 2662–2673 https://doi.org/10.1109/WSC60868.2023.10407348.

Spier, J., and K. Kempf. 1995. “Simulation of Emergent Behavior in Manufacturing Systems”. In *Proceedings of SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 90–94. November 13th-15th, Cambridge, MA, USA, 90-94.

Sundararajan, M., A. Taly, and Q. Yan. 2017. “Axiomatic Attribution for Deep Networks”. In *International Conference on Machine Learning (ICML)*, 3319–3328. August 6th-11th, Sydney, Australia, 3319-3328.

Zhan, Z., D. Chen, J.-P. Mei, Z. Zhao, J. Chen, C. Chen, *et al*. 2024. “Conditional Image Synthesis with Diffusion Models: A Survey”. *arXiv preprint arXiv:2409.19365*.

AUTHOR BIOGRAPHIES

VAMSI KRISHNA PENDYALA is a Ph.D. student in the Computer Science program in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University (ASU), Tempe, AZ, USA. He can be reached at vpendya2@asu.edu

HESSAM S. SARJOUGHIAN is an Associate Professor of Computer Science and Computer Engineering in the School of Computing and Augmented Intelligence (SCAI) at Arizona State University (ASU), Tempe, Arizona. His research interests include model theory, poly-formalism modeling, machine learning, collaborative modeling, simulation for complexity science, and M&S frameworks/tools. He is the co-director of the Arizona Center for Integrative Modeling and Simulation https://acims.asu.edu. He can be contacted at hessam.sarjoughian@asu.edu.

EDWARD J. YELLIG is the director of Operational Decisions Support Technology at Intel Corporation. He has been with Intel for 26 years and has a Ph.D. in Operations Research with an emphasis on discrete event modeling of large-scale systems. His focus has been on developing fab models for determining capital requirements and is also responsible for the real-time digital twin tactical models. He can be contacted at edward.j.yellig@intel.com.