VAE-GAN Hybrid Model for Generating Fusion Data

Alisa Yang (Equal Contribution)
Nick Sheft (Equal Contribution)
Jack Phillips (Equal Contribution)
May 3, 2025

1 INTRODUCTION & MOTIVATION

Fusion plasma data is very expensive and difficult to obtain. The general approach is to use a fusion simulator, whose performance can be hard to control and remains costly both computationally and financially. Thus, we explored the use of neural networks and deep learning methods, specifically a VAE-GAN hybrid model, to generate high-quality fusion data that matches the data produced by the simulator.

With our model, we aim to generate high-fidelity fusion data in a much more scalable and cost-efficient manner, enabling broader experimentation and faster research without relying heavily on expensive simulation runs. This data would influence the design of more performant nuclear reactors.

More broadly, our interest includes understanding the roles of diffusion and convection transport parameters in fusion plasma. These parameters are central to how energy and particles move within a reactor. They are driven by complex, non-linear, and dynamic processes that are difficult to model and often defy prediction. By developing high-fidelity generative models that can replicate these subtle behaviors, we take a step toward enabling deeper analysis of such phenomena in future work.

2 METHODOLOGY

2.1 Dataset

We used a fusion simulator called FlowNetTK developed by Jim Slone as part of the Mordijck lab to generate a 7.5GB dataset that contains 10,000 records of fusion data (distribution in **Fig. 1**). Each data sample consists of 6 channels or features describing various physical quantities across spatial and temporal grids:

1) Diffusion
2) Convection
3) Rho
4) Time
5) Density
6) Source (Source is the "fueling". Essentially, in order to sustain the experiment, we have to add some amount of plasma over time. This profile is the source. It is high at the edge as we fuel from the edge.)

Each sample has a spatial-temporal structure where the horizontal axis represents the spatial coordinate and the vertical axis represents the time evolution, creating a grid-like image per channel (a sample shown below in **Fig. 2**).

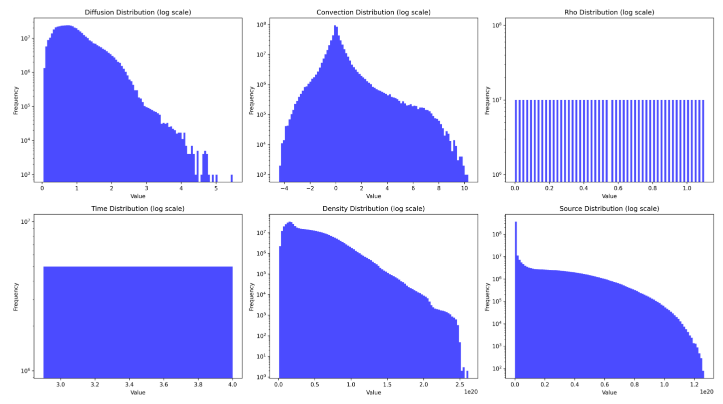

Fig. 1. Data distribution of All Channels

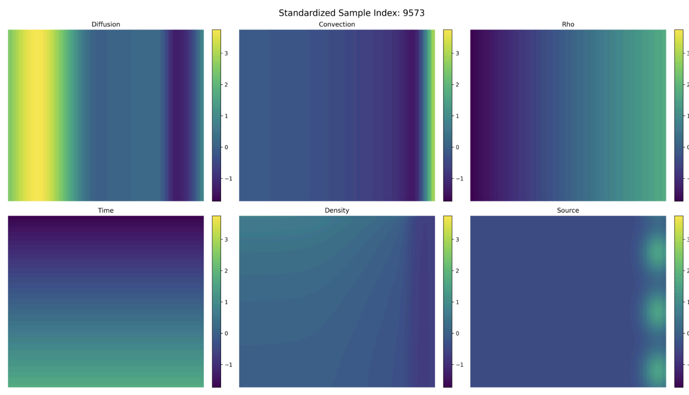

Fig. 2. A Sample of All Channels

2.2 VAE-GAN Hybrid Model

Fusion data exhibits complex spatial and temporal dependencies, making it challenging to model using conventional generative approaches. Traditional VAEs are effective at learning latent representations and generating new samples, but they often produce outputs that are overly smooth or blurry, as the VAE objective prioritizes reconstruction accuracy over visual fidelity.

On the other hand, GANs are well-known for generating highly realistic and sharp outputs by training a generator and a discriminator in an adversarial setup. However, GANs can be unstable to train, especially for high-dimensional and structured scientific data like fusion simulations. GANs also lack an explicit encoder, making it difficult to perform inference on real data or interpolate meaningfully between samples.

To address the limitations of standalone VAEs and GANs, we adopt a VAE-GAN hybrid model (illustrated in **Fig. 3**). This architecture combines the strengths of both:

- The VAE encoder learns a meaningful, structured latent space that allows for smooth interpolation and efficient sampling.
- The GAN discriminator pushes the decoder (generator) to produce outputs that are not only good reconstructions but also indistinguishable from real fusion data at a fine-grained level.

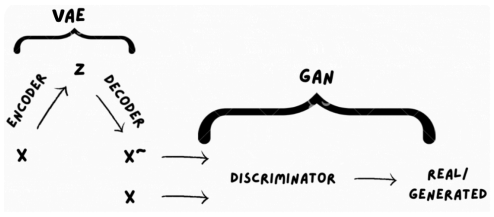

Fig. 3. VAE-GAN Architecture

During training, the model jointly optimizes a reconstruction loss, a KL divergence loss, and an adversarial loss. By leveraging both probabilistic modeling and adversarial training, the VAE-GAN hybrid model is particularly well-suited for generating realistic and diverse fusion datasets. This approach allows us to both synthesize high-quality synthetic fusion data and learn compact latent representations that can be useful for downstream tasks such as classification, anomaly detection, or simulation acceleration.

2.3 Data Preprocessing

After loading the raw fusion data, we observed large differences in scale across the channels. To normalize the input space and stabilize training, we standardized each channel separately to have mean = 0 and standard deviation = 1.

Initially, our dataset included Time and Rho channels, but we noticed that the model struggled to converge effectively: the discriminator would overpower the generator quickly and the overall loss balance and sample quality suffered. Following discussions and feedback from professors, we decided to remove Time and Rho from the training set. These two channels are relatively static and carry less dynamic information compared to Diffusion, Convection, Density, and Source. By removing these channels, the model learned to generate and discriminate more effectively.

2.4 Architecture

2.4.1 *Model Architecture*

1) **VAE "Generator"**: We used a Convolutional Variational Autoencoder (CVAE) rather than a simple fully connected VAE. Because our fusion data is structured as spatial-temporal grids, it can be naturally treated as images. Convolutional architectures are well-suited for preserving spatial patterns across both time and space, helping the model better capture realistic plasma behaviors.
    a. The **encoder** comprises four convolutional layers that progressively reduce the spatial-temporal resolution by performing downsampling operations. Each convolutional layer reduces the size of the feature maps through a combination of stride and padding, enabling the model to extract increasingly abstract features while compressing the data. A linear layer then compresses this representation into a latent space, parameterized by a mean and log-variance vector. During training, latent vectors are sampled using the reparameterization trick.
    b. The **decoder** reverses this process by first transforming the latent vector into a high-dimensional tensor using a fully connected layer, which is reshaped into a lower-resolution feature map. It then employs a series of four transposed convolutional layers to progressively upsample the spatial and temporal dimensions of the data. Each transposed convolution increases the resolution, gradually reconstructing the input dimensions while preserving spatial-temporal coherence. SELU activation functions are applied after each layer to ensure stable and non-linear transformations. This structure enables the decoder to generate realistic fusion data that closely mimics the original simulation outputs. Specifically:

- A linear layer first projects the latent vector into a high-dimensional feature space ([batch_size, num_channels * 16, 63, 4]).
- This reshaped tensor is then passed through a stack of ConvTranspose2d layers that upsample the spatial-temporal dimensions step by step:
    - The first deconvolution maps high-level latent features to intermediate resolution.
    - The second and third layers continue to upscale, recovering the spatial and temporal granularity.

    - The final deconvolution outputs the reconstructed tensor in its original dimensions: 4 channels × 1000 time steps × 50 spatial grid points. Each transposed convolution is followed by a SELU activation function to ensure non-linearity and stable gradient flow.

2) **CNN "Discriminator"** (Classifier): The discriminator is a binary classifier consisting of two (originally four, but after optimizing the structure, we decided to use two convolutional layers) and two fully connected layers. It distinguishes real samples from generated ones and drives the generator to produce more realistic data.

3) **VAE-GAN**: To integrate the VAE and CNN classifier into a unified generative adversarial network, we treat the decoder of the pretrained VAE as the generator in a GAN framework, and use a CNN classifier as the discriminator. During adversarial training, the decoder generates synthetic samples by decoding either encoded real inputs or random latent vectors sampled from a normal distribution. The discriminator then evaluates both real and generated samples, and is trained to correctly classify them as real or fake.

    Meanwhile, the generator (VAE decoder) is updated to not only minimize reconstruction loss and KL divergence, but also to generate samples that can fool the discriminator. This adversarial setup forms a VAE-GAN: a generative model where the decoder of the VAE is enhanced through GAN-style feedback. This integration is realized in our train_vae_gan() function, where the generator and discriminator are trained alternately. The discriminator is optimized using a BCE loss between its predictions and true labels (real or fake), while the generator is trained using a weighted sum of:
    - Reconstruction loss (MSE): Preserves fidelity to real data.
    - KL divergence (latent regularization): Regularizes the latent space.
    - Adversarial loss (BCE against real labels): Trains the generator to fool the discriminator.

    This hybrid loss enables the model to generate samples that are both structurally sound and visually plausible.

2.4.2 *Loss Functions and Optimizer*

To optimize the performance of our VAE-GAN, we employed Bayesian Optimization to search for the best hyperparameters across a defined search space. The objective function was designed to minimize the reconstruction error on the validation set (20% of the 10k data). For each trial, the optimization process:

1) Samples a candidate set of hyperparameters.
2) Initializes and pretrains a new VAE model on the training set.
3) Trains the full VAE-GAN using adversarial training.
4) Evaluates reconstruction error on the validation set.
5) Returns the negative of the error as the metric to maximize.

The search space included:

- Latent dimension (150–640)
- Learning rates for VAE (0.00025, 0.001) and discriminator (0.000005, 0.00015)
- KL and adversarial loss weights (lambda_kl (0.001, 2.0), lambda_adv (0.001, 1.0))
- Input noise level for training the discriminator (0.1, 0.4)
- Label smoothing to stabilize GAN training (0.1, 0.45)
- Beta for VAE pretraining (0.01, .1)

This process enabled us to efficiently identify hyperparameters that balanced reconstruction quality and adversarial realism. The best configuration found through Bayesian Optimization was then used to train the final model.

We combined three loss components as mentioned in the VAE-GAN architecture: reconstruction loss (MSE), KL divergence loss, and adversarial loss (BCE). For optimization, we used the Adam optimizer with tuned learning rates and hyperparameters from the Bayesian Optimization process.

2.4.3 *Model Training and Evaluation*

We adopted a two-stage training process:

1) **Pretraining the VAE**: The model is first trained using a combination of MSE for reconstruction and a KL divergence term to regularize the latent space. The decoder learns to accurately reproduce the input data from the latent representation without the influence of adversarial feedback. The KL loss is scaled by a tunable coefficient β to control the tradeoff between reconstruction fidelity and latent structure. This stage uses the pretrain_vae() function and typically runs for 10-20 epochs.

2) **Adversarial Training (VAE-GAN)**: After the latent space is stabilized, we introduce the discriminator and jointly train the generator and discriminator in an adversarial setting using the train_vae_gan() function. Each epoch alternates between updating the discriminator and the generator:
    - The discriminator is trained to distinguish real fusion data from fake samples generated by decoding random latent vectors.
    - Noise is added to both real and fake samples to improve robustness, and label smoothing is applied for training stability.
    - The generator (VAE) is then updated to minimize a hybrid loss combining reconstruction (MSE), KL divergence, and adversarial (BCE) loss. This helps the generator produce data that looks more realistic to the discriminator.

After training, the model's performance is evaluated using both visual inspection and quantitative metrics. The reconstruction quality is measured via MSE, and distributional similarity between real and generated samples is assessed using MMD. These metrics provide insights into the fidelity and diversity of the generated fusion data.

3 RESULTS & DISCUSSION

We evaluated the generator both qualitatively and quantitatively. Upon examining **Fig. 4** Generated vs. Real Samples and Channel-wise Differences graph, although the differences or variances don't seem to be very big for the diffusion, convection, and source channels, we observed oscillations or noise in the generated samples across all four channels to different levels. While some of this can be attributed to the intentional noise introduced during training for stability, it may also reflect limitations in the generator's ability to produce high-fidelity outputs. In contrast, the real data appear noticeably smoother. Despite applying Bayesian optimization and extensive tuning to balance the generator and discriminator, there remains room for improvement in the model's generative quality.

Notably, the Source channel revealed a distinct pattern. As the source represents the external fueling profile—plasma injected from the edge to sustain the experiment—it naturally exhibits higher intensity at the edges. Our model struggled to replicate this pattern accurately. The difference plots for this channel showed structured deviations rather than random noise, indicating that the generator failed to fully capture the underlying dynamics of the source channel. This suggests that additional architectural adjustments or domain-informed loss functions might be needed to better model this channel's complexity.

To complement the visual inspection, we assessed the reconstruction quality using MSE between selected generated samples and their corresponding real data. Among the three samples evaluated, one had an MSE as low as 0.2136, indicating a close match between the generated and real sample. Another had an MSE of 0.3204, reflecting a moderate level of noise but generally preserved structure. However, one sample had a higher MSE of 0.6685. Since our data is standardized, MSEs below 0.3 may be considered reasonable, while values above 0.5 may reflect meaningful structural mismatches. These results underscore that while the model has the capacity to produce high-quality outputs, its consistency across samples may be a concern.

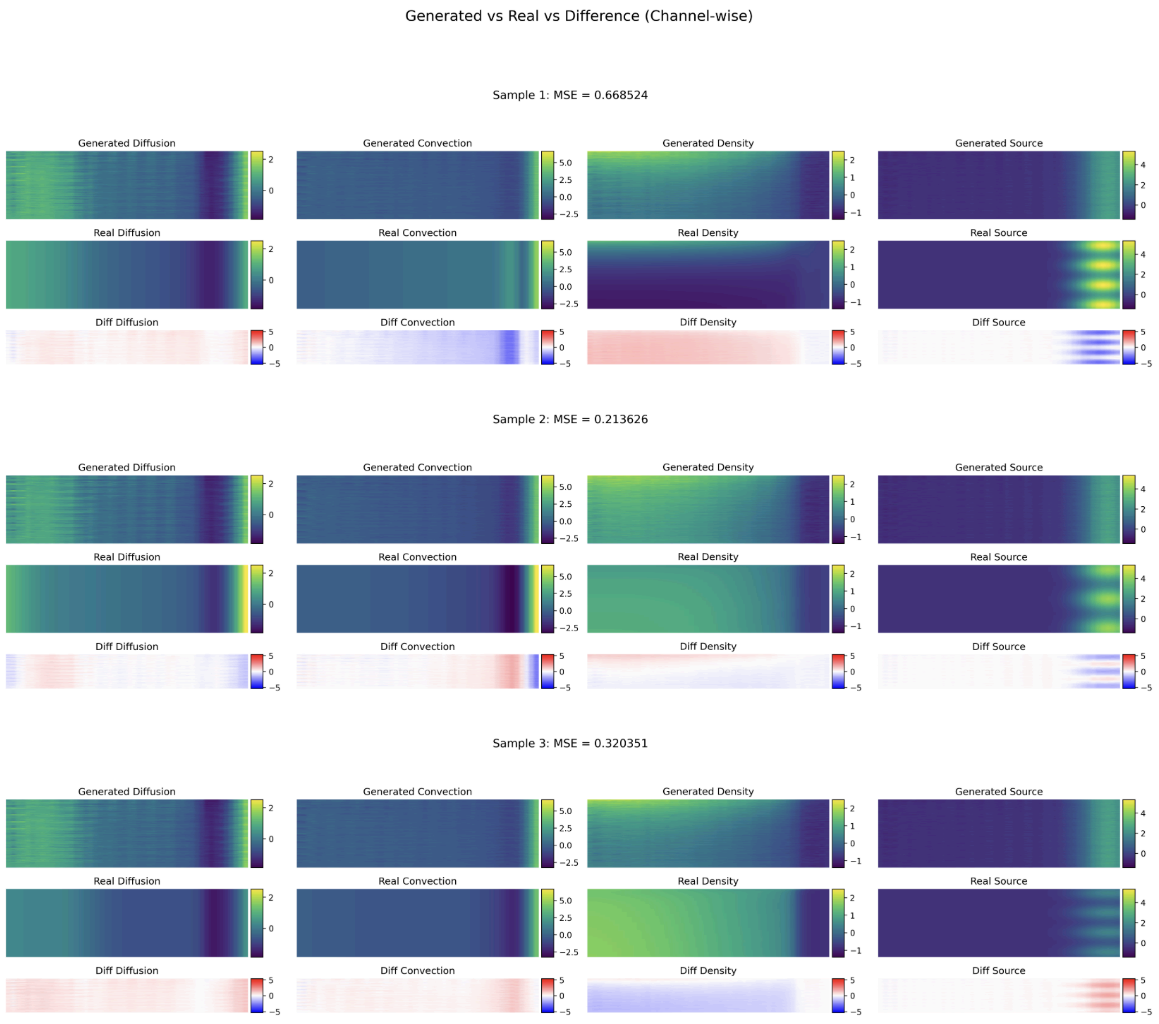

Fig. 4. Generated vs. Real Samples and Channel-wise Differences

Fig. 5 presents the distribution comparison between 10,000 generated and 10,000 real samples across the four physical channels. Visually, the histograms show that the generated data closely mirrors the overall shape, mean (μ), and standard deviation (σ) of the real data across all channels. To statistically assess this alignment, we conducted independent two-sample *t*-tests for each channel. With a significance level α of 0.01, the tests found no significant difference between the real and generated distributions, indicating the generator has learned the central tendencies effectively.

We also computed the MMD, a more sensitive metric that captures higher-order distributional differences beyond mean and variance. Despite the closeness in μ and σ, the MMD values remained relatively high for the four channels. This suggests that while the generator captures the overall range and location of the data, it struggles to replicate the finer structure of the real distributions. This is especially relevant since we calculated the MMD by using the multiscale kernel, which evaluates discrepancies across both local and global patterns. Even small oscillations, high-frequency noise, or distributional asymmetries in the generated data, are penalized by MMD. This is evident in the graphs as the generated data distribution for the diffusion channel shows signs of multimodality, and the density channel has some bigger mismatches in shape, the convection channel has a spikier shape at the center, and the source channel has a skewed tail to the right. This can also be seen and due to the oscillations or noise we saw from the generated samples in **Fig. 4**.

From **Fig. 5**, our findings imply that although the generative model performs well in terms of the mean and standard deviation statistics, subtle differences in distributional shape remain. These results highlight potential areas for refinement, such as applying temporal smoothing to the decoder output or incorporating MMD directly into the training loss to improve distributional fidelity.

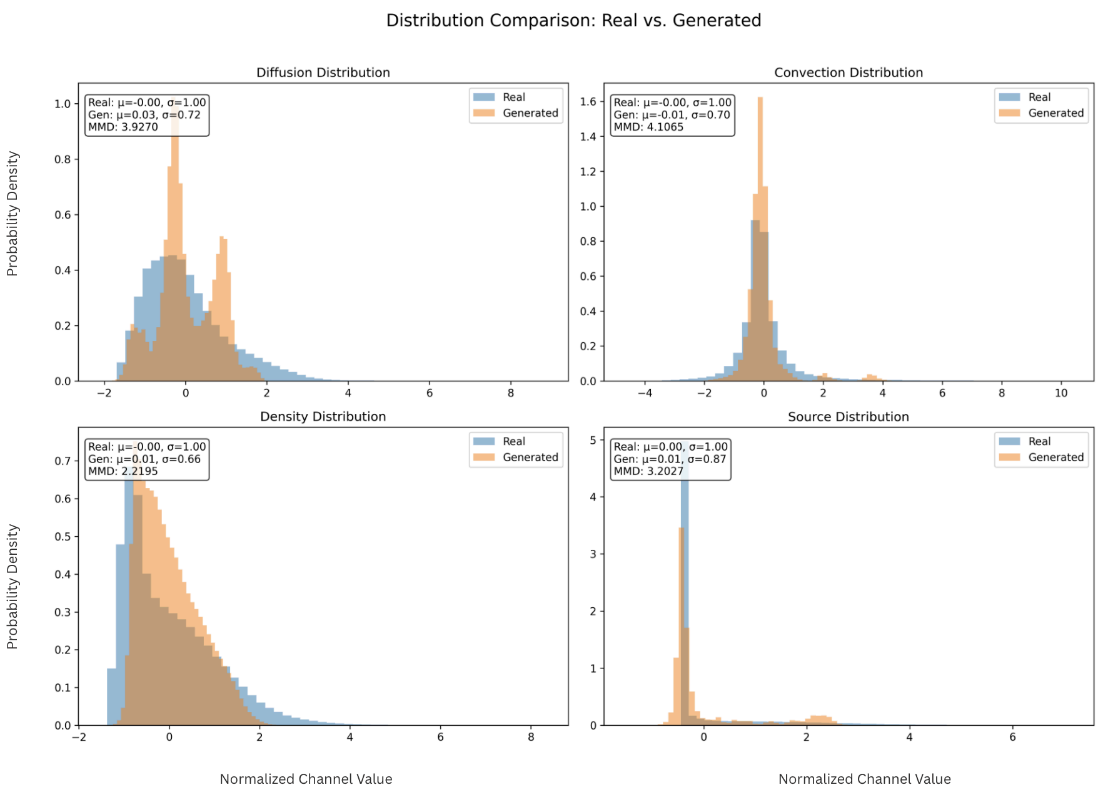

Fig. 5. Distribution Comparison between Generated and Real Data Across Channels

Finally, we plotted the learning curves for our final Bayesian optimized VAE-GAN model.

Discriminator Loss vs. Generator Loss Curves (**Fig 6.1**): As a result of the Bayesian optimization step, the VAE-GAN quickly achieves a constant Nash equilibrium where the loss functions hover around 0.5 and 0.4 respectively. This is a good indication that both are learning and neither is overpowering the other.

Discriminator Confidence on Fake and Real Images (**Fig 6.2**): The dramatic separation between real and fake confidences in the discriminator would normally point towards the generator having a difficult time learning, however from the generated samples, we know this can be questionable. We also know that GAN model losses can be difficult or impossible to interpret, so we should take both the generated samples and the learning curves into consideration.

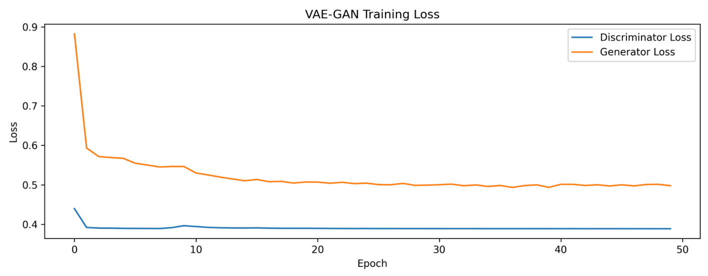

Figure 6.1. Discriminator Loss vs. Generator Loss

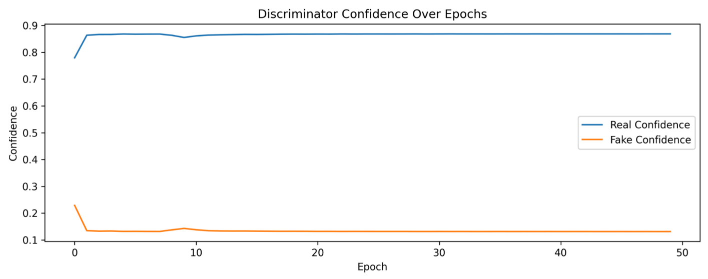

Fig. 6.2. Discriminator Prediction Confidence

4 CONCLUSIONS & FUTURE WORK

In this project, we demonstrated that a VAE-GAN hybrid model can generate relatively high-fidelity fusion simulation data by combining the structured latent representation of a variational autoencoder with the realism enforcement of adversarial training. Our model performed well in capturing overall spatial and temporal patterns, producing samples that closely align with the real data in terms of mean and standard deviation. However, distribution-level metrics such as MMD revealed discrepancies in finer structural details, highlighting areas where the generator can still be improved.

Looking forward, we see several opportunities to enhance our model. Future work could incorporate alternative loss formulations such as WGAN objectives for more stable training, or leverage conditional architectures that allow physical parameters (like fueling rates) to guide the generation process. Additionally, attention mechanisms may improve the model's ability to capture long-range dependencies in time and space. We also see potential in integrating physics-informed constraints, such as through Physics Informed Neural Networks (PINNs), to ensure that the generated data adheres more closely to the underlying scientific laws (which may help to generate better "informed" and higher fidelity plasma data). Finally, since we saw in **Fig. 4** that some channels like source have greater variances between the generated and real data, we examined a little bit of the channel-specific loss weighting, which did show us some interesting findings. But, we haven't achieved a robust conclusion and we hope to explore further. Hopefully, these improvements would potentially make our model even more valuable as a low-cost, scalable tool for accelerating fusion research.

5 REFERENCES

Shambhavi. (n.d.). *An Introduction to VAE-GANs*. Weights & Biases. Retrieved April 21, 2025, from https://wandb.ai/shambhavicodes/vae-gan/reports/An-Introduction-to-VAE-GANs--VmlldzoxMTcxMjM5

Escuccim. (n.d.). *vaegan-pytorc [GitHub repository]*. GitHub. Retrieved April 21, 2025, from https://github.com/escuccim/vaegan-pytorc

Tunali, O. (2019, March 8). *Maximum mean discrepancy in machine learning*. Retrieved April 28, from https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

Suresh, K., & Fanelli, C. (2025). *VAE-GAN MMD Interpretation – Distribution Comparison* [Google Slides presentation]. Retrieved from https://docs.google.com/presentation/d/1rs-nv9id1OAwHznWBWoS5jZqsCGBiYg-2jOYM5nzBSw