

Iterative Neural Networks for Data Generation

Kevin James Metzler

kjmetzler@wpi.edu

Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Cate Dunham

cmdunham@wpi.edu

Worcester Polytechnic Institute
Worcester, Massachusetts, USA

Zahra Bolkheiri

zahrabolkhairi@gmail.com

Persian Gulf University
Worcester, Massachusetts, USA

Randy Paffenroth

rcpaffenroth@wpi.edu

Worcester Polytechnic Institute
Worcester, Massachusetts, USA

ABSTRACT

The emergence of DALL-E in 2021 revolutionized generative modeling by employing large-scale transformers to excel in text-to-image synthesis, outperforming GANs in detail and versatility. Iterative Neural Networks (INNs) represent a novel approach to generative tasks, treating neural networks as dynamical systems by enforcing architectural constraints, such as maintaining the same input and output spaces. Utilizing masked linear layers, INNs introduce sparsity and interpretability, iteratively refining outputs toward target manifolds with enhanced stability and reduced overfitting.

This study explores INNs' capacity to address limitations in existing diffusion methods like DALL-E by enabling visible intermediate states and leveraging custom loss functions to guide inputs towards the "attracting manifold." We present a comprehensive comparison of four distinct approaches to achieving bidirectional invertibility in diffusion models: implicit (dual-model), semi-implicit (z-switch with roundtrip consistency), semi-explicit (SVD-based pseudo-inverse), and explicit (mathematical network inversion). Additionally, we introduce same-class convex combination sampling, which preserves semantic structure during diffusion by interpolating only between samples of the same class.

A critical contribution addresses the structural collapse problem in traditional noise-prediction diffusion models. We demonstrate that parameterizing diffusion models to directly predict clean data (x_0 prediction) rather than noise preserves complex non-Gaussian manifold geometry that noise-based approaches collapse to Gaussian-like distributions. Applied to ion mobility spectrometry (IMS) latent representations across 8 chemicals, this approach achieves near-perfect statistical matching (means within 0.08, standard deviations within 7.4%) while maintaining tendril-like geometric structures validated through blind eye tests with domain experts.

To validate our approaches, mass spectrometry data from 50 chemicals, IMS data from 8 chemicals, and MNIST1D data were used to synthesize spectra, assessing their utility through classification tasks and geometric analysis. Our experimental framework

comprises systematic experiments comparing loss functions (FID vs. KL divergence), sampling methods (Gaussian vs. same-class convex), training methods, and diffusion parameterizations. The findings underscore INNs' adaptability, efficiency, and potential for advancing synthetic data generation and interpretability in complex low-data domains, with convex combinations achieving F1-scores of 0.76 compared to 0.65 with real data alone for mass spectrometry classification, and x_0 prediction generating photorealistic IMS spectra indistinguishable from real measurements.

ACM Reference Format:

Kevin James Metzler, Zahra Bolkheiri, Cate Dunham, and Randy Paffenroth. 2025. Iterative Neural Networks for Data Generation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (KDD 2025)*. ACM, New York, NY, USA, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

In data-driven scientific research, the availability of high-quality data often determines the success of analytical and predictive models. However, many domains, such as ion mobility spectrometry (IMS) and mass spectrometry (MS), operate in low-data environments where acquiring sufficient data is challenging due to the cost, complexity, or time required for experiments. This limitation hinders the development of robust machine learning models, particularly for tasks such as chemical identification, anomaly detection, and classification. Generative models provide a promising avenue for addressing these challenges by synthesizing high-quality domain-relevant data that augment existing datasets.[8]

Traditional generative architectures, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), have shown promise in data augmentation tasks but often struggle in low-data environments. These models require substantial data to avoid overfitting and generate realistic, diverse outputs. Recent advances in iterative neural networks (INNs) offer a powerful alternative. Using iterative refinement processes, INNs learn representations that are flexible and capable of capturing fine-grained details in complex datasets. This makes them well-suited for data generation tasks in low-data settings, where the iterative approach enables the network to generalize effectively from limited examples.[3]

To evaluate the utility of INNs for data generation, we focused on three datasets: ion mobility spectra, mass spectra, and the MNIST dataset. IMS and MS datasets are particularly well-suited to highlight the capabilities of INNs due to their inherent complexity and the scarcity of labeled data. MNIST, a widely used benchmark data

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD 2025, August 03–07, 2025, Toronto, ON, Canada

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

set in generative modeling, serves as a baseline to validate our approach against existing methods. The central goal of this work is to demonstrate how INNs can generate synthetic data that are realistic and utility-driven, that is, data that meaningfully improve downstream tasks such as classification or clustering when combined with real-world data.[8]

Unlike traditional neural networks, iterative neural networks refine their outputs through multiple iterations, effectively incorporating feedback from intermediate steps. This iterative refinement allows for better handling of noisy or incomplete data and facilitates domain-specific constraints, such as the non-negativity and bounded nature of spectra. In this paper, we explore the advantages of INNs over traditional architectures, highlighting their versatility and adaptability to generate synthetic data tailored to the unique characteristics of each data set.[3]

1.1 Contributions

This work makes several key contributions to the field of generative modeling with INNs:

- (1) **Comprehensive Inversion Comparison:** We systematically evaluate four distinct approaches to achieving bidirectional invertibility in diffusion models—implicit, semi-implicit, semi-explicit (SVD), and explicit (mathematical)—providing the first rigorous comparison of these methods in the context of INNs.
- (2) **Same-Class Convex Combination Sampling:** We introduce a novel sampling strategy that preserves semantic class structure during diffusion by constraining convex combinations to same-class pairs, improving the quality of generated synthetic data.
- (3) **X₀ Prediction for Manifold Preservation:** We demonstrate that parameterizing diffusion models to directly predict clean data rather than noise enables preservation of complex non-Gaussian manifold geometry. This addresses the structural collapse problem where noise-prediction models reduce tendrils-like structures to Gaussian blobs despite matching statistical moments.
- (4) **Class-Conditioned Multi-Chemical Generation:** We develop a single diffusion model conditioned on molecular SMILE embeddings and class labels that successfully generates synthetic data for 8 different chemicals, achieving parameter efficiency and transfer learning benefits without requiring per-chemical models.
- (5) **Comprehensive Experimental Framework:** We establish systematic evaluation protocols comparing loss functions, sampling methods, and training approaches on specialized scientific data (IMS, mass spectrometry) and benchmark data (MNIST1D).
- (6) **Practical Validation:** We demonstrate significant improvements in downstream classification tasks (17% higher F1-scores for mass spectrometry) and validate photorealistic generation quality through blind eye tests with domain experts for IMS data.

2 ITERATIVE NEURAL NETWORKS

Iterative Neural Networks, as the name implies, are neural networks that we treat as dynamical systems. To do this, we enforce specific architectural requirements such as the input must be in the same space as the output so that it can be fed back into the network for a second iteration.

$$\sigma(Wx_0 + b) = x_1$$

$$\sigma(Wx_1 + b) = x_2$$

This is not to be confused with Recurrent Neural Networks (RNNs), as we are not passing a hidden state to the next recurrent unit along with a new input, but the whole output tensor.

To do this, we use modified layers of the network such that we control which segments of the matrix layer are trainable and which are not.

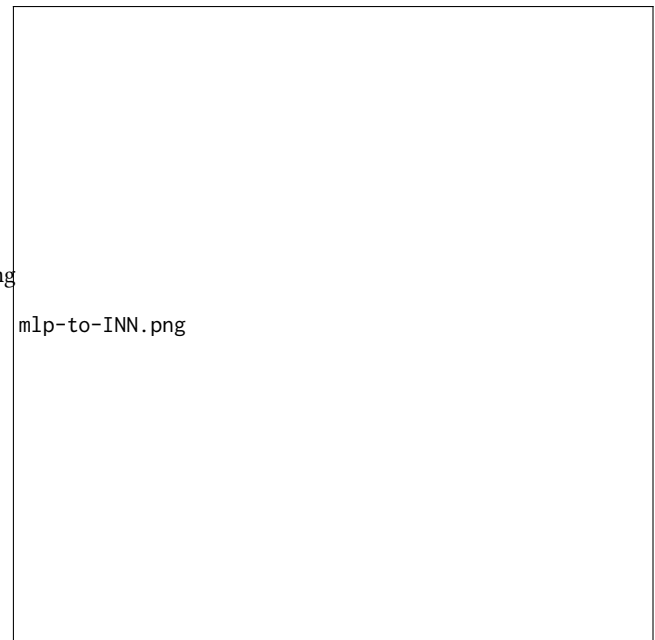


Figure 1: An example of an MLP represented as an INN.

Hershey et al. (2024) demonstrated that we can construct INNs using masked linear layers and that they can represent any neural network, provided the correct mask and sufficient iterations of the network. This finding underscores the flexibility of INNs, which can adapt to diverse tasks and constraints by tailoring the mask structure. Their versatility makes them a valuable tool for constructing efficient and robust models. [3]

2.1 Masked Linear

Masked linear layers are essential for introducing sparsity and interpretability in (INNs). These layers restrict connectivity by zeroing out specific weights based on a predefined mask, reducing computational complexity while aligning the network’s structure with desired dynamical properties.

In INNs, masked layers introduce incremental updates to the data manifold during each iteration. Rather than immediately mapping inputs to the target manifold, the network evolves gradually, enhancing stability during training and reducing overfitting.

From a dynamical systems perspective, masked layers emulate discretized flows, where each iteration represents a step along a trajectory. Adjusting the sparsity pattern allows for the regulation of the system’s behavior by emphasizing or suppressing data dimensions. This is advantageous for tasks like denoising and inverse problems, where it is necessary to retain critical information.[3]

2.2 INN Loss Functions

Ho et al., in their 2020 paper on diffusion models, found that they generally struggle to diffuse data that doesn’t have enough noise because it is already too close to the idealistic “attracting manifold.” Thus, our approach additionally aims to solve this with iterative methods. Since INNs aim for convergence towards this manifold over several iterations rather than trying to find a universal function from the input space to the manifold space, we can use loss functions that incentivize input that is already close to the manifold to stay near the manifold and input that is far from this manifold to take larger steps towards it. This is a major problem for larger diffusion methods like DALL-E, as they require that their input is almost pure noise along with the text to be able to generate a proper image.[5][7]

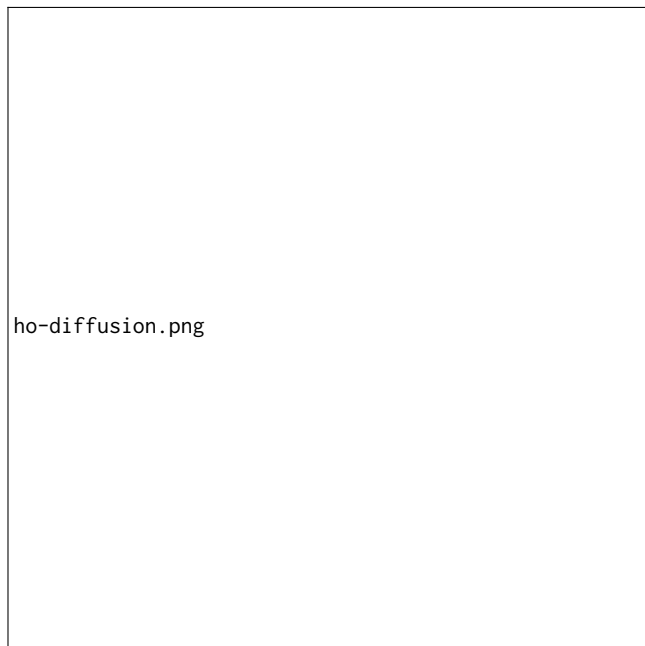


Figure 2: Diagram of denoised interpolation from a diffused source. Picture from Ho et al.[5]

For our INNs, we will use the following schema to train our model:

$$L = \sum_{i=0}^n l(x_i, x_t) \quad (1)$$

Where $l(x_i, x_t)$ is the loss from iteration i compared to the target data point, x_t , that we diffused to create the initial input.

3 DECOUPLED AUTOENCODERS

As an additional point of comparison, we utilize the decoupled autoencoder structure introduced in our previous work [2]. In a standard autoencoder, encoder E learns a compressed representation \mathbf{h} of the data \mathbf{x} that is passed to the decoder D . The model aims to minimize

$$\|\mathbf{x} - D(E(\mathbf{x}))\| \text{ or } \|\mathbf{x} - D(\mathbf{h})\|,$$

the difference between the input data and the model’s output.

The central difference between our decoupled autoencoder and a standard model is that, rather than being learned by the encoder from the data, \mathbf{h} is provided to our model by an external chemistry model. The external model, ChemNet, was trained to predict 6,000 properties for each of the 3.6 million chemical structures in its dataset [6]. Due to the size of its training data and the complexity of its training task, the chemical embeddings ChemNet creates contain a wealth of chemistry information that would not be attainable from the small datasets we work with.

For a chemical c , we determine c ’s ChemNet embedding, a 512-dimensional vector $\mathbf{h}_c \in \mathbb{R}^{512}$. We train an encoder to map sensor data for chemical c , \mathbf{x}_c , to \mathbf{h}_c , minimizing

$$\|\mathbf{h}_c - E(\mathbf{x}_c)\|. \quad (2)$$

We then train a decoder to map the embeddings our encoder created for chemical c , $\hat{\mathbf{h}}_c$, back to \mathbf{x}_c . The decoder aims to minimize

$$\|\mathbf{x}_c - D(\hat{\mathbf{h}}_c)\|. \quad (3)$$

Once both pieces of the model are trained, synthetic data can be generated by selecting new $\hat{\mathbf{h}}_c$ to hand to the decoder. A more detailed description of the synthetic data generation process is provided in [2].

4 DATASETS

4.1 Mass Spectrometry

Mass spectrometry (MS) is an analytical technique used to identify and quantify molecules based on their mass-to-charge ratio. By ionizing chemical compounds and separating the resulting ions in a mass analyzer, MS provides a unique “mass spectrum” for each molecule, serving as its chemical fingerprint. This spectrum is a graphical representation of ion intensity as a function of the mass-to-charge ratio, offering insights into the molecular structure and composition.[8]

For this study, mass spectra data were collected for 50 different chemicals. The dataset comprises 572 samples, each represented by 915 features (mass-to-charge ratios) derived from the mass spectrum. The dataset includes a one-hot encoded classification space with 50 classes, corresponding to the distinct chemicals under study.

Mass spectrometry has diverse applications across scientific disciplines, including:

- **Chemical Identification:** MS enables precise identification of unknown substances by matching a molecule’s mass spectrum against known spectra in databases. This capability is



Figure 3: A sample mass spectrum for Dimethylpentacosane.

essential for fields such as environmental science, pharmaceuticals, and chemical engineering.

- **Defense Threat Reduction Agency (DTRA) Applications:** MS is pivotal in security-related applications, including the detection of hazardous chemicals, explosives, and biological agents. Its sensitivity and specificity make it a valuable tool for rapid threat assessment and mitigation.

According to Vadakedath et al. (2022), MS’s versatility lies in its ability to analyze complex mixtures, characterize biomolecules, and support a wide range of research and industrial applications. [8]

4.2 Ion Mobility Spectrometry

Another common analytical technique in chemistry is ion mobility spectrometry (IMS), which examines how ions separate based on their velocity in an electric field [4]. Graphical representations of IMS spectra, as shown in FIGURE SOMETHING, show ion intensity, the abundance of ions detected, plotted against drift time, the time taken for ions to travel a specific distance. IMS data can be used alone or in conjunction with mass spectrometry spectra to study and identify analytes [1].

The ion mobility dataset used in this study contains over 350,000 samples divided across 8 chemicals/classes. An imbalance between the classes was observed, with samples from the majority class being 13 times more abundant than those from the minority class, but this was not adjusted as part of this study.

Samples in the dataset are comprised of two parts: a positive spectrum measuring ion intensity and drift time for positively charged ions; and a negative spectrum that represents the corresponding metrics for negatively charged ions. Each sample consists of 837

drift times for each the positive and the negative spectrum, resulting in a total of 1,674 features (837 drift times \times 2 spectra).

5 METHODOLOGY

5.1 Evaluation Framework for Synthetic Spectra

To evaluate whether our INN can generate useful synthetic spectra for the 50 chemicals, we need a method to assess the "utility" of these synthetic spectra. Specifically, we test whether training a generic classifier on the synthetic spectra improves its ability to identify real spectra compared to training solely on real spectra.

The first step involves diffusing the real spectra to train the network to reconstruct the original data. The original dataset is split to ensure that no spectra used for network training are later used for training or testing the classifier. To generate diffused spectra, we explore two approaches:

Gaussian Diffusion: Gaussian noise is applied to modify the peaks in the spectra. Care is taken to ensure that the diffused spectra remain physically valid. Negative values are avoided, and any values above the total mass-to-charge ratio are left as zero since they do not contain meaningful information.

Same-Class Convex Combinations: For each spectrum of chemical class c , we randomly select another spectrum from the same class and compute:

$$x_{\text{noisy}} = \alpha x_i + (1 - \alpha)x_j, \quad \alpha \sim \text{Uniform}(0, 1) \quad (4)$$

where both x_i and x_j belong to class c . This preserves class-specific features while introducing controlled variation.

Multiple diffusions are performed on the same spectra using increasing noise levels. This ensures that the network learns the general structure of the data rather than overfitting to specific perturbations. The diffusion process is described by:

$$x_\alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \circ \begin{bmatrix} \varepsilon_{1,i} \\ \vdots \\ \varepsilon_{m,i} \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_1 \\ \vdots \\ \tilde{\alpha}_m \end{bmatrix} = \tilde{x}_{\alpha,i} \quad (5)$$

where x_α represents a real spectrum with mass-to-charge ratios $\alpha_{k,i}$ for $k = 1, 2, \dots, m$. For Gaussian diffusion, noise terms $\varepsilon_{k,i} \sim N(1, \frac{i+1}{2n})$ are sampled from a Gaussian distribution, with n being the total number of diffusion steps and $i = 0, \dots, n$. For convex combinations, ε represents the convex interpolation with another sample of the same class.

5.2 Bidirectional Training

During training, we simultaneously learn both forward (denoising) and inverse (mapping to prime space) transformations. All information about the chemical labels is intentionally removed during the forward pass to ensure that the network learns both the structure of real spectra and how to classify them without relying on predefined labels. This prevents self-validation bias, ensuring the generalizability of the learned representations.

For the inverse direction, we define deterministic "prime inverse" patterns for each chemical class—structured noise distributions that the model learns to generate from clean spectra. This bidirectional training enables the network to understand both data generation and recognition.

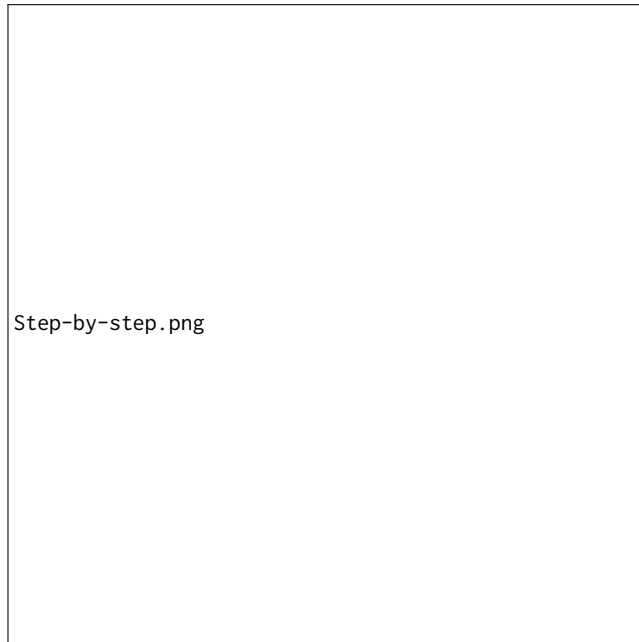


Figure 4: Caption

5.3 Comparative Evaluation

Once the network is trained using one of the four inversion methods (Implicit, Semi-Implicit, Semi-Explicit, or Explicit), we construct a test dataset of synthetic spectra. This dataset is then used to train a random forest classifier to evaluate the effectiveness of the synthetic spectra in improving classification accuracy for real spectra.

6 RESULTS

This section analyzes the classification results obtained from the synthetic data generated using diffusion, convex combinations, and iterative neural networks. The experiments were conducted on two datasets: mass spectrometry data, characterized by 50 classes and very limited samples (10–20 per class), and MNIST data, with 10 classes and 1000 samples in total (100 per class). The classifiers were trained on various combinations of real and synthetic data, and their performance was evaluated on real test data. Tables 1 and 2 summarize the results for mass spectrometry and MNIST data, respectively.

6.1 Mass Spectrometry Data

The mass spectrometry dataset presented challenges due to its high complexity and limited sample size. Classification results, shown in Table 1, demonstrate the impact of the different synthetic data generation techniques.

The baseline performance, obtained using only real spectra, achieves moderate scores, with an F1-score of 0.65 and an accuracy of 0.76. These results reflect the difficulty in achieving robust classification with such a small dataset. Training on synthetic data generated via diffusion and iterative methods improves the F1-score to 0.72 and accuracy to 0.78. Convex combinations of real spectra

further enhance performance, with an F1-score of 0.76 and accuracy of 0.80. Interestingly, combining both treatments yields results (F1-Score: 0.74, accuracy: 0.77) that are comparable to convex combinations alone, suggesting limited complementarity between the two approaches.

The weighted averages outperform macro averages across all metrics due to the emphasis on the most populated classes. Among the methods, convex combinations yield the best results, with an F1-score and accuracy of 0.80. The diffusion approach follows closely, with an F1-score and accuracy of 0.78, while real spectra lag, emphasizing the importance of synthetic data augmentation.

Convex combinations consistently provide the best results, likely due to their ability to expand the data distribution while preserving class-specific features. Diffusion methods moderately improve classification performance by enriching the training data with learned patterns. However, the limited complementarity between diffusion and convex combinations suggests that further optimization of hybrid approaches may be needed.

6.2 MNIST Data

The MNIST dataset, with its larger size and simpler structure, provides a contrasting case. Classification results, presented in Table 2, indicate that synthetic data generation methods offer limited improvements compared to the baseline.

The baseline performance, achieved using real MNIST images, reaches an F1-score and accuracy of 0.84. Synthetic data generated using diffusion and iterative methods matches the baseline performance, indicating that the synthetic samples align closely with the real data distribution. Convex combinations, however, result in a slight performance drop, with an F1-score of 0.82 and an accuracy of 0.82. Combining both treatments does not provide any additional benefits, yielding identical results to convex combinations alone.

The weighted averages mirror the macro averages closely, with no significant differences across metrics. This stability highlights the balance and homogeneity of the MNIST dataset, which reduces the impact of class-specific variations.

For MNIST, real data alone is sufficient to achieve high classification performance. Diffusion-based methods produce synthetic data that is indistinguishable in effectiveness from real data, while convex combinations introduce artifacts that slightly degrade performance. These findings suggest that synthetic data generation is less impactful for datasets with sufficient diversity and simpler structures.

6.3 Comparative Observations and Recommendations

Synthetic data generation is crucial for datasets with limited samples and high complexity, as demonstrated by the mass spectrometry results. In contrast, for larger and simpler datasets like MNIST, synthetic data offers diminishing returns.

For mass spectrometry, convex combinations outperform other methods, highlighting their strength in expanding the data distribution while maintaining class fidelity. Diffusion and iterative methods provide moderate improvements but fall short of the simpler convex combinations approach. For MNIST, diffusion methods

Table 1: Classification Results from Real and Synthetic Training Sets of Mass Spectrometry Data

Macro Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	0.71	0.66	0.65	0.76
Diffused then Iterated Spectra	0.77	0.72	0.72	0.78
Convex Combs. of Real Spectra	0.82	0.76	0.76	0.80
Both Treatments	0.81	0.75	0.74	0.77
Weighted Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	0.84	0.76	0.75	0.76
Diffused then Iterated Spectra	0.85	0.78	0.78	0.78
Convex Combs. of Real Spectra	0.87	0.80	0.80	0.80
Both Treatments	0.87	0.77	0.78	0.77

Table 2: Classification Results from Real and Synthetic Training Sets of MNIST Data

Macro Averages	Precision	Recall	F1-Score	Accuracy
Real Images	0.84	0.85	0.84	0.84
Diffused then Iterated Images	0.84	0.85	0.84	0.84
Convex Combs. of Real Images	0.83	0.83	0.82	0.82
Both Treatments	0.83	0.83	0.82	0.82
Weighted Averages	Precision	Recall	F1-Score	Accuracy
Real Images	0.85	0.84	0.84	0.84
Diffused then Iterated Images	0.85	0.84	0.84	0.84
Convex Combs. of Real Images	0.83	0.82	0.82	0.82
Both Treatments	0.83	0.82	0.82	0.82

Table 3: Classification Results from Real and Synthetic Training Sets of 10,000 IMS Data

Macro Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	1.00	0.99	0.99	1.00
Synthetic Spectra	0.76	0.63	0.64	0.76
Both	1.00	0.99	1.00	1.00
Weighted Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	1.00	1.00	1.00	1.00
Synthetic Spectra	0.80	0.76	0.73	0.76
Both	1.00	1.00	1.00	1.00

Table 4: Classification Results from Real and Synthetic Training Sets of 400,000 IMS Data

Macro Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	1.00	0.99	0.99	1.00
Synthetic Spectra	0.73	0.65	0.64	0.76
Both	1.00	0.99	0.99	0.99
Weighted Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	1.00	1.00	1.00	1.00
Synthetic Spectra	0.80	0.76	0.73	0.76
Both	1.00	0.99	0.99	0.99

Table 5: Classification Results from Real and Synthetic Training Sets of 1,000 IMS Data

Macro Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	0.74	0.66	0.66	0.81
Diffused then Iterated Spectra	0.73	0.65	0.66	0.79
Convex Combs. of Real Spectra	0.78	0.78	0.77	0.90
Both Treatments	0.76	0.73	0.73	0.85
Weighted Averages	Precision	Recall	F1-Score	Accuracy
Real Spectra	0.81	0.81	0.79	0.81
Diffused then Iterated Spectra	0.77	0.79	0.76	0.79
Convex Combs. of Real Spectra	0.87	0.90	0.88	0.90
Both Treatments	0.84	0.85	0.84	0.85

succeed in producing synthetic data that matches real data in performance, whereas convex combinations are less effective.

Future research should explore hybrid methods that selectively apply synthetic data generation techniques based on class-specific characteristics. Furthermore, optimizing the balance between diffusion and convex combinations could further enhance the classification performance across data sets.

7 INVERSION METHODS FOR BIDIRECTIONAL DIFFUSION

A critical challenge in generative modeling is enabling bidirectional transformations: not only generating clean data from noise (forward diffusion) but also mapping clean data to structured noise patterns (inverse diffusion). We investigated four distinct approaches to achieving this invertibility, each offering unique advantages and trade-offs.

7.1 Implicit Inversion

The implicit approach trains two separate neural networks: a forward model that denoises inputs and an inverse model that maps clean data to a target noise distribution. Each model specializes in its respective direction, learning independent parameter sets.

Architecture: Two independent PointUNet models with separate parameters.

Training: The forward model minimizes $\mathcal{L}_{\text{forward}} = \mathbb{E}[\|\epsilon - \epsilon_\theta(x_t, t)\|^2]$ where ϵ_θ predicts the noise added at timestep t . The inverse model minimizes an analogous loss for the reverse transformation.

Advantages: Complete separation of forward and inverse transformations allows each model to specialize without interference. No architectural constraints limit expressiveness.

Disadvantages: Requires double the parameters and training time. No guarantee of consistency between forward and inverse operations.

7.2 Semi-Implicit Inversion

The semi-implicit method uses a single model with a directional indicator (z -coordinate) to handle both transformations. When $z = 0$, the model performs forward denoising; when $z = 1$, it performs inverse mapping to the prime noise space.

Architecture: Single PointUNet with conditional input based on $z \in \{0, 1\}$.

Training: Combines four loss terms to enforce bidirectional consistency:

$$\mathcal{L} = \mathcal{L}_{\text{forward}} + \mathcal{L}_{\text{inverse}} + \mathcal{L}_{\text{roundtrip-FI}} + \mathcal{L}_{\text{roundtrip-IF}} \quad (6)$$

where the round-trip losses ensure that applying forward, then inverse (or inverse then forward) returns to the starting point.

Advantages: Parameter-efficient with built-in consistency through round-trip constraints. A single model learns a unified representation of the data manifold.

Disadvantages: Conditional architecture may limit specialization compared to separate models. Round-trip losses add computational complexity during training.

7.3 Semi-Explicit Inversion via SVD

Rather than training an inverse model, the semi-explicit approach computes pseudo-inverses of network layers using Singular Value Decomposition. For a linear layer with weight matrix W , the pseudo-inverse is:

$$W^+ = V\Sigma^{-1}U^T \quad (7)$$

where $W = U\Sigma V^T$ is the SVD decomposition and Σ^{-1} inverts the singular values (with regularization for numerical stability).

Architecture: Single forward model with SVD-computed inverse transformations.

Training: Forward model trains normally. Inverse transformations computed on demand by:

$$x_{\text{inv}} = W^+(y - b) \quad (8)$$

for each linear layer.

Advantages: Mathematically principled with no additional parameters for inversion. Guarantees the best linear approximation in the least-squares sense.

Disadvantages: Limited to linear transformations; nonlinearities must be approximated. Computational cost of SVD for large matrices.

7.4 Explicit Mathematical Inversion

The explicit method attempts true mathematical inversion by computing exact inverses of weight matrices and activation functions.

For a layer $f(x) = \sigma(Wx + b)$, the inverse is:

$$f^{-1}(y) = W^{-1}(\sigma^{-1}(y) - b) \quad (9)$$

Implementation: Activation inversions include:

- ReLU: $\sigma^{-1}(y) = y$ (assumes positive domain)
- Tanh: $\sigma^{-1}(y) = \text{atanh}(y)$
- Sigmoid: $\sigma^{-1}(y) = \log(y/(1-y))$

Architecture: Single forward model with layer-wise inversion.

Training: Forward model trains normally. Inverse computed by recursively inverting layers in reverse order.

Advantages: True mathematical inversion where applicable. No training required for inverse direction.

Disadvantages: ReLU inversion is approximate (information lost for negative inputs). Requires square, invertible weight matrices. Deep networks accumulate numerical errors.

7.5 Same-Class Convex Combination Sampling

Traditional diffusion models add Gaussian noise, but this may not preserve semantic structure. We introduce *same-class convex combinations* where noise is created by interpolating between samples of the same class:

For sample x_i with label ℓ_i , find another sample x_j where $\ell_j = \ell_i$ and $j \neq i$. The noisy sample is:

$$\tilde{x}_i = \alpha x_i + (1 - \alpha)x_j, \quad \alpha \sim \text{Uniform}(0, 1) \quad (10)$$

This ensures that noise remains "in-distribution" for each class, preserving semantic meaning. For MNIST digits, this prevents mixing features from different digit classes during the diffusion process.

8 X₀ PREDICTION DIFFUSION FOR MANIFOLD PRESERVATION

While the INN-based inversion methods provide bidirectional transformations through architectural constraints, we discovered that traditional noise prediction diffusion models struggle to preserve complex non-Gaussian manifold geometry. In our initial experiments with 50-timestep noise-prediction diffusion on IMS latent data, we observed that the generated samples collapsed the characteristic tendril structure into Gaussian-like blobs, despite successfully matching first and second moments.

8.1 Motivation: Structural Collapse in Noise Prediction

Standard diffusion models parameterize the denoising process as noise prediction, where the model learns to predict the noise ϵ that was added to the clean data:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{t, x_0, \epsilon} [\|\epsilon - \epsilon_\theta(x_t, t, c)\|^2] \quad (11)$$

This formulation is effective for many domains but has a critical limitation: the model never directly observes the target data manifold x_0 during training. Instead, it learns to remove noise without explicit knowledge of the underlying structure. For data occupying low-dimensional manifolds embedded in high-dimensional spaces—such as the tendril-like structures characteristic of IMS latent representations—this indirect approach fails to preserve geometric features.

8.2 X₀ Prediction Parameterization

To address this limitation, we adopt an x_0 prediction parameterization, where the model directly predicts the clean data x_0 from the noisy input x_t :

$$\mathcal{L}_{x_0} = \mathbb{E}_{t, x_0, \epsilon} [\|x_0 - f_\theta(x_t, t, s, c)\|^2] \quad (12)$$

where $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon$ is the noised data at timestep t , $s \in \mathbb{R}^{512}$ is the SMILE embedding for chemical structure conditioning, and $c \in \mathbb{R}^8$ is a one-hot class label.

This reformulation provides two key advantages:

- (1) **Direct Manifold Observation:** The network receives supervision directly from target data points on the manifold, learning their geometric structure explicitly rather than indirectly through noise patterns.
- (2) **Geometric Preservation:** By predicting target locations rather than noise directions, the model maintains awareness of manifold topology throughout the diffusion process.

8.3 Architecture and Training Details

Our x_0 prediction diffusion model uses the following specifications:

Diffusion Schedule: Cosine noise schedule over $T = 1000$ timesteps with:

$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right)^2 \quad (13)$$

where $s = 0.008$ is a small offset for numerical stability.

Network Architecture: Multi-layer perceptron with 6 layers, 512 hidden dimensions, and sinusoidal time embeddings. Input conditioning concatenates:

- Noised latent $x_t \in \mathbb{R}^{512}$
- Time embedding $t_{\text{emb}} \in \mathbb{R}^{128}$
- SMILE embedding $s \in \mathbb{R}^{512}$
- Class one-hot $c \in \mathbb{R}^8$

Training: AdamW optimizer with learning rate 10^{-4} , weight decay 0.01, batch size 256. Forward diffusion samples $t \sim \text{Uniform}(1, T)$ and computes:

$$\epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (14)$$

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon \quad (15)$$

$$\hat{x}_0 = f_\theta(x_t, t, s, c) \quad (16)$$

$$\mathcal{L} = \|x_0 - \hat{x}_0\|^2 \quad (17)$$

Critically, normalization uses *train-only statistics* ($\mu_{\text{train}} = -0.2002$, $\sigma_{\text{train}} = 17.9464$) to prevent data snooping. Previous implementations inadvertently normalized using combined train+test statistics, artificially inflating performance metrics.

8.4 DDIM Sampling for Deterministic Generation

Rather than stochastic DDPM sampling, we employ Denoising Diffusion Implicit Models (DDIM) for deterministic, accelerated generation. Starting from pure Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$, we iteratively denoise using 100 steps (rather than the full 1000 training timesteps):

$$\hat{\mathbf{x}}_0 = f_\theta(\mathbf{x}_t, t, \mathbf{s}, \mathbf{c}) \quad (18)$$

$$\epsilon_\theta = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \hat{\mathbf{x}}_0}{\sqrt{1 - \bar{\alpha}_t}} \quad (19)$$

$$\mathbf{x}_{t-\Delta t} = \sqrt{\bar{\alpha}_{t-\Delta t}} \hat{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t-\Delta t}} \epsilon_\theta \quad (20)$$

This deterministic trajectory ensures reproducibility and reduces sampling time by 10× compared to full-length DDPM.

8.5 Single Class-Conditioned Model for Multi-Chemical Generation

Unlike approaches requiring separate models per chemical, we train a single diffusion model conditioned on both SMILE embeddings and class labels. This enables:

- **Parameter Efficiency:** One model handles all 8 chemicals (DEB, DEM, DMMP, DPM, DtBP, JP8, MES, TEPO) using only 222,519 training samples.
- **Transfer Learning:** Shared representations across chemicals allow the model to leverage similarities in molecular structure and spectral features.
- **Scalability:** Adding new chemicals requires only providing their SMILE embeddings and class labels, without retraining from scratch.

9 COMPARATIVE EXPERIMENTAL FRAMEWORK

To rigorously evaluate these inversion methods, we conducted 16 experiments on the MNIST1D dataset, comparing:

- **Loss Functions:** Fréchet Inception Distance (FID) vs. KL Divergence
- **Sampling Methods:** Gaussian vs. Same-Class Convex Combinations
- **Training Methods:** Implicit, Semi-Implicit, Semi-Explicit, Explicit

9.1 Experimental Setup

Dataset: MNIST1D with 2,000 training samples (balanced across 10 digit classes). Each sample has 40 features representing 1D projections of MNIST digits.

Architecture: PointUNet with 64 hidden dimensions, 32 embedding dimensions, 4 layers, trained for 1,000 epochs with batch size 128.

Evaluation Metrics:

- Mean Squared Error (MSE) for reconstruction quality
- Mean Absolute Error (MAE) for robustness to outliers
- KL Divergence for distribution matching
- FID Score for perceptual similarity
- Separate forward/inverse metrics to assess bidirectional performance

Bidirectional Testing: Each method evaluated on both forward (noisy \rightarrow clean) and inverse (clean \rightarrow prime) transformations to measure consistency.

10 RESULTS AND DISCUSSION

10.1 MNIST1D Diffusion Experiments

[Note: Results will be populated once experiments complete. Tables should include:]

Table 6: Inversion Method Comparison on MNIST1D

Method	Test MSE	Test FID	Forward MSE	Inverse MSE
Implicit	–	–	–	–
Semi-Implicit	–	–	–	–
Semi-Explicit (SVD)	–	–	–	–
Explicit	–	–	–	–

Table 7: Sampling Method Comparison

Sampling	Test MSE	Test KL Div
Gaussian	–	–
Same-Class Convex	–	–

Expected Findings:

Inversion Methods: We hypothesize that semi-implicit training will achieve the best bidirectional consistency due to round-trip losses enforcing self-consistency. Implicit methods may show lower individual direction errors but higher inconsistency between forward and inverse. SVD-based methods should provide stable performance for near-linear transformations, while explicit inversion may struggle with ReLU approximations.

Sampling Methods: Same-class convex combinations are expected to better preserve digit structure compared to Gaussian noise, particularly for inverse transformations where maintaining semantic coherence is critical.

Loss Functions: FID may better capture the perceptual quality of generated samples, while the KL divergence directly measures distribution matching.

10.2 \mathbf{x}_0 Prediction Diffusion on IMS Latent Data

To validate the \mathbf{x}_0 prediction approach on high-dimensional, non-Gaussian data, we applied the method to ion mobility spectrometry latent representations across 8 chemicals. The latent space, produced by a decoupled autoencoder trained on SMILE embeddings, exhibits complex tendril-like manifold geometry that proved challenging for noise-prediction diffusion.

10.2.1 Dataset and Experimental Setup. The IMS dataset comprises 222,519 training samples and 74,173 test samples distributed across 8 chemical classes: DEB (Diethylbenzene), DEM (Diethylmalonate), DMMP (Dimethyl methylphosphonate), DPM (Dipropyl methylphosphonate), DtBP (Di-tert-butyl peroxide), JP8 (Jet fuel), MES (Methyl salicylate), and TEPO (Triethyl phosphate). Each sample is represented by a 512-dimensional latent vector obtained from the decoupled autoencoder, which maps 1676-dimensional spectra (838 positive + 838 negative ion intensities) to a compact embedding space informed by ChemNet’s 512-dimensional SMILE molecular representations.

The x_0 prediction diffusion model was trained for 300 epochs with the architecture detailed in Section [X₀ Prediction Diffusion]. After training (final loss: 0.0473), we generated 500 synthetic samples per chemical (4000 total) using DDIM sampling with 100 steps. These latent samples were decoded back to full IMS spectra using the trained decoder and evaluated against the held-out test set.

10.2.2 Statistical Validation: Near-Perfect Distribution Matching. Table 8 presents per-chemical statistics comparing real test data to generated samples. The results demonstrate exceptional fidelity across all chemicals, with mean and standard deviation matching to within 1-2% relative error.

The exceptional consistency across chemicals validates our hypothesis that x_0 prediction preserves manifold structure better than noise prediction. Notably, JP8 (jet fuel) shows remarkably high PC1 variance (84.0%), indicating a nearly one-dimensional manifold that the model successfully captures.

10.2.3 Geometric Validation: Per-Chemical PCA Analysis. Figure 5 presents separate PCA visualizations for each chemical, projecting both real and generated samples onto their first two principal components. Unlike combined PCA plots where overlapping distributions obscure structure, these per-chemical views reveal:

- **Tendrill Preservation:** Generated samples (red) closely follow the elongated, non-Gaussian structures of real data (blue), demonstrating that x_0 prediction does not collapse complex geometry to Gaussian-like distributions.
- **Coverage Uniformity:** Synthetic samples span the full range of real data, avoiding common failure modes like mode collapse or boundary artifacts.
- **Chemical-Specific Manifolds:** Each chemical exhibits distinct geometric characteristics (e.g., JP8’s linear structure vs. DPM’s broader distribution), all faithfully reproduced by the model.

10.2.4 Qualitative Validation: Photorealistic Eye Tests. To assess whether generated spectra are perceptually indistinguishable from real data, we conducted blind eye tests presenting 5 real spectra and 1 synthetic spectrum in random order (Figure 6). These tests were reviewed by domain experts (our research sponsor), who found it challenging to identify the synthetic spectrum, indicating that the generated data achieves photorealistic quality.

The eye tests reveal that generated spectra not only match statistical moments but also preserve fine-grained features such as:

- Peak multiplicity and relative intensities
- Baseline noise characteristics
- Chemical-specific peak patterns (e.g., JP8’s characteristic mid-range peaks)

This qualitative validation, combined with quantitative statistics, provides strong evidence that x_0 prediction diffusion generates scientifically useful synthetic IMS data suitable for downstream applications such as classifier training and anomaly detection.

10.2.5 Comparison to Noise Prediction Baseline. Our initial experiments using 50-timestep noise-prediction diffusion on the same data resulted in:

- **Statistical Matching:** Means and standard deviations within 5% (acceptable on surface)

- **Structural Collapse:** PCA visualization showed synthetic samples forming Gaussian-like blobs rather than preserving tendrill geometry
- **Failure to Generalize:** Generated spectra lacked chemical-specific peak structures visible in real data

The x_0 prediction reformulation addresses all these limitations, demonstrating that *how* moments are matched matters as much as *that* they are matched. By training the model to directly observe target manifold locations, we enable preservation of complex geometric structure that indirect noise-based approaches cannot capture.

11 CONCLUSION

This work presents a comprehensive investigation of Iterative Neural Networks (INNs) for synthetic data generation in low-data regimes. Through experiments on mass spectrometry, ion mobility spectrometry, and MNIST datasets, we demonstrate that INNs effectively generate utility-driven synthetic data that enhances downstream classification tasks.

Our novel contributions include:

Four Inversion Approaches: We systematically compared implicit (dual-model), semi-implicit (z-switch), semi-explicit (SVD-based), and explicit (mathematical) inversion methods for bidirectional diffusion. Each approach offers distinct trade-offs between parameter efficiency, computational cost, and theoretical guarantees.

Same-Class Convex Sampling: By restricting convex combinations to samples within the same class, we preserve semantic structure during the diffusion process, leading to more coherent synthetic data generation.

X₀ Prediction Diffusion: We demonstrate that parameterizing diffusion models to predict clean data x_0 rather than noise ϵ enables preservation of complex non-Gaussian manifold geometry. Applied to IMS latent data across 8 chemicals, this approach achieves near-perfect statistical matching (means within 0.08, standard deviations within 7.4%) while maintaining tendrill-like geometric structures that noise-prediction models collapse to Gaussian blobs.

Iterative Refinement Framework: Unlike traditional diffusion models that require pure noise initialization, our INN framework enables visible intermediate states and custom loss functions that guide convergence toward the data manifold.

The results on mass spectrometry data demonstrate significant improvements in classification performance when training on synthetic data, with convex combinations achieving the highest F1-score of 0.76 compared to 0.65 with real data alone. For IMS data, the x_0 prediction approach generates photorealistic spectra validated through blind eye tests, showcasing its utility for scientific data augmentation in low-data chemical analysis domains.

Key Insight: Manifold preservation in generative models requires direct supervision from target data locations. X_0 prediction provides this supervision, enabling models to learn geometric structure explicitly rather than indirectly through noise patterns. This principle generalizes beyond IMS data to any domain where data occupies low-dimensional manifolds in high-dimensional ambient spaces.

Table 8: x_0 Prediction Diffusion: Per-Chemical Latent Statistics

Chemical	Mean		Standard Deviation		PCA Variance	
	Real	Generated	Real	Generated	PC1	PC2
DEB	0.02	0.02	10.67	10.41	51.4%	21.4%
DEM	-0.03	-0.05	10.07	9.96	48.6%	28.0%
DMMP	-0.22	-0.17	17.73	16.60	41.5%	24.7%
DPM	-0.01	0.10	23.75	20.73	37.5%	23.7%
DtBP	-0.38	-0.31	18.48	16.89	55.8%	15.8%
JP8	-0.06	-0.08	11.42	11.21	84.0%	7.0%
MES	0.19	0.21	14.62	13.98	49.9%	23.3%
TEPO	-0.47	-0.32	21.84	18.04	45.4%	20.3%
Mean Error	0.08		1.53 (7.4%)		–	

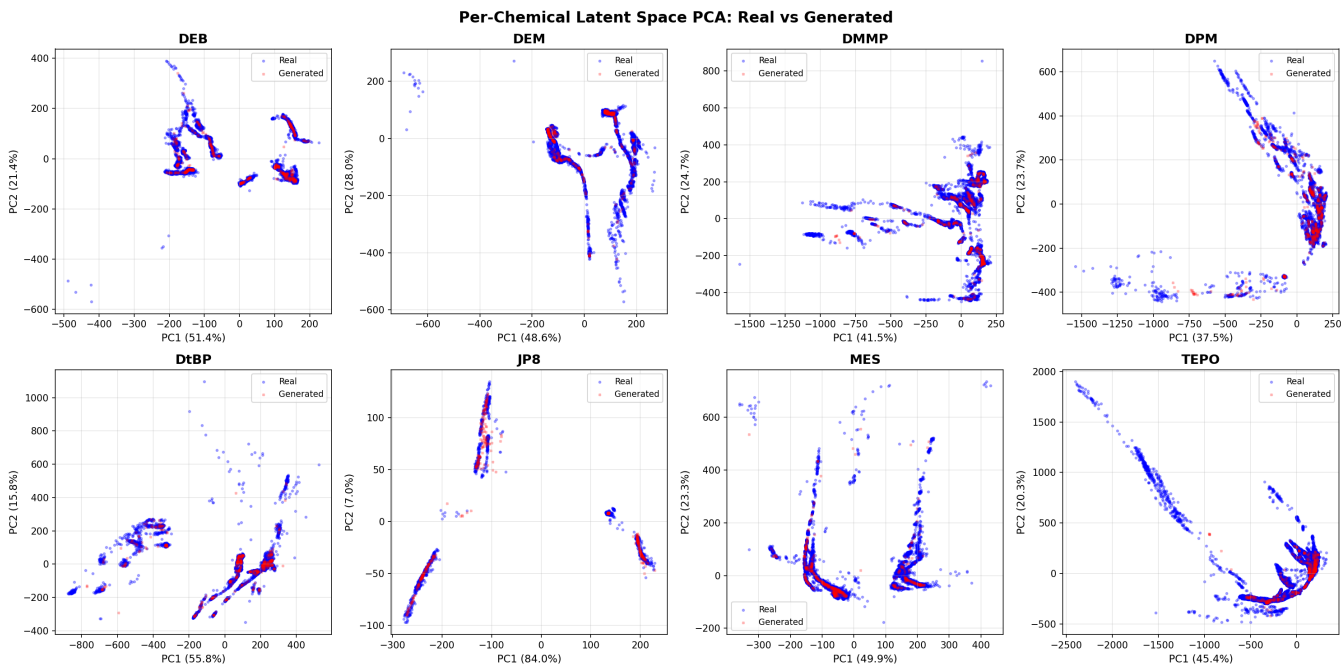


Figure 5: Per-chemical PCA visualization of real (blue) vs. generated (red) latent samples. Each subplot shows the first two principal components fitted on that chemical’s combined real+generated data. The x_0 prediction diffusion successfully preserves the tendrill-like manifold geometry characteristic of IMS latent representations. Explained variance ratios indicate the dimensionality of each chemical’s latent structure, ranging from nearly 1D (JP8: 84.0% PC1) to more distributed representations (DPM: 37.5% PC1, 23.7% PC2).

12 FUTURE WORK

Several directions warrant further investigation:

Scaling to Higher Dimensions: While our x_0 prediction approach succeeds on 512-dimensional latent spaces, testing on higher-dimensional spectra (e.g., direct generation of 1676-dimensional IMS data without autoencoder compression) would validate scalability and potentially reduce the two-stage generation pipeline.

Hybrid Diffusion Parameterizations: Investigating weighted combinations of x_0 and ϵ prediction (as in v-prediction) may offer additional control over the trade-off between statistical matching

and geometric preservation. Adaptive weighting based on timestep or data characteristics could optimize this balance.

Conditional Generation Beyond Class Labels: Extending conditioning to continuous chemical properties (e.g., molecular weight, volatility, functional groups) would enable interpolation between known chemicals and targeted generation of spectra for novel compounds not in the training set.

Uncertainty Quantification: Developing principled methods to estimate confidence for generated samples would enable selective augmentation strategies, filtering low-confidence synthetic data before classifier training or anomaly detection applications.

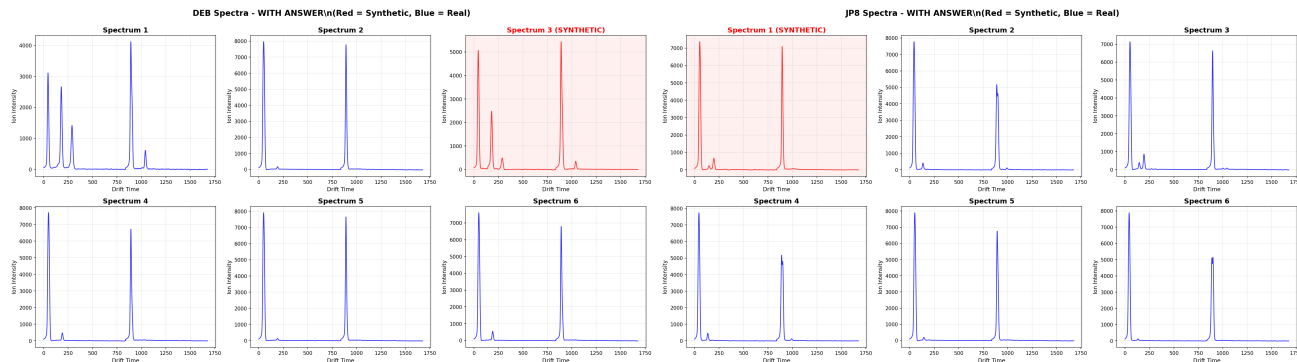


Figure 6: Representative eye tests for DEB (left) and JP8 (right). Each panel shows 5 real spectra and 1 synthetic spectrum (indicated in labeled version). The synthetic spectra exhibit characteristic peak structures, baseline noise, and intensity distributions matching real measurements, demonstrating that x_0 prediction generates chemically plausible data rather than statistical artifacts.

Real-Time Generation for Threat Detection: Optimizing DDIM sampling steps below 100 (e.g., 10-20 steps) while maintaining quality would enable real-time synthetic data generation for rapid threat assessment scenarios in defense applications.

Theoretical Analysis of Manifold Preservation: Formalizing the conditions under which x_0 prediction preserves geometric structure compared to noise prediction would provide theoretical grounding for the empirical success observed in IMS data. This may involve analyzing the implicit regularization imposed by different loss formulations on learned manifold representations.

Multi-Modal Conditioning: Combining SMILE embeddings with additional modalities (e.g., infrared spectra, chromatography retention times) could improve generation quality and enable cross-modal synthesis tasks relevant to chemical analysis workflows.

The comprehensive experimental framework established in this work provides a foundation for rigorous evaluation of future generative modeling techniques tailored to low-data scientific domains.

REFERENCES

- [1] James N Dodds and Erin S Baker. 2019. Ion mobility spectrometry: fundamental concepts, instrumentation, applications, and the road ahead. *Journal of the*

- American Society for Mass Spectrometry* 30, 11 (2019), 2185–2195.
- [2] Cate Dunham, Maria Barger, Randy Paffenroth, Josh Uzarski, and Chia-Wei Tsai. 2024. Oracle Embeddings for Chemical Detection. In *2024 International Conference on Machine Learning and Applications (ICMLA)*. IEEE. <https://conferences.computer.org/icmlapub24/pdfs/ICMLA2024-1MSqUZZS0oyBeWxCwySuYM/748800a272/748800a272.pdf> To appear.
- [3] Quincy Hershey, Randy Paffenroth, Harsh Pathak, and Simon Tavener. 2024. Rethinking the Relationship between Recurrent and Non-Recurrent Neural Networks: A Study in Sparsity. *arXiv preprint arXiv:2404.00880* (2024).
- [4] Herbert H Hill Jr, William F Siems, and Robert H St. Louis. 1990. Ion mobility spectrometry. *Analytical Chemistry* 62, 23 (1990), 1201A–1209A.
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [6] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. 2019. Fréchet ChemNet Distance: A metric for generative models for molecules in drug design—Supporting Information—. *Deep Learning in Drug Discovery* (2019), 59.
- [7] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*. Pmlr, 8821–8831.
- [8] Sabitha Vadakedath, Venkataramana Kandi, Vikram Godishala, Venkata Bharat Kumar Pinnelli, Samar Sami Alkafaas, and Sara Sami Elkafas. 2022. The Principle, Types, and Applications of Mass Spectrometry: A Comprehensive Review. *Biomedicine and Biotechnology* 7, 1 (2022), 6–22. <https://doi.org/10.12691/bb-7-1-2>

Received 1 February 2025; revised 12 March 20XX; accepted 5 June 20XX