Amplitude extraction with generative AI

Glòria Montaña

Jefferson Lab

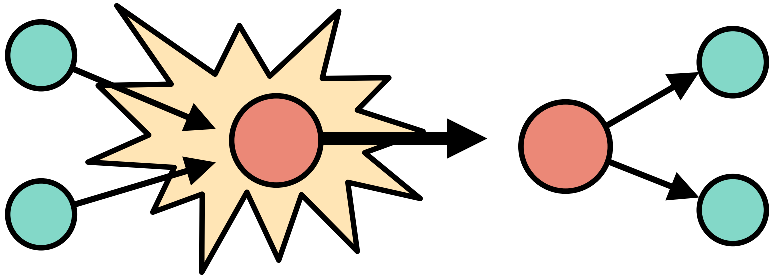
In collaboration with M. Battaglieri, A. Pilloni, Y. Li and others

“AI for Hadron Spectroscopy at JLab” workshop
Jefferson Lab, 4-5 June 2025

A(i)DAPT
AI for Data Analysis and PreservaTion

Exotic Hadrons Topical Collaboration

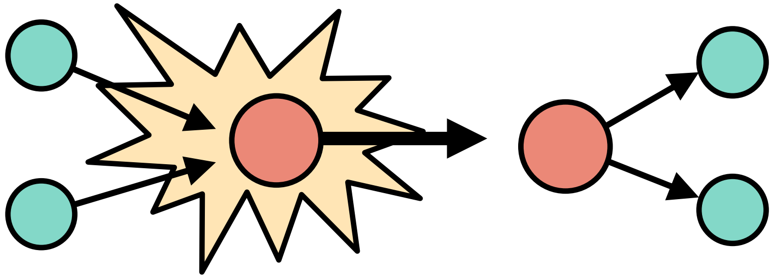
PHYSICS MOTIVATION



The **scattering amplitude** $\mathcal{A}$ is the fundamental quantity of interest in hadron reactions:

- Encodes the underlying dynamics of the interaction
- Crucial for understanding resonance production, decays…
- Is a complex quantity: magnitude + phase

PHYSICS MOTIVATION



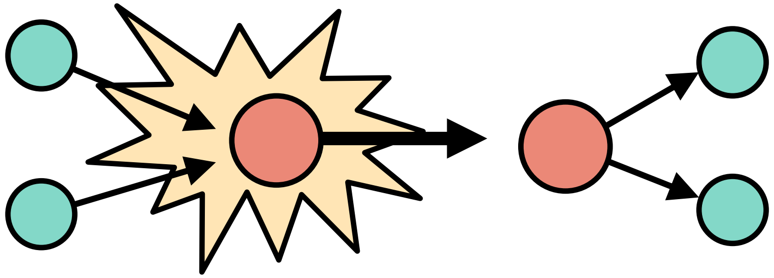
The **scattering amplitude** $\mathcal{A}$ is the fundamental quantity of interest in hadron reactions:

- Encodes the underlying dynamics of the interaction
- Crucial for understanding resonance production, decays…
- Is a complex quantity: magnitude + phase

The **cross section** σ is an experimentally observable quantity:

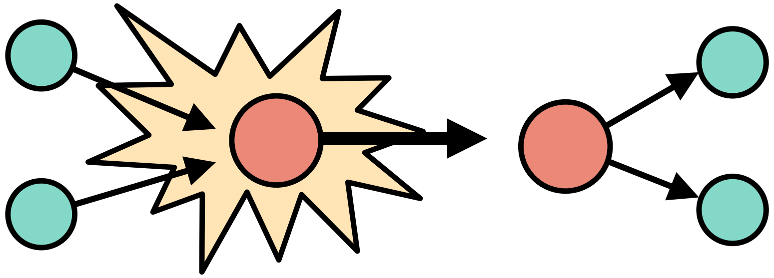
- Related to $|\mathcal{A}|^2$
- The information about the phase is lost

PHYSICS MOTIVATION



The reconstruction of the amplitude from the differential cross section is hard, even for the simplest elastic $2 \rightarrow 2$ scattering.

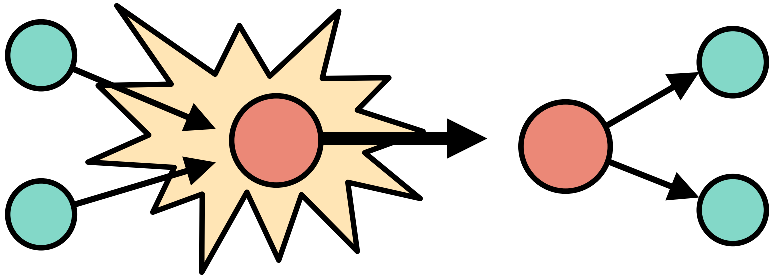
PHYSICS MOTIVATION



The reconstruction of the amplitude from the differential cross section is a hard inverse problem, even for the simplest elastic $2 \rightarrow 2$ scattering.

Can we use modern machine-learning techniques to recover the scattering amplitude from experimental data of cross sections?

PHYSICS MOTIVATION



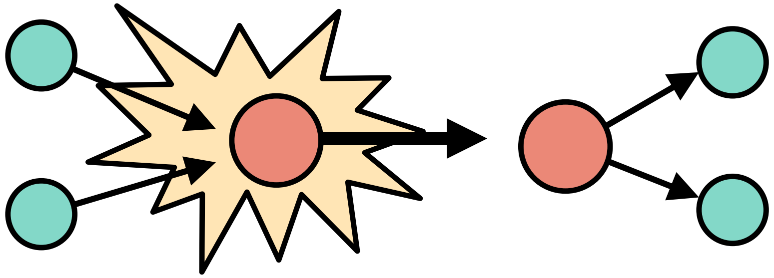
The reconstruction of the amplitude from the differential cross section is a hard inverse problem, even for the simplest elastic $2 \to 2$ scattering.

Can we use modern machine-learning techniques to recover the scattering amplitude from experimental data of cross sections?

Why physics-constrained **GANs**?

- Learn distributions and patterns of the (pseudo) data
- Incorporate physics constraints

PHYSICS MOTIVATION



The reconstruction of the amplitude from the differential cross section is a hard inverse problem, even for the simplest elastic 2 → 2 scattering.

Can we use modern machine-learning techniques to recover the scattering amplitude from experimental data of cross sections?

Why physics-constrained **GANs**?

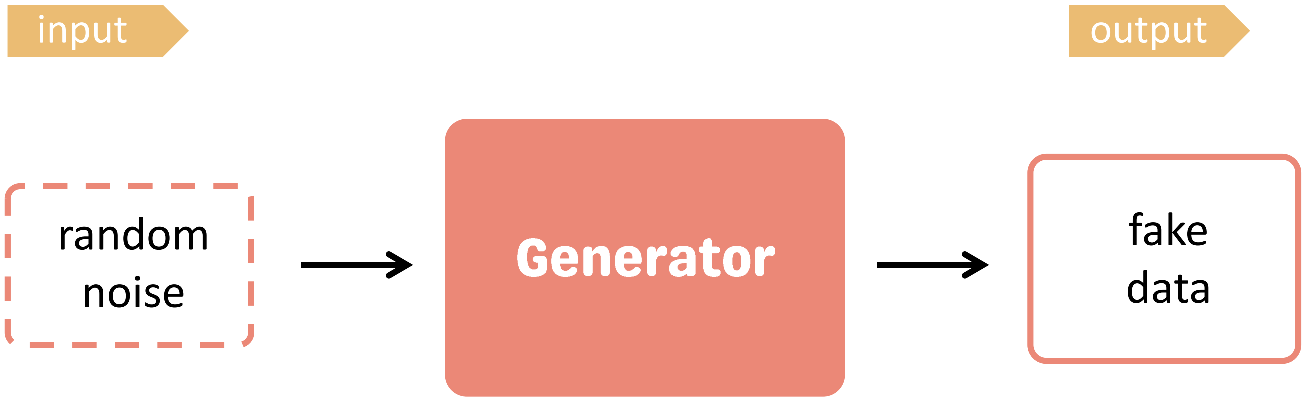
- Learn distributions and patterns of the (pseudo) data
- Incorporate physics constraints

The modulus and phase of the scattering amplitude are related by the **unitarity relation**.

GANs (Generative Adversarial Networks) IN A NUTSHELL

Two neural networks:

- The **generator** needs to capture the data distribution

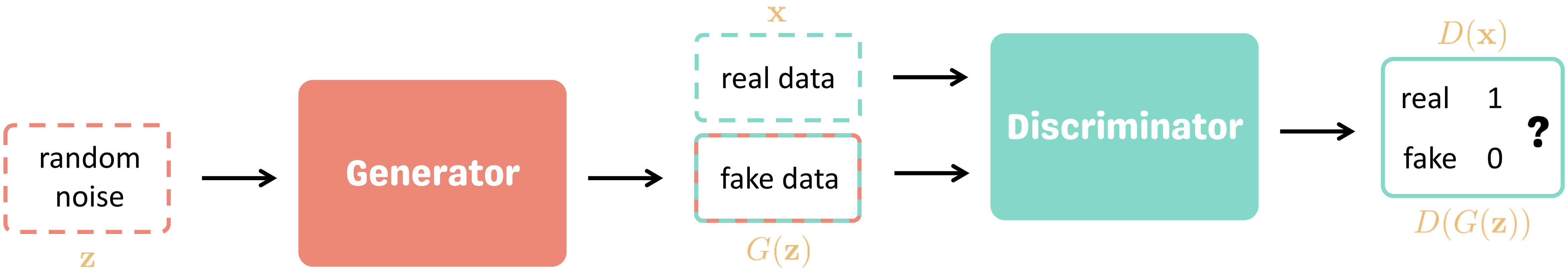


GANs (Generative Adversarial Networks) IN A NUTSHELL

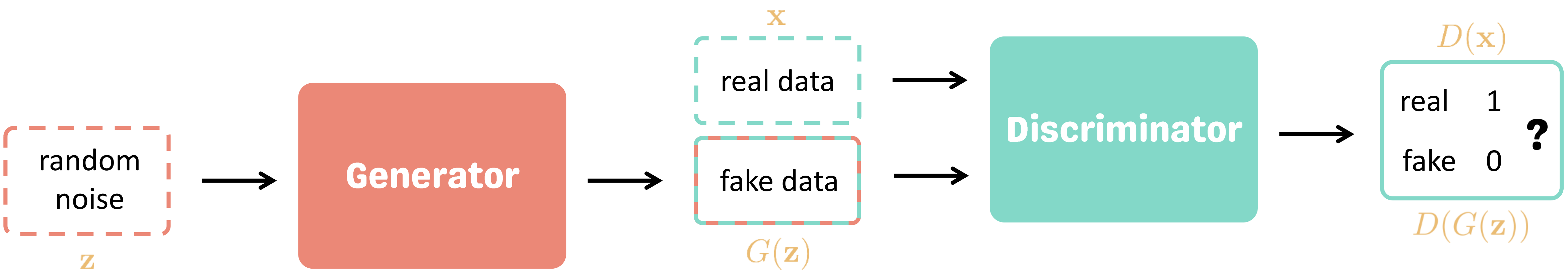
Two neural networks:

- The **generator** needs to capture the data distribution
- The **discriminator** estimates the probability that a sample comes from the training data rather than from the generator

GANs (Generative Adversarial Networks) IN A NUTSHELL



GANs (Generative Adversarial Networks) IN A NUTSHELL

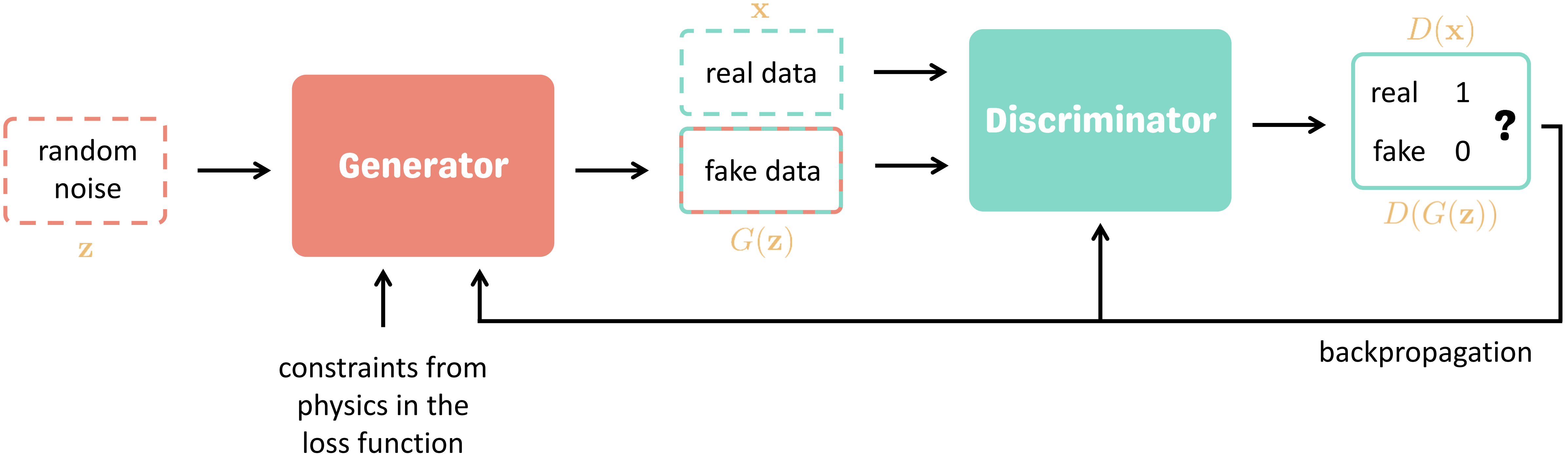


GAN Minmax Game

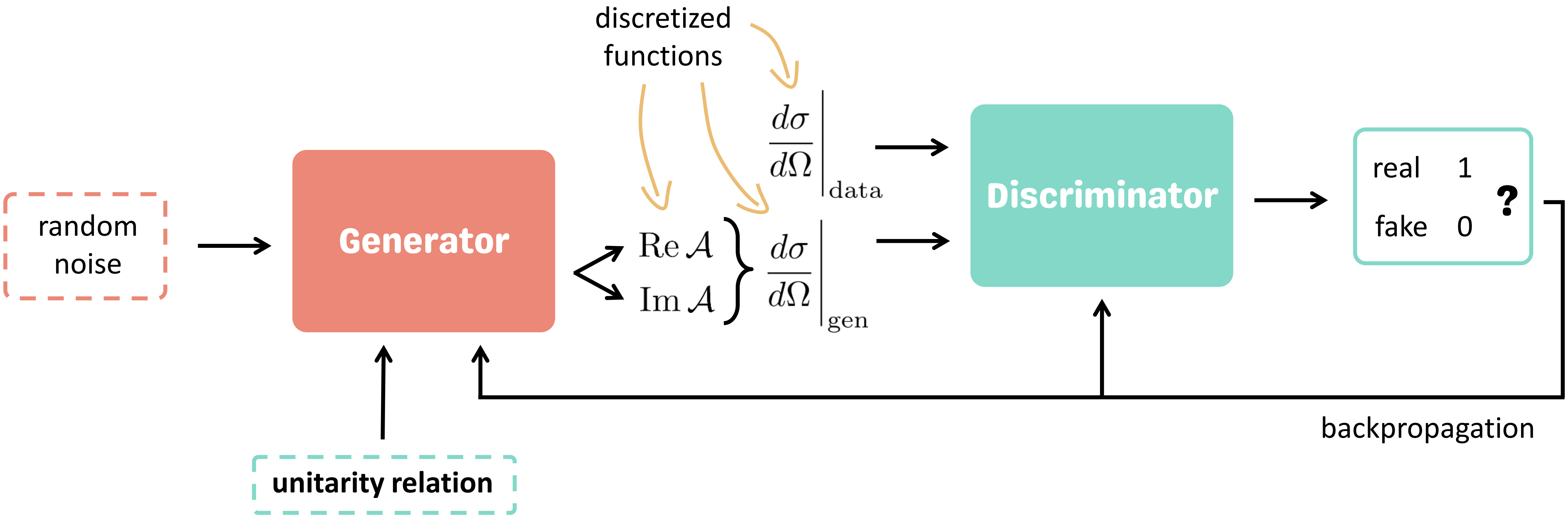
$$\min_G \max_D V(D,G) = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})]}_{\text{real data}} + \underbrace{\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]}_{\text{fake data}}$$

	real data	fake data
G minimizes ↓	N/A	*D* classifies as 0
D maximizes ↑	*D* classifies as 1	*D* classifies as 0

INTRODUCING PHYSICS CONSTRAINTS



INTRODUCING PHYSICS CONSTRAINTS



PION-PION ELASTIC SCATTERING

Elastic scattering $\pi^+\pi^- \to \pi^+\pi^-$

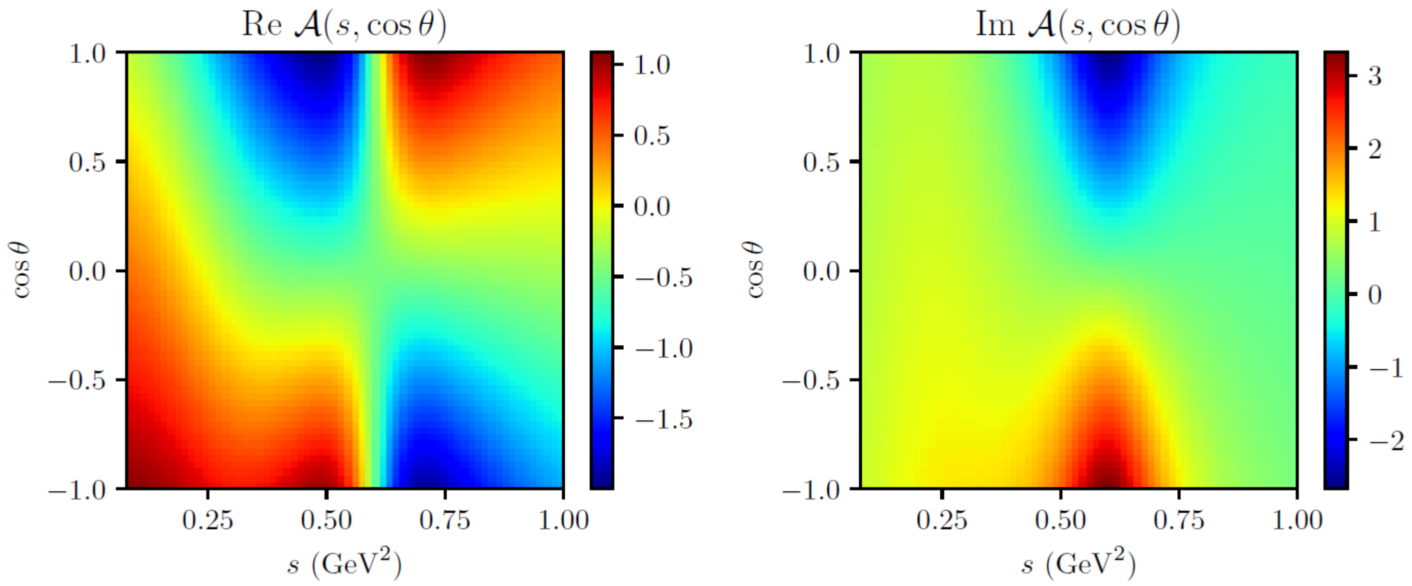
- dominated by $f_0(500)$ and $\rho(770)$ resonances

$$\mathcal{A}(s, \cos\theta) = \sum_{\ell=0}^{n} (2\ell+1) f_\ell(s) P_\ell(\cos\theta)$$

Partial-wave decomposition of the amplitude truncated to $n = 1$ and Breit-Wigner type partial waves:

$$\mathcal{A}(s, \cos\theta) = f_0(s) + 3 f_1(s) \cos\theta$$

$$f_\ell = \frac{m_\ell \Gamma_\ell}{m_\ell^2 - s - i m_\ell \Gamma_\ell}$$



PION-PION ELASTIC SCATTERING

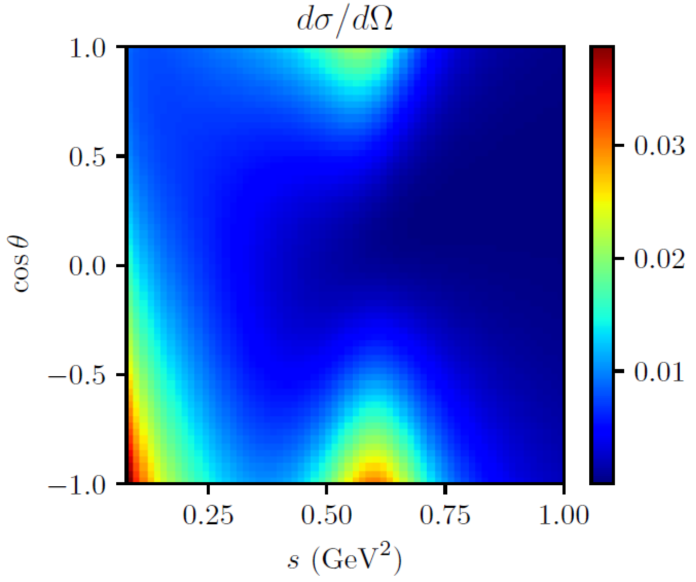
Elastic scattering $\pi^+\pi^- \to \pi^+\pi^-$

➜ dominated by $f_0(500)$ and $\rho(770)$ resonances

$$\mathcal{A}(s, \cos\theta) = \sum_{\ell=0}^{n} (2\ell+1) f_\ell(s) P_\ell(\cos\theta)$$

Differential cross section

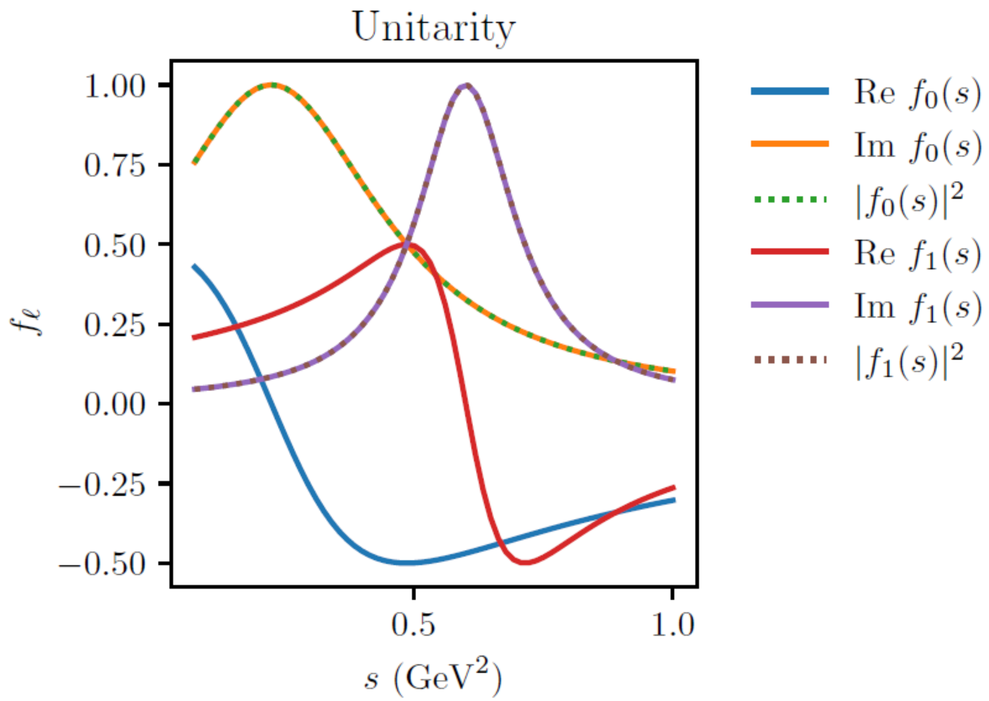
$$\frac{d\sigma}{d\Omega} = \frac{1}{64\pi^2} \frac{1}{s} |\mathcal{A}(s, \cos\theta)|^2$$



CONSTRAINTS: UNITARITY

- Unitarity of the partial waves

$$\text{Im}\ f_\ell(s) = |f_\ell(s)|^2$$

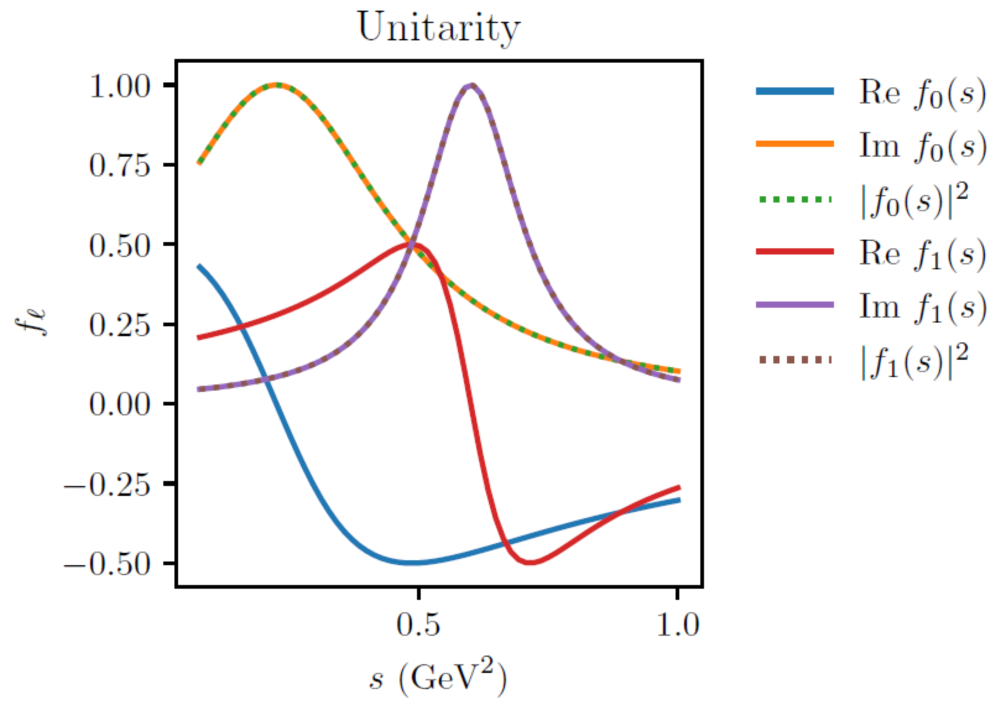


$$f_\ell(s) = \frac{1}{2}\int_{-1}^{+1} dz P_\ell(z)\mathcal{A}(s,z)$$

CONSTRAINTS: UNITARITY

- Unitarity of the partial waves

$$\text{Im } f_\ell(s) = |f_\ell(s)|^2$$



$$f_\ell(s) = \frac{1}{2}\int_{-1}^{+1} dz P_\ell(z)\mathcal{A}(s,z)$$

- Integral unitarity relation for the full amplitude

$$\text{Im } \mathcal{A}(s,z) = \frac{1}{4\pi}\int_0^{2\pi} d\phi \int_{-1}^{+1} dz' \mathcal{A}(s,z')\mathcal{A}^*(s,z'')$$

$$z'' = zz' + \sqrt{1-z^2}\sqrt{1-z'^{\,2}}\cos\phi$$

or, equivalently

$$\sin\Phi(s,z) = \int_0^{2\pi} d\phi \int_{-1}^{+1} dz' \frac{|\mathcal{A}(s,z')||\mathcal{A}(s,z'')|}{4\pi|\mathcal{A}(s,z)|} \times \cos\left[\Phi(s,z') - \Phi(s,z'')\right]$$

Phase ambiguity: $\mathcal{A}(s,z) \to -\mathcal{A}^*(s,z)$

$\Phi(s,z) \to \pi - \Phi(s,z)$

IMPLEMENTATION OF CONSTRAINTS

- GAN Loss Function:

MSE Loss Measure the mean squared error between the target and output.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\text{output}_i - \text{target}_i)^2$$

IMPLEMENTATION OF CONSTRAINTS

GAN Loss Function:

MSE Loss Measure the mean squared error between the target and output.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{output}_i - \mathrm{target}_i)^2$$

Physics Constraints Loss Functions:

Unitarity Loss Enforce unitarity by comparing the modulus squared of the integral of the scattering amplitudes over angular variables to the imaginary part.

$$\mathcal{L}_{\mathrm{u}} = \frac{1}{N \cdot N_s \cdot N_z}\sum_{i=1}^{N}\sum_{j=1}^{N_s}\sum_{k=1}^{N_z}\Big(\big|\mathrm{Im}\,\mathcal{A}(s,z) - \mathrm{Re}\,\mathcal{I}(s,z)\big| + \big|\mathrm{Im}\,\mathcal{I}(s,z)\big|\Big)$$

with $\mathcal{I}(s,z) = \frac{1}{4\pi}\int_{-1}^{1} dz' \int_{0}^{2\pi} d\phi\,(\mathcal{A}(s,z')\mathcal{A}(s,z''(z,z',\phi)))$

Integral approximator: Simpson’s rule

Integral sampling points: $[\cos\theta \times \phi] = 64 \times 10$

IMPLEMENTATION OF CONSTRAINTS

GAN Loss Function:

MSE Loss Measure the mean squared error between the target and output.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\mathrm{output}_i - \mathrm{target}_i)^2$$

Physics Constraints Loss Functions:

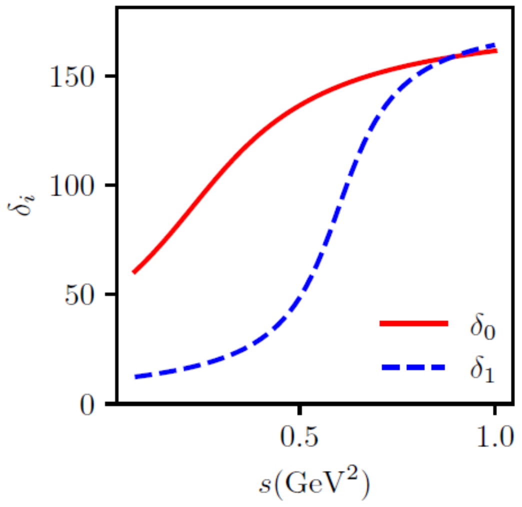
Unitarity Loss Enforce unitarity by comparing the modulus squared of the integral of the scattering amplitudes over angular variables to the imaginary part.

d0 Loss Ensure the positive derivative of the f_0 phase shift.

d1 Loss Ensure the positive derivative of the f_1 phase shift.

$$\mathcal{L}_{D_\ell} = \frac{1}{N}\sum_{i=1}^{N}\log\left(\max\left(0, -\Delta\delta_\ell(s)\right) + 1\right)$$

$$\delta_\ell = \mathrm{atan}\left(\frac{\mathrm{Im}\, f_\ell(s)}{\mathrm{Re}\, f_\ell(s)}\right), \qquad f_\ell(s) = \frac{1}{2}\int_{-1}^{+1} dz P_\ell(z)\mathcal{A}(s,z)$$



IMPLEMENTATION OF CONSTRAINTS

GAN Loss Function:

MSE Loss Measure the mean squared error between the target and output.

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\text{output}_i - \text{target}_i)^2$$

Physics Constraints Loss Functions:

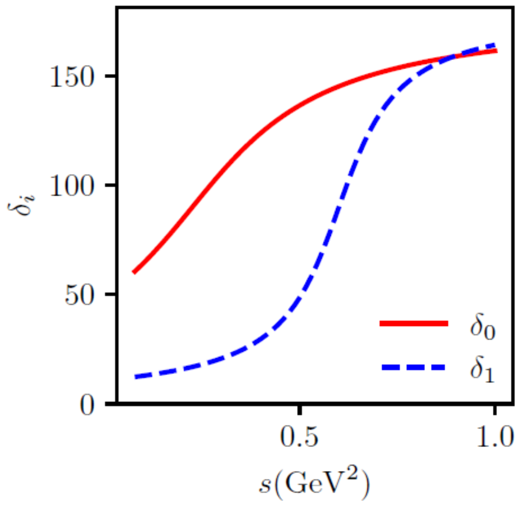
Unitarity Loss Enforce unitarity by comparing the modulus squared of the integral of the scattering amplitudes over angular variables to the imaginary part.

d0 Loss Ensure the positive derivative of the f_0 phase shift.

d1 Loss Ensure the positive derivative of the f_1 phase shift.

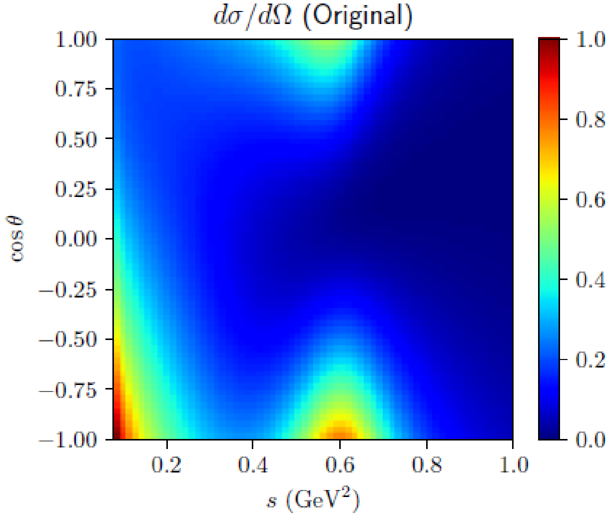
$$\mathcal{L}_{D_\ell} = \frac{1}{N}\sum_{i=1}^{N}\log\left(\max\left(0, -\Delta\delta_\ell(s)\right) + 1\right)$$

$$\delta_\ell = \text{atan}\left(\frac{\text{Im } f_\ell(s)}{\text{Re } f_\ell(s)}\right), \qquad f_\ell(s) = \frac{1}{2}\int_{-1}^{+1} dz P_\ell(z)\mathcal{A}(s,z)$$



Better way to constrain the phase?

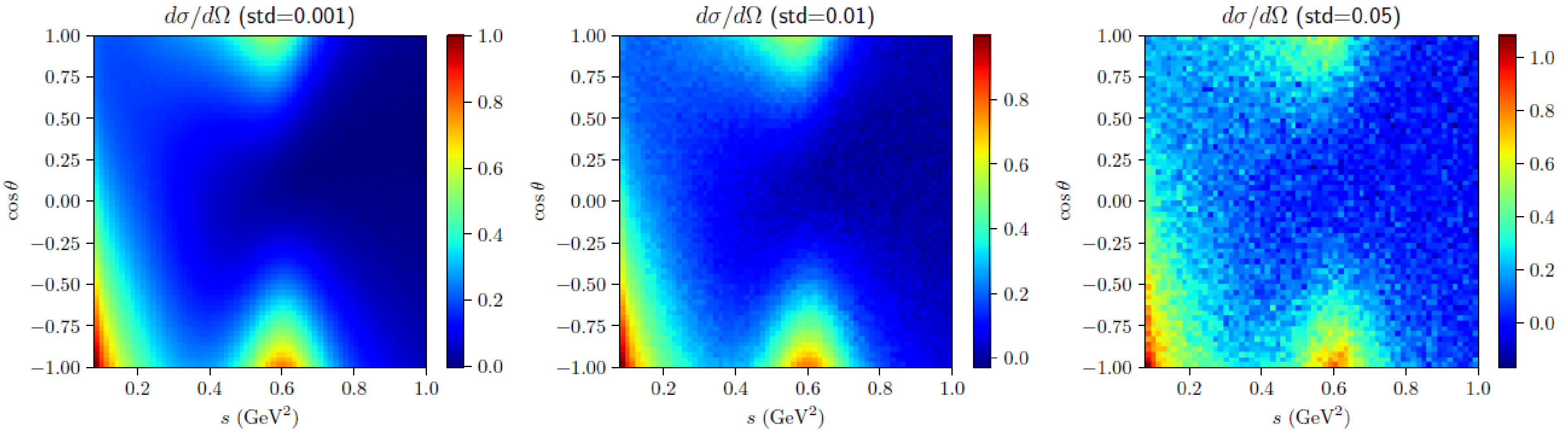
TRAINING DATASET



Normalized differential cross section discretized in grid:

$$64 \times 64, \;\; s \in [(2m_\pi)^2, 1 \text{ GeV}^2], \;\; \cos\theta \in [-1, 1]$$

Training samples with additional gaussian noise:



PRELIMINARY RESULTS

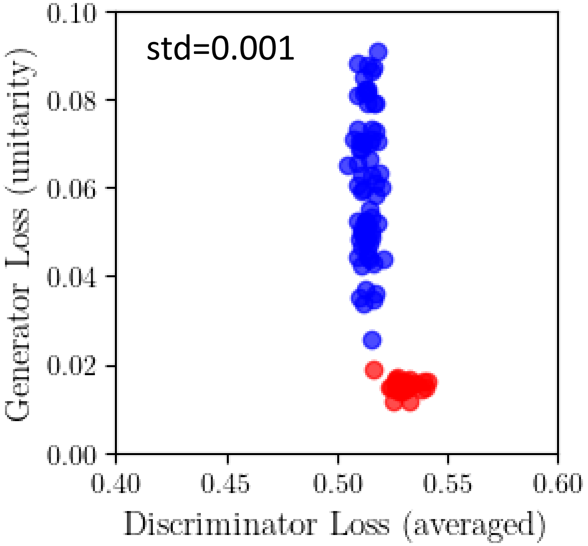
- Trained 100 GANs for 200 epochs:

Stop training if unitarity loss is smaller than 0.02 and changes less than 0.01 and for 10 consecutive epochs:

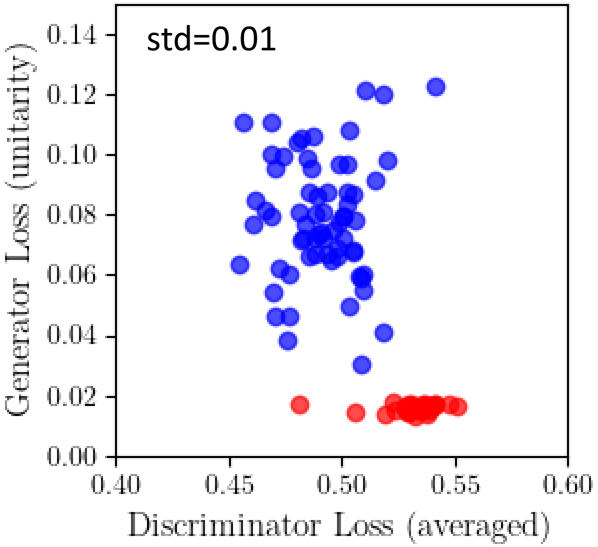
$$\mathcal{L}_{\mathrm{u}} < 0.02$$

$$\mathcal{L}_{\mathrm{u},n} - \mathcal{L}_{\mathrm{u},n-1} < 0.01$$

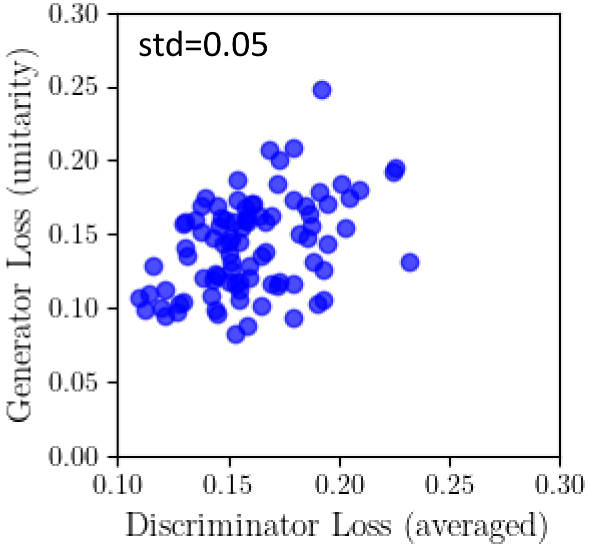
29/100 converged



33/100 converged

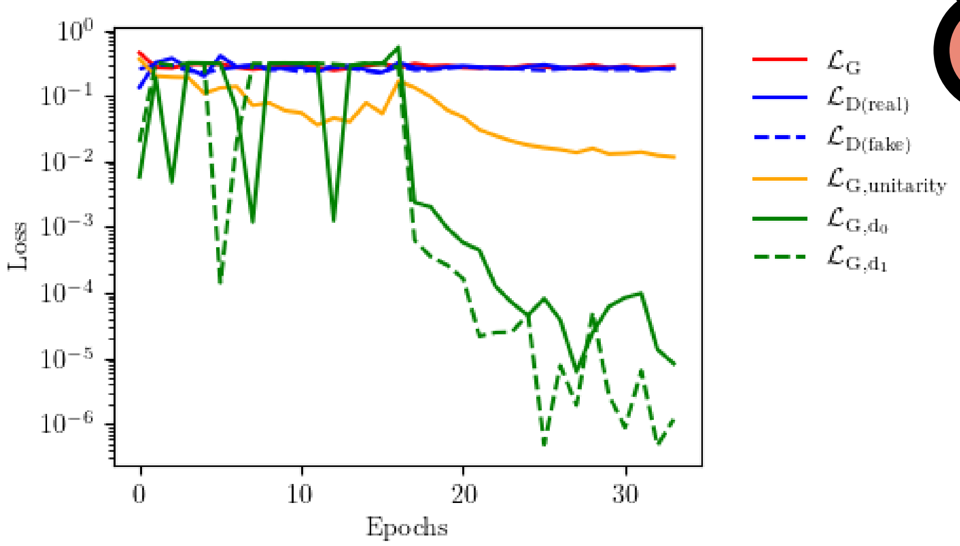


0/100 converged

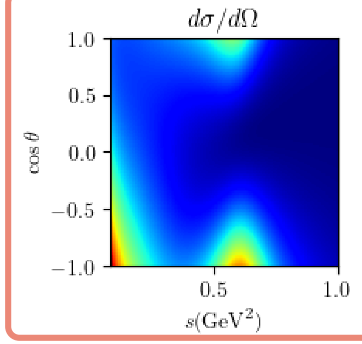


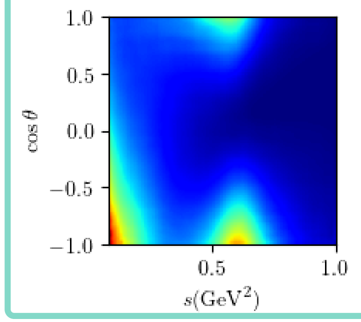
PRELIMINARY RESULTS

- Example of converged GAN with std=0.01:



model ("true" without noise)

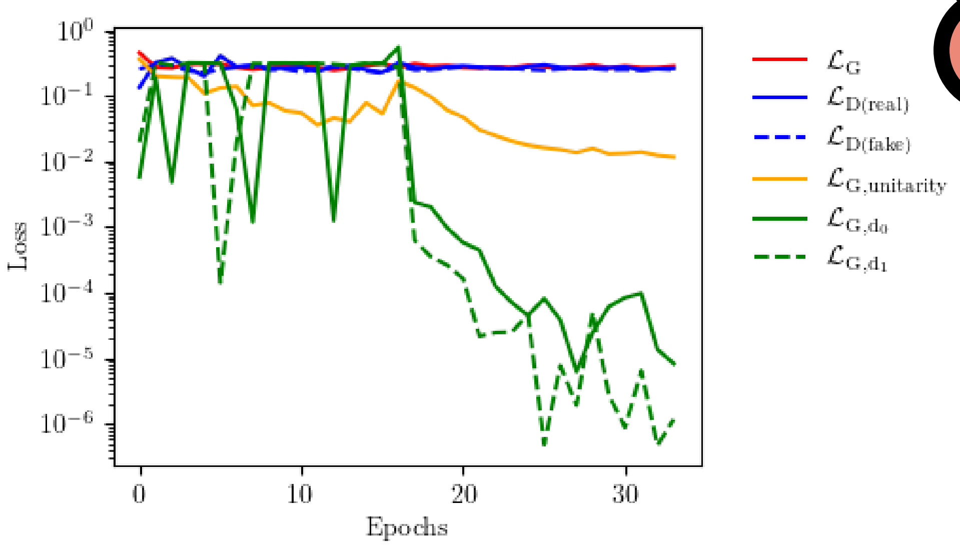




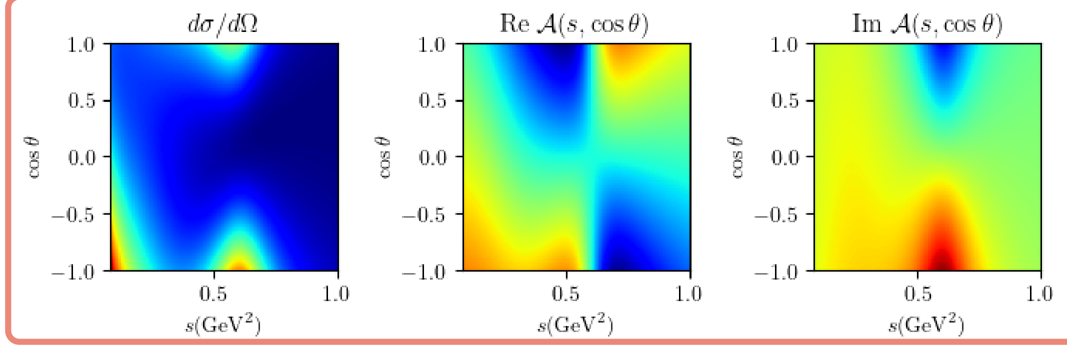
generated ("fake")

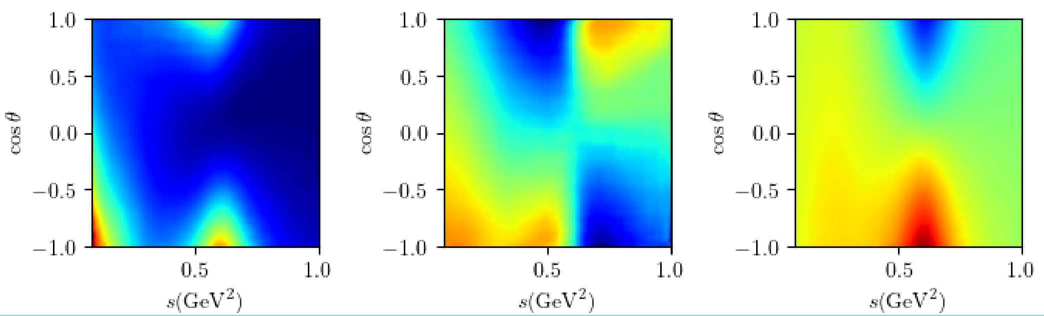
PRELIMINARY RESULTS

- Example of converged GAN with std=0.01:



model ("true" without noise)

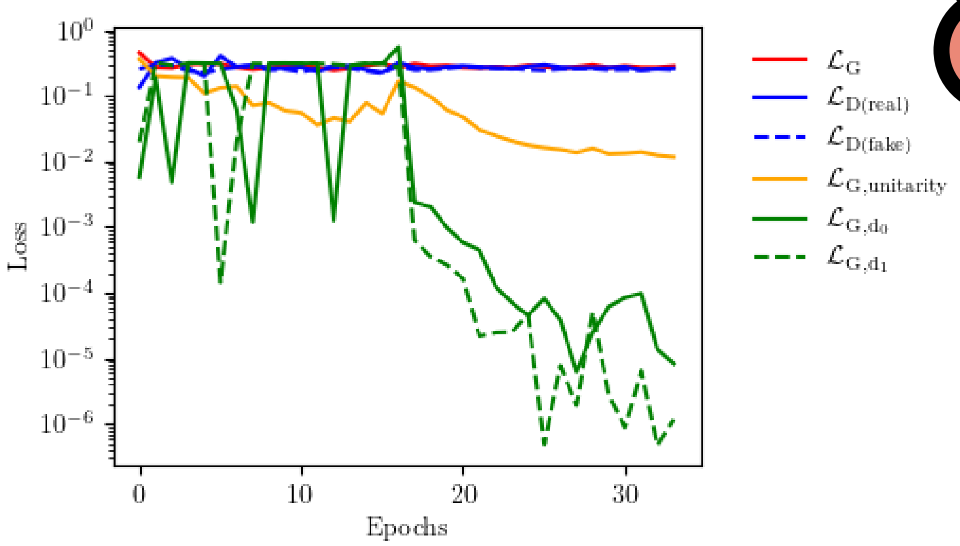




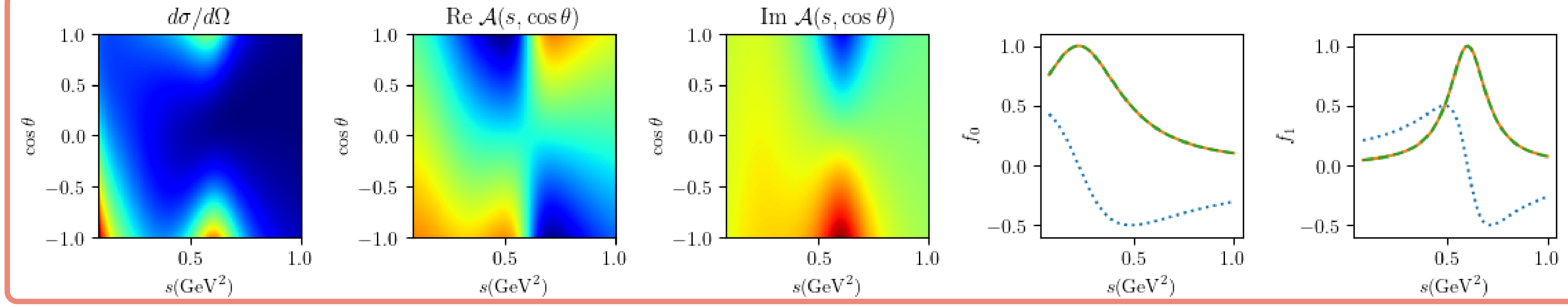
generated ("fake")

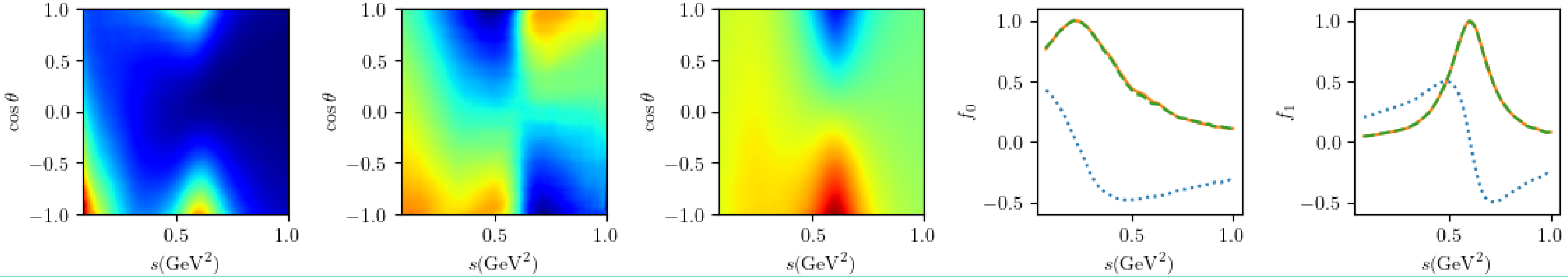
PRELIMINARY RESULTS

- Example of converged GAN with std=0.01:



model ("true" without noise)

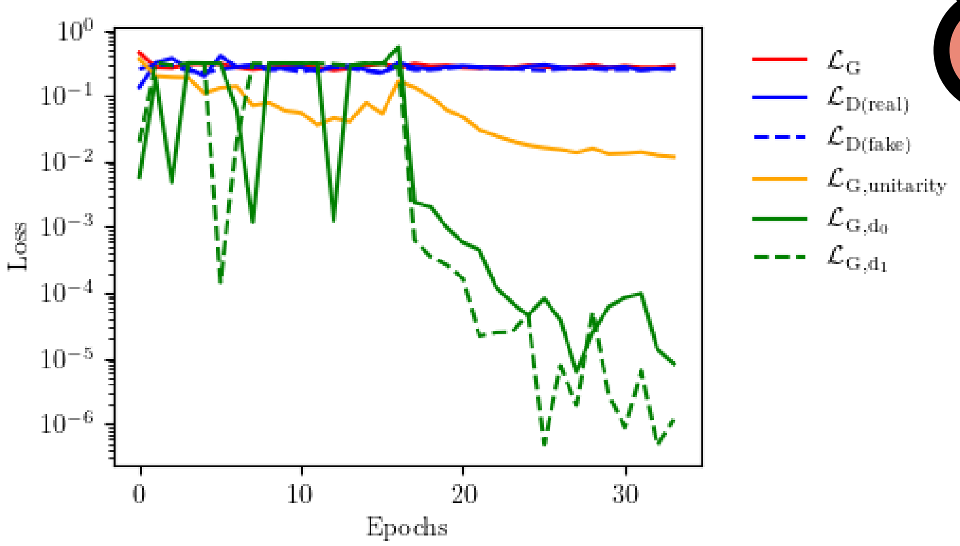




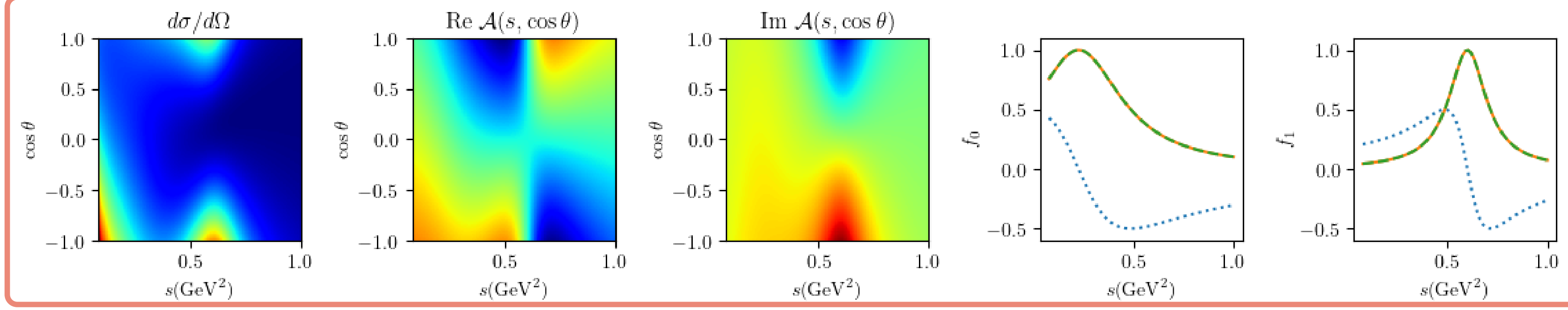
generated ("fake")

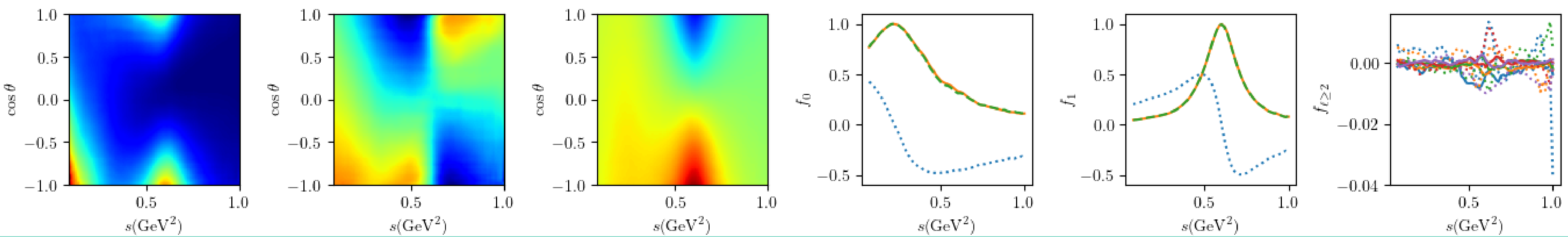
PRELIMINARY RESULTS

- Example of converged GAN with std=0.01:



model ("true" without noise)

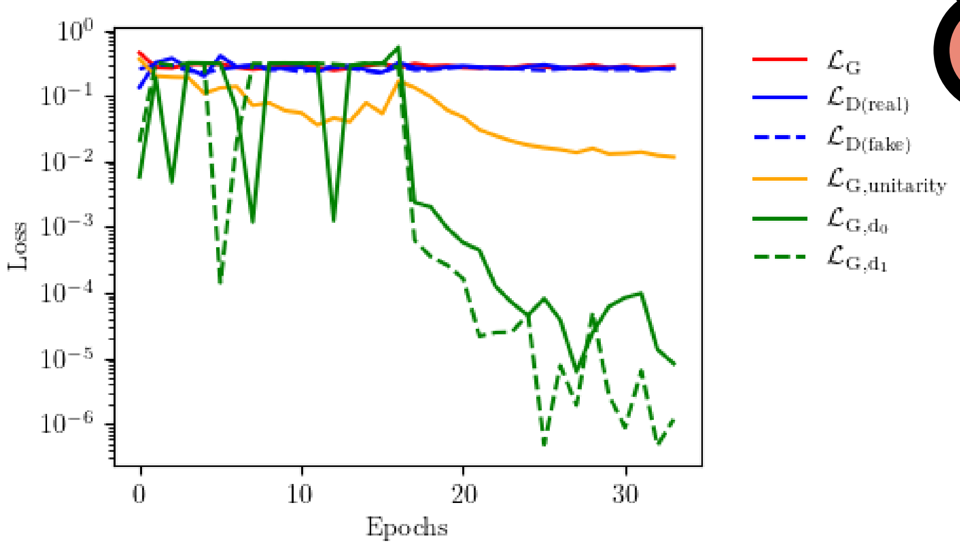




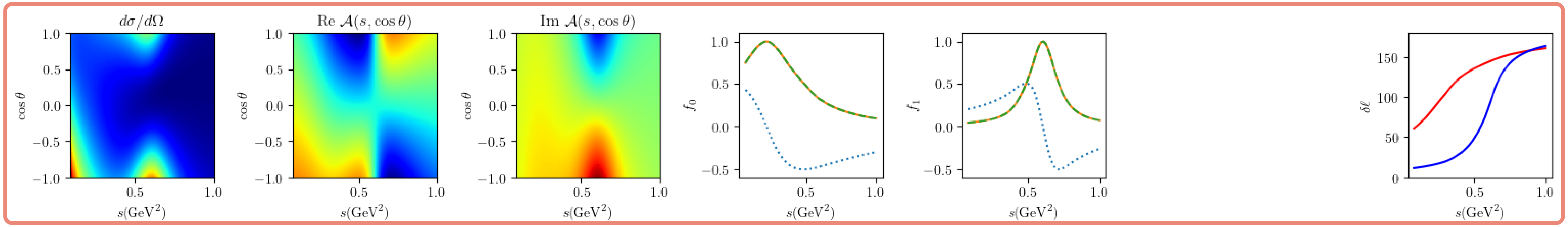
generated ("fake")

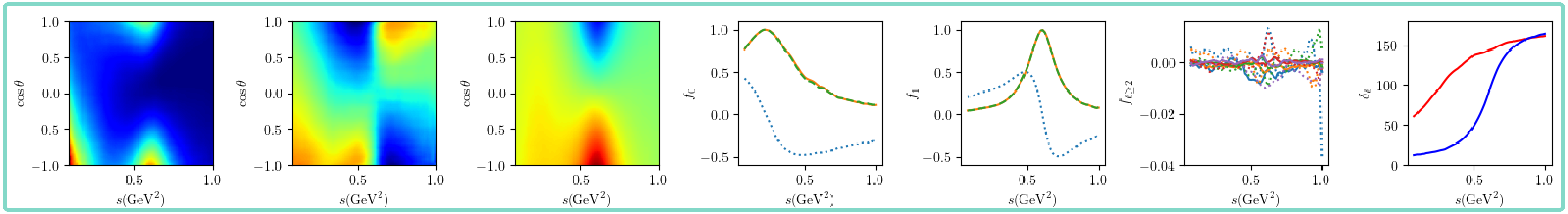
PRELIMINARY RESULTS

- Example of converged GAN with std=0.01:



model ("true" without noise)

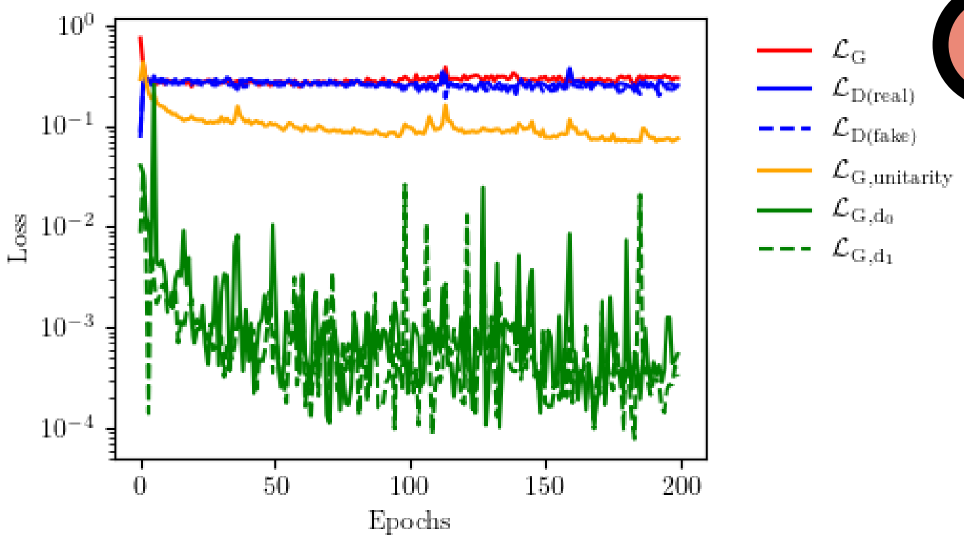




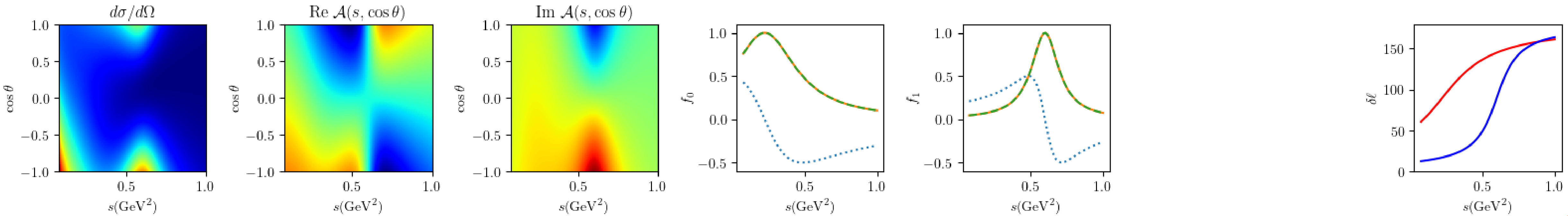
generated ("fake")

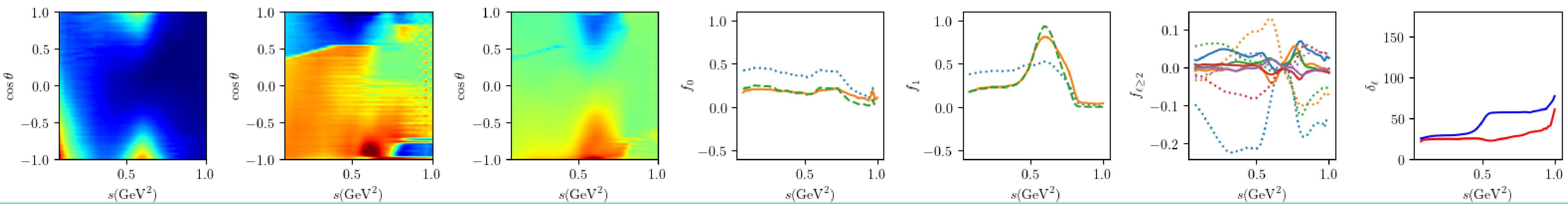
PRELIMINARY RESULTS

- Example of non-converged GAN with std=0.01:



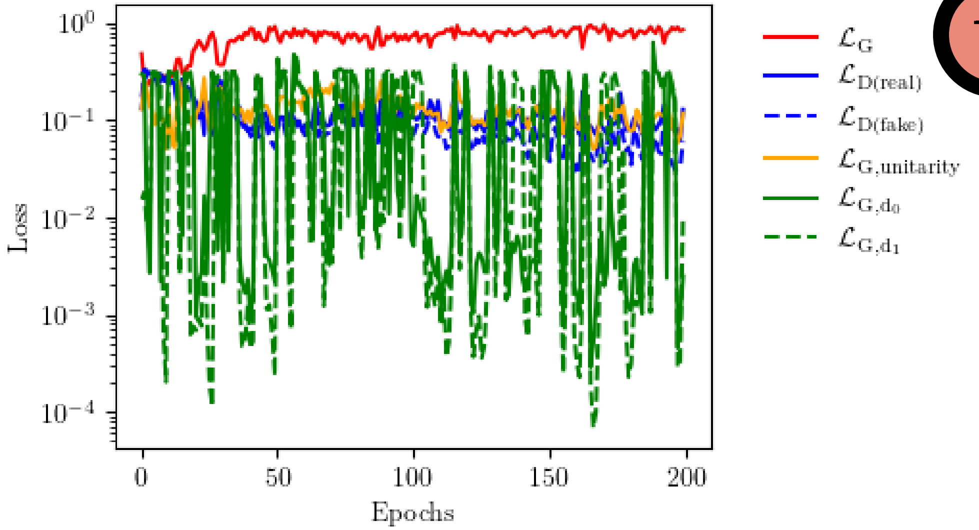
model ("true" without noise)





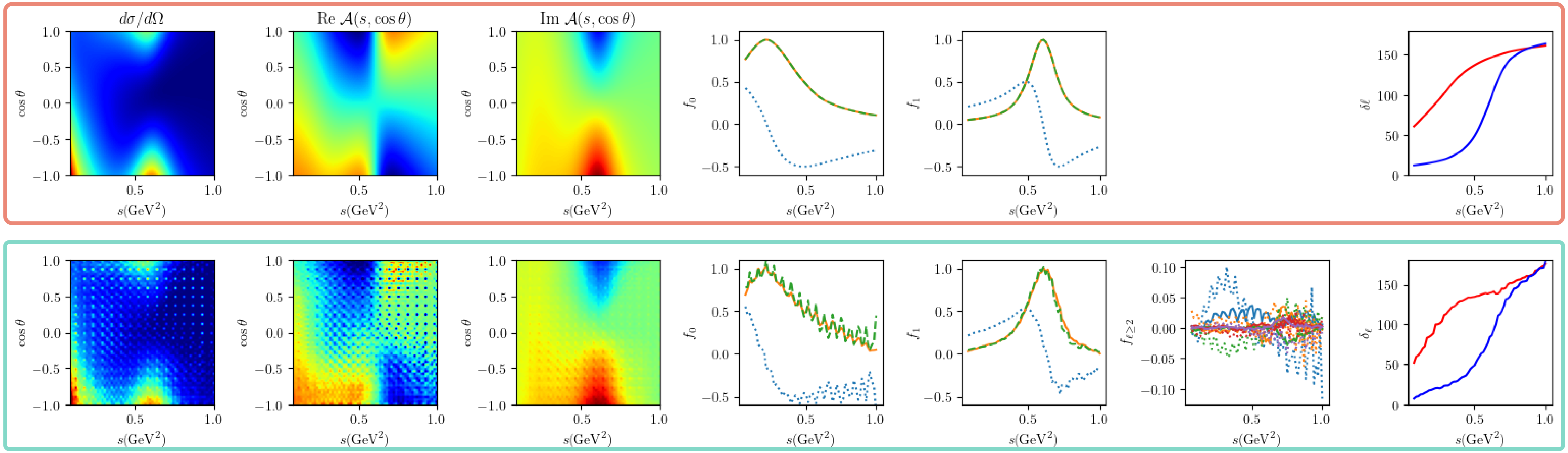
generated ("fake")

PRELIMINARY RESULTS



- Example of a "not too bad" non-converged GAN with std=0.05:

model ("true" without noise)



generated ("fake")

CONCLUSIONS & OUTLOOK

Current achievements:

We developed a physics-constrained GAN framework for the direct extraction of complex amplitudes from simulated cross-section data.

Integrated a unitarity loss and explicit phase constraints to guide the GAN training, ensuring the physical validity and uniqueness of the recovered amplitudes.

What's next?

Optimize the GAN architecture and fine-tune hyperparameters to maximize performance and training stability.

Explore additional/alternative physics-informed constraints to stabilize the GAN training further.

Perform a quantitative analysis and error estimation.

CONCLUSIONS & OUTLOOK

Longer-term future directions:

Explore diffusion models as an alternative generative model for amplitude extraction.

Extend from cross-section to the event-level analysis using, e.g. normalizing flows.

Generalize to more complex processes and data.

Preliminary status, but the results of using physics-constrained generative models to extract amplitudes from cross sections employing are promising.

Back-up slides

IMPLEMENTATION DETAILS

python
PyTorch

GAN architecture:

Generator

Layer Type	Input Dimensions	Output Dimensions	Activation/Other Details
Fully Connected	(batch_size, noise_dim)	(batch_size, $4 \times 4 \times 1024$)	BatchNorm, LeakyReLU (0.2)
Reshape	(batch_size, $4 \times 4 \times 1024$)	(batch_size, 4, 64, 64)	Reshapes tensor
ConvTranspose2d (1)	(batch_size, 4, 64, 64)	(batch_size, 64, 64, 64)	BatchNorm, LeakyReLU (0.2), Kernel=17
ConvTranspose2d (2)	(batch_size, 64, 64, 64)	(batch_size, 64, 64, 64)	BatchNorm, LeakyReLU (0.2), Kernel=17
ConvTranspose2d (3)	(batch_size, 64, 64, 64)	(batch_size, 64, 64, 64)	BatchNorm, LeakyReLU (0.2), Kernel=17
ConvTranspose2d (4)	(batch_size, 64, 64, 64)	(batch_size, 64, 64, 64)	BatchNorm, LeakyReLU (0.2), Kernel=17
Conv2d (Final)	(batch_size, 64, 64, 64)	(batch_size, 2, 64, 64)	Produces 2-channel image output

Lambda

Layer Type	Input Dimensions	Output Dimensions	Transformation Details
Lambda	(batch_size, 2, 64, 64)	(batch_size, 1, 64, 64)	Maps generator output to discriminator input

Too complex?
Too simple?

Discriminator

Layer Type	Input Dimensions	Output Dimensions	Activation/Other Details
Conv2d (1)	(batch_size, 1, 64, 64)	(batch_size, 64, 32, 32)	LeakyReLU (0.2), Dropout (0.3), Kernel=4
Conv2d (2)	(batch_size, 64, 32, 32)	(batch_size, 128, 16, 16)	BatchNorm, LeakyReLU (0.2), Dropout (0.3)
Flatten	(batch_size, 128, 16, 16)	(batch_size, $128 \times 16 \times 16$)	Flattens for FC layers
Fully Connected (1)	(batch_size, $128 \times 16 \times 16$)	(batch_size, 64)	LeakyReLU (0.2), Dropout (0.3)
Fully Connected (2)	(batch_size, 64)	(batch_size, 1)	Outputs real/fake score

IMPLEMENTATION DETAILS

- Other hyperparameters:

Generator Optimizer	Adam Learning rate: 0.0001
Discriminator Optimizer	Adam Learning rate: 0.00001
Batch Size	256
Training Size	40×256
Input Noise Dimension	100
Epochs	Total: 200 (with stopping if convergence achieved)
Weights for Generator Losses	[MSE, unitarity, d0, d1] = [1,1,10,10]
Device	GPU

How to optimize?