4. Results:

4.1 Morphology Generation with Conditional GAN

To evaluate the generative capability of the conditional GAN (cGAN) architecture, we selected a representative subset of 16 process parameter combinations covering a wide spectrum of line morphologies observed in the training dataset. This subset was chosen to test the model's ability to produce diverse output patterns when conditioned on known input parameters. Figure S1(A) presents a side-by-side visual comparison between experimentally acquired binary morphology images (left) and the corresponding cGAN-generated predictions (right). The generated morphologies qualitatively resemble the structural features of the experimental prints, including variations in line width, overspray behavior, and trace continuity.

To further assess performance, Figures S1(B–D) show kernel density estimations (KDEs) comparing the distribution of three morphological metrics—line width (Lw), overspray extent (Φ), and edge roughness (R_q)—between the experimental and generated image sets. Across all three metrics, the predicted distributions closely track the empirical distributions, capturing both the central tendency and spread of the data.

These results lead to several key observations:

(i) The cGAN demonstrates the ability to synthesize a broad range of morphology types, without collapsing to a narrow output space or favoring specific features;

(ii) The model shows sensitivity to distinct process parameter combinations, suggesting successful learning of the underlying parameter-to-structure mapping; and

(iii) The strong agreement in statistical distributions supports that the generator not only produces realistic images but also preserves the variability and structural statistics of the training dataset.

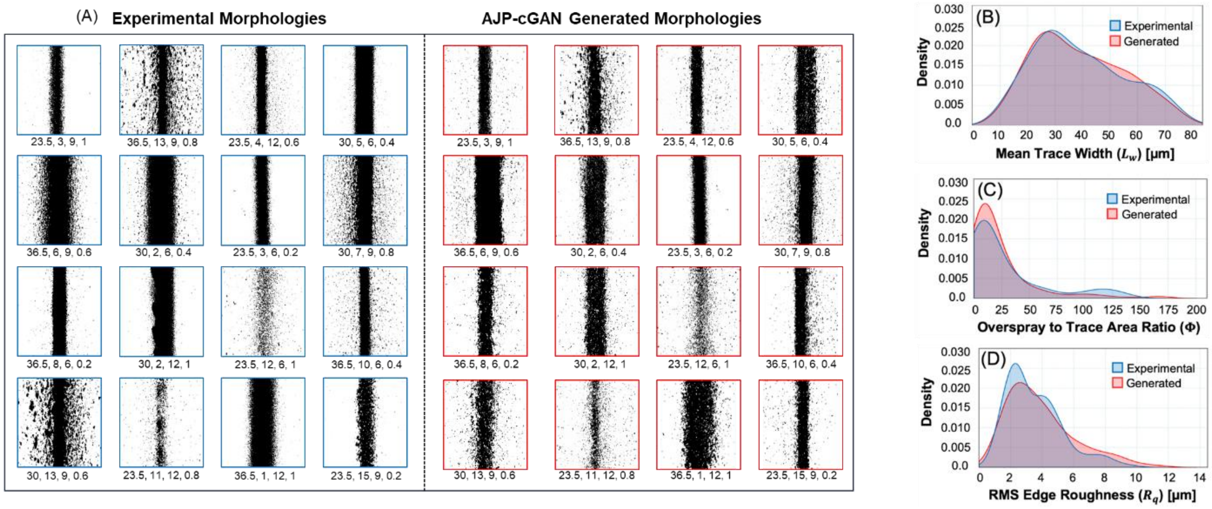

Figure S1: Comparison between experimental and cGAN-generated binary morphologies across 16 print conditions. (A) Visual comparison of real (left) and generated (right) images. (B–D)

Kernel density estimates showing alignment of experimental and predicted distributions for line width (L_w), overspray (Φ), and edge roughness (R_q).

4.2 cGAN Evaluation Across Process Parameters

The conditional GAN (cGAN) model was evaluated across varying levels of four key process parameters to assess its ability to generate morphology patterns that align with experimental observations. For each parameter—Focusing Ratio (FR), Carrier Gas Flow Rate (CGFR), Stage Speed (SS), and Atomizer Voltage (ATM)—all other parameters were held constant to isolate the effect of the variable under study. Binary morphology masks were used for both experimental and generated data to enable direct visual and quantitative comparison. Figures illustrate side-by-side comparisons and statistical plots for key morphological descriptors, with corresponding image similarity metrics shown as column charts.

4.2.1 Effect of Focusing Ratio (FR)

The cGAN model demonstrated high sensitivity to changes in FR. As FR increased, the generated morphologies exhibited progressive narrowing of printed traces, accurately mirroring the trend observed in experimental data, as illustrated in Figure S2(A). This is reflected in the line width (L_w) measurements in Figure S2(C), where both experimental and generated values decreased from ~68 µm to ~36 µm across the tested FR range. The model also replicated the non-monotonic behavior of overspray (Φ)—initially decreasing with FR due to improved collimation Figure S2(E), but increasing at higher values, likely due to flow instability. While edge roughness (R_q) showed moderate variability, the model captured its downward trend, highlighting its capacity to encode both global and local structural features, shown in Figure S2(E). High similarity scores in Jaccard Index, Precision, and F1 Score above 0.7 in Figure S2(B) confirm the model's strong alignment with experimental data across FR conditions.

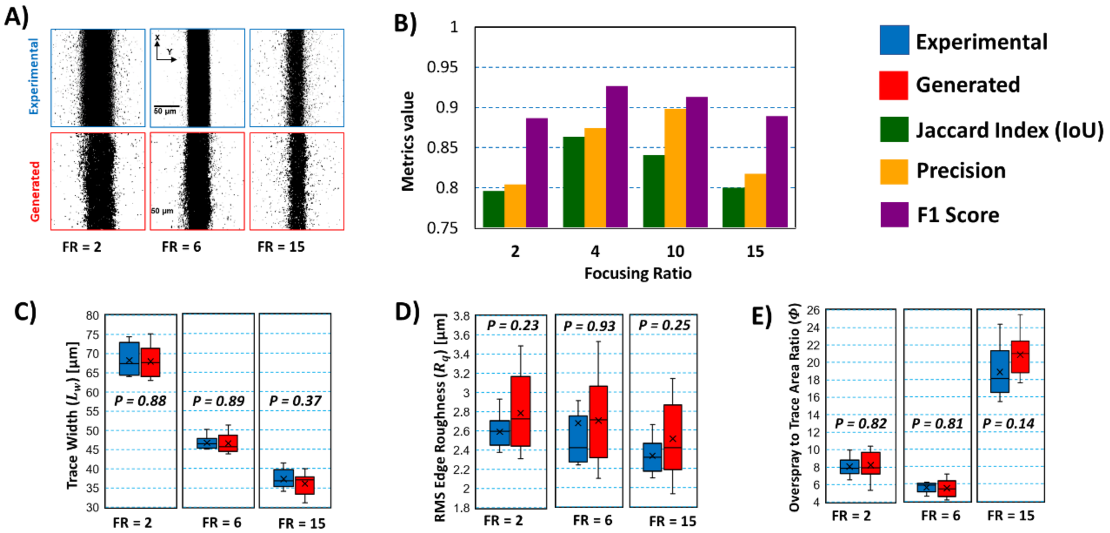

Figure S2: Comparison of experimental and cGAN-generated morphologies across varying FR, including (A) binary images, (B) image similarity metrics, and box plots for (C) line width (L_w), (D) edge roughness (R_q), and (E) overspray ratio (Φ).

4.2.2 Effect of Carrier Gas Flow Rate (CGFR)

The cGAN-generated outputs reflected the expected trend of increasing trace width with rising CGFR as illustrated in Figure S3(A), as more aerosolized droplets were transported into the deposition stream. Morphological metrics such as line width [Figure S3(B)] extracted from generated images were statistically indistinguishable from experimental counterparts across all CGFR levels. Overspray behavior also aligned well: Φ increased consistently [Figure S3(E)], a result of jet destabilization and secondary droplet formation at higher gas flow rates. The model effectively represented changes in both overspray spread [Figure S3(E)] and droplet-induced edge degradation [Figure S3(D)], with high correspondence in R_q values. Image similarity scores in Figure S3(B) remained stable across the CGFR range, supporting the robustness of the generator in modeling mass transport-driven morphology changes.

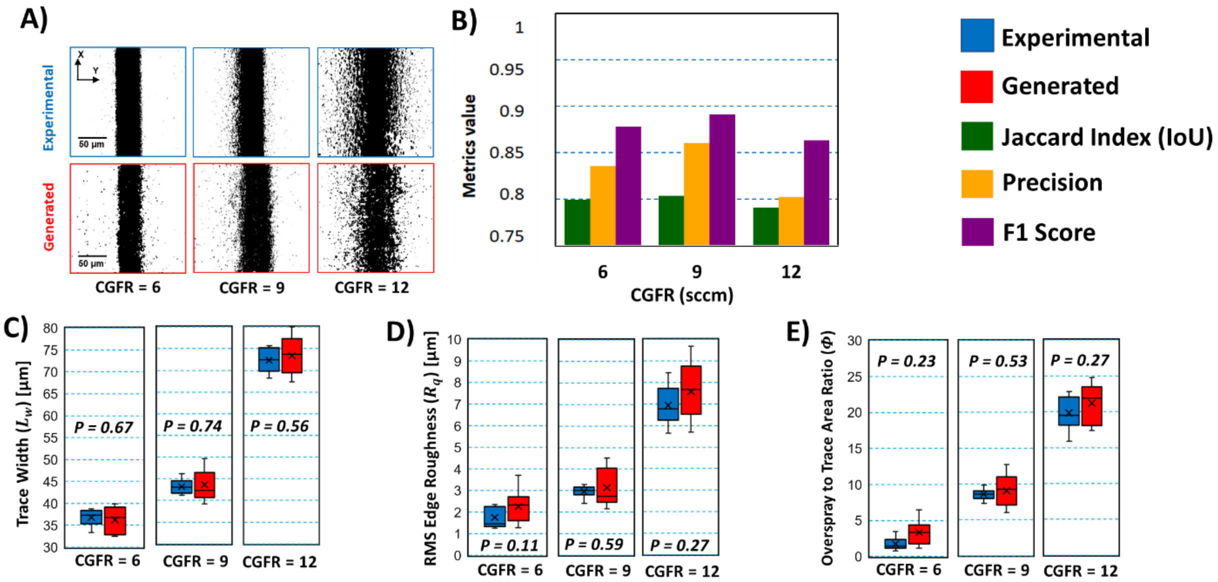

Figure S3: Comparison of experimental and cGAN-generated morphologies across varying CGFR, including (A) binary images, (B) image similarity metrics, and box plots for (C) line width (L_w), (D) edge roughness (R_q), and (E) overspray ratio (Φ).

4.2.3 Effect of Atomizer Voltage (ATM)

The cGAN was also evaluated across varying ATM values, which affect droplet size and atomization rate Figure S4(E). As voltage increased, generated traces became wider and more textured, aligning with the physical trend of increased material throughput, as shown in Figure S4(A). The model predicted both the broadening of trace width (L_w) [Figure S4(C)] and the corresponding degradation in edge smoothness (R_q) [Figure S4(D)]. The overspray droplet area (CA_{OS}) was also reasonably approximated, although spatial distributions showed slight deviations at higher voltages. Nevertheless, prediction accuracy remained high, with only minor reductions in similarity metrics in Figure S4(B) at extreme ATM levels, indicating that the generator retained its capability even under more complex atomization conditions.

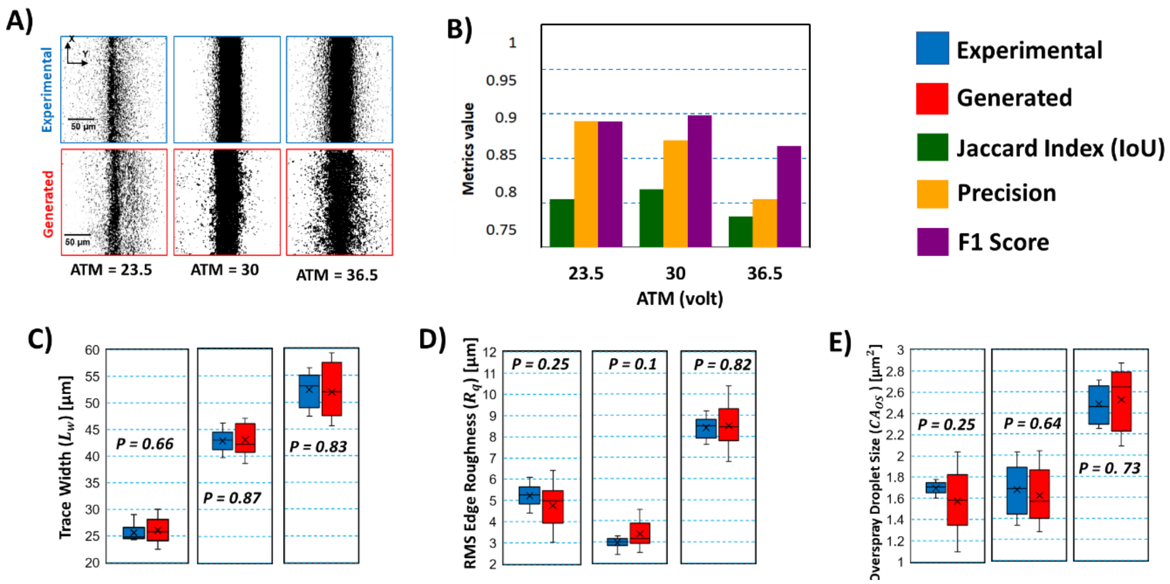

Figure S4: Comparison of experimental and cGAN-generated morphologies across varying ATM, including (A) binary images, (B) image similarity metrics, and box plots for (C) line width (L_w), (D) edge roughness (R_q), and (E) overspray ratio (Φ).

4.2.4 Effect of Stage Speed (SS)

Varying stage speed resulted in clear changes in generated morphology, particularly in material continuity and trace density. At lower speeds, generated traces were wide and continuous, while higher speeds led to thinner, fragmented lines, reflecting reduced droplet overlap, shown in Figure S5(A). The cGAN learned these spatial patterns and reproduced them with high fidelity. Continuity score (ϑ) in Figure S5(D) representing morphological segmentation, decreased with SS in both predicted and actual datasets. The model captured this trend well, suggesting that it effectively internalized the spatiotemporal relationship between stage velocity and trace formation in Figure S5(A & C). Across SS levels, similarity metrics in Figure S5(B) were consistently high, indicating precise prediction of both feature placement and structural transitions.

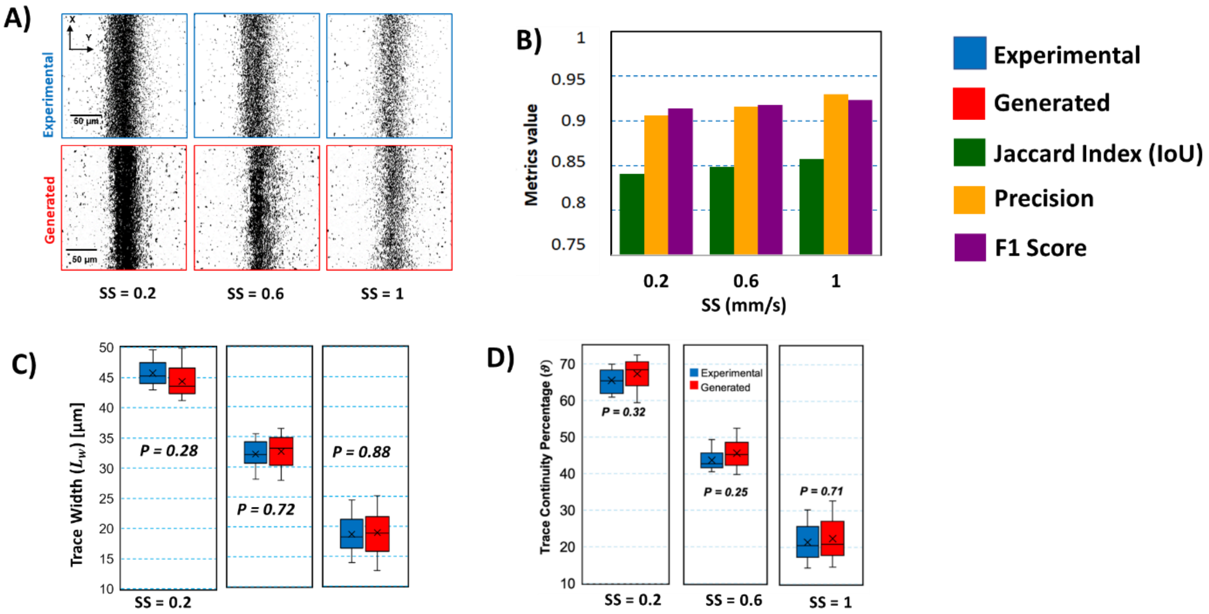

Figure S5: Comparison of experimental and cGAN-generated morphologies across varying SS, including (A) binary images, (B) image similarity metrics, and box plots for (C) line width (L_w), (D) trace continuity (ϑ).

4.3 cGAN prediction scenarios

Figure S6 illustrates three representative scenarios observed in the morphology prediction results generated by the residual conditional GAN model. In Scenario A, which accounts for approximately 80% of the cases, the model successfully generates deposit morphologies that closely match the corresponding experimental images in terms of line width, overspray distribution, and edge sharpness. This indicates effective learning of the underlying process-structure relationship and strong generalization across well-represented process conditions.

Scenario B, which accounts for approximately 18% of the generated outputs, reveals cases where the residual cGAN model over-predicts overspray and exhibits excessive black pixelation, deviating from the corresponding experimental observations. One key underlying issue in these cases may be partial mode collapse, a common failure mode in GANs where the generator starts producing similar or redundant outputs regardless of variations in the input noise vector z. This collapse limits the diversity of generated morphologies, leading to repetitive, artifact-laden patterns that fail to capture the full range of conditional variations — particularly when the training data is imbalanced or contains dominant classes such as high-overspray conditions. So, while the generator did not completely fail, it struggled to maintain diversity and realism under certain conditions, especially for high-overspray or underrepresented regions of the data distribution.

While the residual architecture itself offers advantages such as improved gradient flow and better preservation of spatial structure, it does not inherently prevent mode collapse. However, residual blocks can mitigate its severity by enabling deeper networks that learn more robust, fine-grained features — which helps retain subtle conditional variations across samples. In this context, residual cGAN serves as a more stable foundation, but additional mechanisms are necessary to fully address mode collapse.

To mitigate the risk of mode collapse during training, several architectural choices were intentionally integrated into the model design. The use of residual blocks within the generator played a crucial role in stabilizing training by improving gradient flow and enabling the preservation of spatial features across deeper layers. This structural enhancement allowed the model to learn more robust mappings between latent vectors and morphology outputs, thereby reducing the likelihood of collapse into repetitive or oversimplified patterns. Additionally, conditional embeddings for process parameters (ATM, CR, FR, PS) helped ensure that the generator responded meaningfully to different input conditions, further supporting output diversity. While these choices provided a degree of protection against full mode collapse, the model did not incorporate more targeted strategies such as mini-batch discrimination, feature matching, or Wasserstein loss with gradient penalty (WGAN-GP) — all of which have been shown to promote sample diversity and reduce generative redundancy (Frogner et al., 2015). As such, although complete mode collapse was not observed, the presence of repetitive overspray artifacts in a subset of outputs (Scenario B) suggests that partial mode collapse may still have occurred under certain conditions (Park et al., 2020). Future iterations of the model may benefit from incorporating explicit diversity-promoting mechanisms to further enhance robustness and morphological variability.

Finally, Scenario C (~2–3%) encompasses cases where the model generates deposit morphologies for experimentally unviable or underrepresented process parameter combinations. Although the predicted patterns appear structurally coherent and physically plausible in isolation, they do not correspond to any observed experimental outcomes. This behavior highlights the generator’s capacity to generalize beyond the training distribution, but also exposes a limitation: the model lacks an inherent understanding of physical feasibility (Saxena & Cao, 2024). In the absence of explicit constraints, the generator may produce morphologies that violate known process-material interactions, such as fluid flow dynamics, sintering behavior, or jet stability in AJP.

To address this, future work should explore the integration of physics-informed priors or hybrid modeling frameworks that embed domain knowledge directly into the generation process. For example, incorporating constraints derived from printability windows, material property limits, or simulation-derived feasibility maps can guide the model to produce only physically valid outputs. Additionally, introducing physically meaningful regularization terms or adopting hybrid models that combine data-driven learning with first-principles physics (e.g., differentiable simulation modules) can further improve the alignment between generated morphologies and real-world constraints. These enhancements would ensure that the model not only generates diverse and realistic patterns but also respects the underlying physical limitations of the printing process.

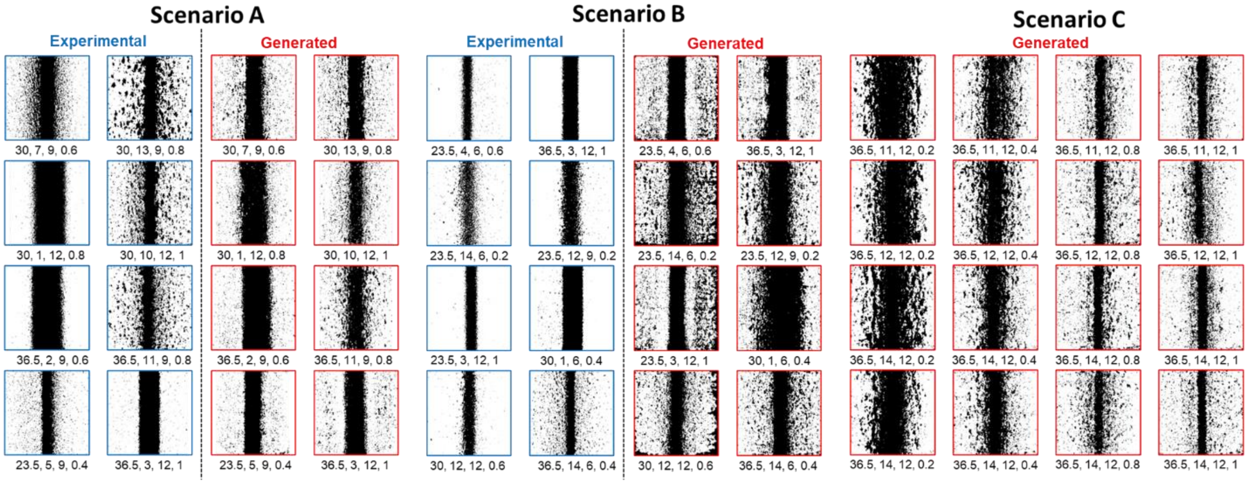

Figure S6: illustrates three prediction scenarios for AJP morphology generation.

5. Residual Conditional GAN Model

In conventional conditional GAN (cGAN) architectures, the generator learns to map a random noise vector and conditioning labels to realistic images. While effective in some cases, standard cGANs often suffer from training instability, vanishing gradients, and poor preservation of spatial structure — particularly when synthesizing high-resolution, fine-detailed images.

In our application — morphology generation for Aerosol Jet Printing (AJP) — these limitations are critical. Capturing precise features such as edge continuity, overspray, and line width requires stable learning and deep representations. Standard convolutional blocks in the generator struggle to maintain these fine-grained features over multiple upsampling layers.

To address this, we introduce a Residual Conditional GAN (Residual cGAN), where residual blocks are incorporated into the generator architecture (Xu et al., 2019). Each residual block learns a refinement over its input rather than a full transformation, allowing the network to reuse and preserve existing features. This design improves gradient flow, accelerates convergence, and enhances the quality and consistency of generated morphologies.

5.1 Residual Block Design

To address the limitations of standard convolutional blocks in deep generative networks, we incorporate residual blocks within the generator architecture (Park et al., 2022). In a standard block, the input undergoes a series of transformations represented as $F(x)$, and the output is entirely determined by this transformation. This structure lacks a mechanism to retain original feature information, which can lead to the degradation of critical spatial features and hinder effective gradient propagation, particularly in deeper networks (Khan et al., 2020). In contrast, a residual block introduces a skip connection that adds the original input x directly to the transformed output $F(x)$, resulting in a final output of $F(x) + x$. This formulation allows the network to learn a residual mapping — that is, to focus on the incremental refinement needed rather than reconstructing the entire output from scratch. As illustrated in Figure S7, the residual block structure facilitates improved information flow across layers, enhances training stability, and preserves key morphological features such as edge continuity and overspray in generated AJP patterns (Esmaeili et al., 2023). The use of residual blocks is particularly beneficial in tasks requiring fine-grained detail and structural consistency, offering both better convergence and higher-quality image synthesis.

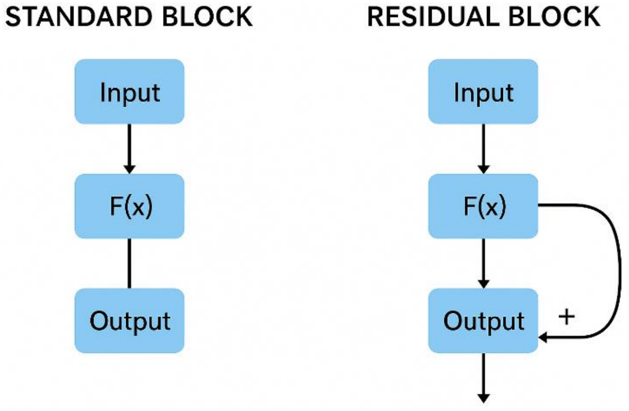

Figure S7: Architectural comparison between a standard block of cGAN (left) and a residual block (right).

5.2 Residual Architecture:

The Residual cGAN builds upon the standard conditional GAN architecture by integrating residual blocks into the generator network. The core idea is to improve training stability and feature preservation by enabling the network to learn residual functions instead of complete transformations. The model consists of a generator and a discriminator, both conditioned on process parameters: atomizer voltage (ATM), carrier gas rate (CR), flow rate (FR), and print speed (PS). The comparison of the cGAN Generator and Residual Block cGAN Generator is provided in Table TS1.

Table TS1: Architecture comparison of cgan generator and Residual Block cGAN Generator.

Component	Standard cGAN Generator	Residual Block cGAN Generator
Input	z + embedded labels	z + embedded labels
First Linear Layer	Linear → reshape to (B, 128, H, W)	Linear → reshape to (B, 128, H, W)
Upsampling Strategy	Upsample → Conv → BN → LeakyReLU (×2–3)	Upsample → Conv → BN → PReLU (×2), followed by residual refinement
Residual Block	Not present	Conv → BN → PReLU → Conv → BN + skip connection
Feature Flow	Sequential, each layer overwrites previous features	Additive refinement; original features preserved and enhanced
Depth Support	Limited by vanishing gradients	Deeper networks supported via improved gradient flow

Activation Function	LeakyReLU	PReLU (learnable)
Normalization	BatchNorm2d	BatchNorm2d
Output Layer	Conv2d (→ 1) → Tanh()	Conv2d (→ 1) → Tanh()
Final Output Shape	(B, 1, 256, 256)	(B, 1, 256, 256)
Training Stability	Moderate; sensitive to hyperparameters and learning rate	Improved; more stable due to residual connections

5.3 Residual cGAN Train Strategy:

The training strategy for the residual conditional GAN (cGAN) largely followed the standard cGAN setup, utilizing adversarial training with binary cross-entropy loss, conditional label embeddings, and the Adam optimizer with a learning rate of 0.0002 and betas (0.5, 0.999). Despite the deeper generator architecture introduced by residual blocks, no major changes were required in the training loop (Code available at GitHub repository). However, due to the increased model complexity, training required more computational resources, with each epoch taking approximately 3 hour and 20 minutes. The residual architecture contributed to more stable convergence and improved feature preservation, though care was taken to monitor for overfitting and maintain gradient stability through appropriate weight initialization and batch normalization.

6. Residual cGAN results:

Figure S8 illustrates the ability of the residual cGAN model to capture the morphological effects of individual AJP process parameters. By varying one parameter at a time while holding others constant, the model demonstrates sensitivity to key physical trends observed in experimental data. As SS increases, the generated deposit lines become progressively narrower and more uniform, reflecting the reduced material accumulation associated with faster substrate movement. Variations in flow rate FR result in visibly thicker lines and increased overspray, consistent with the higher volume of material delivered to the substrate. Changes in CGFR affect the focusing of the aerosol stream: lower CGFR produces sharper, more focused lines, while higher CGFR leads to broader and more dispersed morphologies. Finally, increasing ATM intensifies atomization and results in greater droplet dispersion, which the model captures through increasingly grainy and diffuse line structures. These outputs confirm that the residual cGAN successfully learns physically meaningful process-structure relationships and can serve as a valuable tool for simulating morphology evolution across a wide parameter space in AJP.

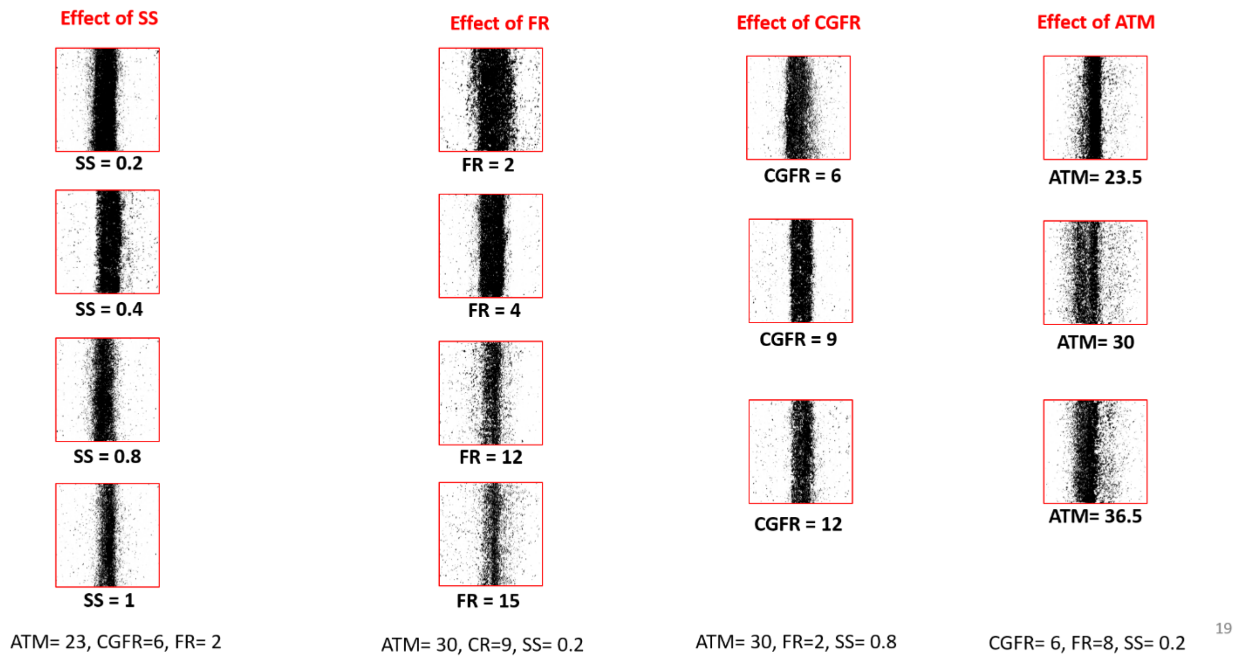

Figure S8: Residual cGAN-generated morphologies showing the effect of individual process parameters (SS, FR, CGFR, ATM). The model captures expected trends such as line narrowing with increased SS and overspray with higher FR, CGFR, and ATM, demonstrating its ability to learn process-structure relationships in AJP.

7. Challenges and Limitations

Despite the promising performance of the residual conditional GAN in capturing process-structure relationships in AJP morphology prediction, several technical and computational challenges were encountered during model development and experimentation.

First, due to time constraints and high training cost, the generated images were not subjected to comprehensive statistical, perceptual, or human-in-the-loop qualitative evaluation. While visual inspection confirmed morphological consistency across parameter variations, a more rigorous assessment—using metrics such as Fréchet Inception Distance (FID), Structural Similarity Index (SSIM), or downstream classification accuracy—remains to be performed in future work.

The training time per epoch was approximately 3 minutes and 20 seconds, which, when scaled to hundreds of epochs and parameter combinations, posed a significant computational bottleneck. Attempts to increase model expressiveness by incorporating RGB image channels (rather than binary grayscale inputs) further exacerbated memory usage and GPU load. Moreover, the RGB-based predictions appeared less smooth and more artifact-prone, likely due to increased complexity in learning three-channel representations without additional architectural modifications.

Additionally, initial experiments with StyleGAN-based architectures did not yield satisfactory results in this application. Although StyleGAN is known for high-fidelity image synthesis, it struggled to converge meaningfully on the AJP morphology data, potentially due to insufficient data volume, lack of continuous style-space correlation with process parameters, or incompatibility with discrete conditional labels.

Another significant challenge encountered was the integration of continuous-valued process parameters (e.g., stage speed, atomizer voltage, flow rates) into the generative framework. While conditional GANs are inherently designed to learn from discrete class labels, extending the conditioning mechanism to continuous variables requires architectural adaptations such as feature-wise modulation (FiLM), continuous embedding networks, or auxiliary regression-based objectives. These methods introduce additional complexity and instability in training, particularly when the mapping between continuous parameter values and morphology is nonlinear, non-monotonic, or sparsely sampled. To avoid these issues and ensure more stable conditioning, we discretized the process parameters into finite categorical levels based on known process windows and experimental resolution. While this approach facilitated robust embedding and convergence, it also introduced a degree of granularity loss in capturing fine variations across the process parameter space. Future work may address this limitation by implementing hybrid conditioning strategies that combine both discrete and continuous representations in a unified generative model.

These challenges underscore the trade-offs between model complexity, interpretability, and computational feasibility. They also highlight the need for future efforts focused on optimization of training efficiency, integration of physics-informed constraints, and hybrid generative approaches that balance realism with controllability and scalability.

8. Conclusion

In this work, we demonstrated the effectiveness of conditional Generative Adversarial Networks (cGANs), particularly a residual-enhanced cGAN architecture, in predicting deposit morphology in Aerosol Jet Printing (AJP) based on discrete process parameters. The residual cGAN exhibited improved visual fidelity, sharper edge definition, and better structural continuity compared to the standard cGAN, as observed through qualitative analysis. While a full quantitative evaluation is deferred due to computational constraints, the results indicate the model's strong potential in capturing meaningful process-structure relationships in printed morphologies.

We also identified three promising application scenarios for the trained residual cGAN model: (i) as an interactive visualization tool to support operators or decision algorithms in selecting suitable print parameters;

(ii) as a reference model within an in-situ monitoring system for detecting real-time deviations between predicted and actual deposit profiles; and

(iii) as a computationally efficient surrogate model to accelerate sequential design-of-experiment strategies such as Bayesian optimization.

Despite these promising outcomes, several challenges were noted. The integration of continuous conditioning variables proved unstable, prompting a shift to discretized parameter encoding. Moreover, the model faced limitations in training efficiency, generalization under sparse data, and prediction accuracy in experimentally unviable regions. Early trials using RGB images and StyleGAN architectures were also computationally intensive and yielded suboptimal results, while time constraints prevented rigorous statistical validation.

For future work, we recommend incorporating physics-informed constraints derived from governing principles of aerosol dynamics, droplet behavior, and substrate interactions to improve generalization and reduce artifact generation. Enhancing the model with temporal or signal-based data, such as real-time flow rates or droplet size distributions, may allow the capture of dynamic process effects. Expanding into three-dimensional morphology modeling using voxel grids or point cloud data, potentially with diffusion-based generative models, represents a promising direction for simulating volumetric structure evolution. Additionally, extending the output to include functional property predictions, such as electrical resistance or adhesion, will support a more comprehensive process–structure–function framework for intelligent, data-driven control in functional printing systems.

Reference:

Esmaeili, M., Abbasi-Moghadam, D., Sharifi, A., Tariq, A., & Li, Q. (2023). ResMorCNN model: hyperspectral images classification using residual-injection morphological features and 3DCNN layers. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *17*, 219-243.

Frogner, C., Zhang, C., Mobahi, H., Araya, M., & Poggio, T. A. (2015). Learning with a Wasserstein loss. *Advances in neural information processing systems*, *28*.

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, *53*, 5455-5516.

Park, S. W., Huh, J. H., & Kim, J. C. (2020). BEGAN v3: avoiding mode collapse in GANs using variational inference. *Electronics*, *9*(4), 688.

Park, S., & Shin, Y. G. (2022). Generative residual block for image generation. *Applied Intelligence*, 1-10.

Saxena, D., & Cao, J. (2021). Generative adversarial networks (GANs) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, *54*(3), 1-42.

Xu, K., Cao, J., Xia, K., Yang, H., Zhu, J., Wu, C., ... & Qian, P. (2019). Multichannel residual conditional GAN-leveraged abdominal pseudo-CT generation via Dixon MR images. *IEEE Access*, *7*, 163823-163830.