Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. **Machine learning**
8. Parton showers
9. Multijet merging

ML for event generation

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{g_i(x)} \right\rangle_{x \sim g_i(x)}$$

ML for event generation

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{g_i(x)} \right\rangle_{x \sim g_i(x)}$$

Amplitude surrogates

Train surrogate for **expensive matrix element calculation**, especially for loop ME.

Note: simple (non-NN) surrogate for loops already used in MG@NLO.

Can use reweighting to get unbiased integral.

ML for event generation

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{g_i(x)} \right\rangle_{x \sim g_i(x)}$$

Amplitude surrogates

Train surrogate for **expensive matrix element calculation**, especially for loop ME.

Note: simple (non-NN) surrogate for loops already used in MG@NLO.

Can use reweighting to get unbiased integral.

Neural importance sampling

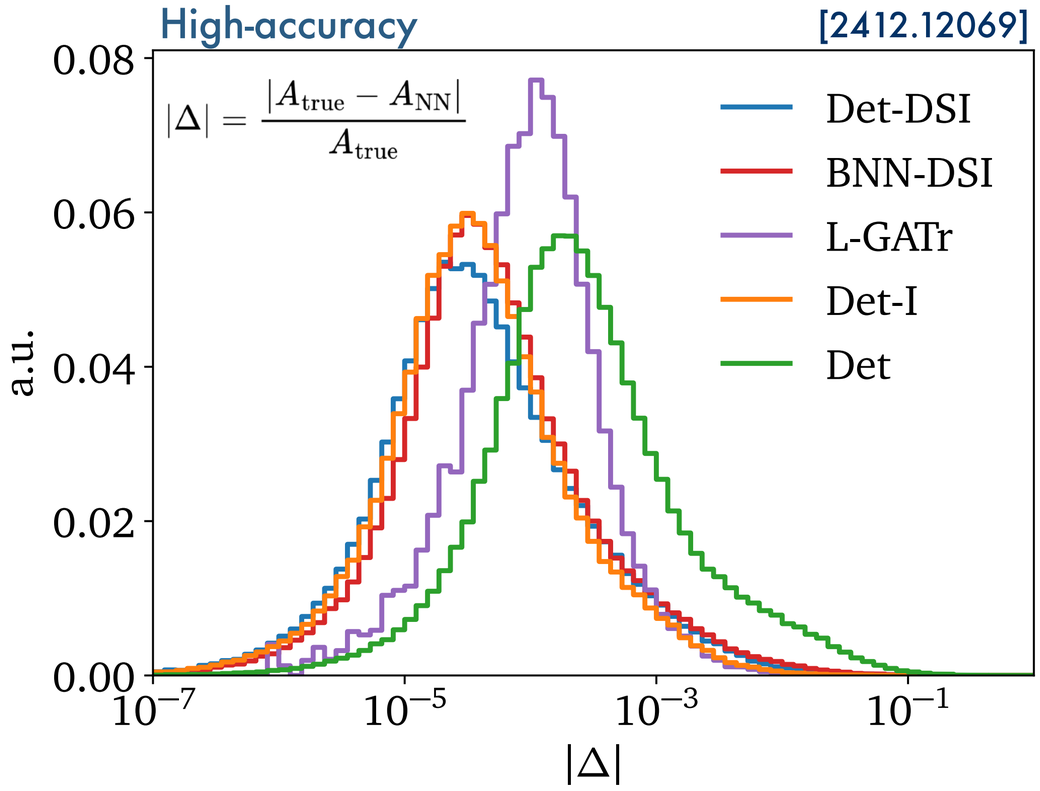
Use generative model to build more efficient sampler.

Need network architecture that implements learnable change of variables.

MadNIS

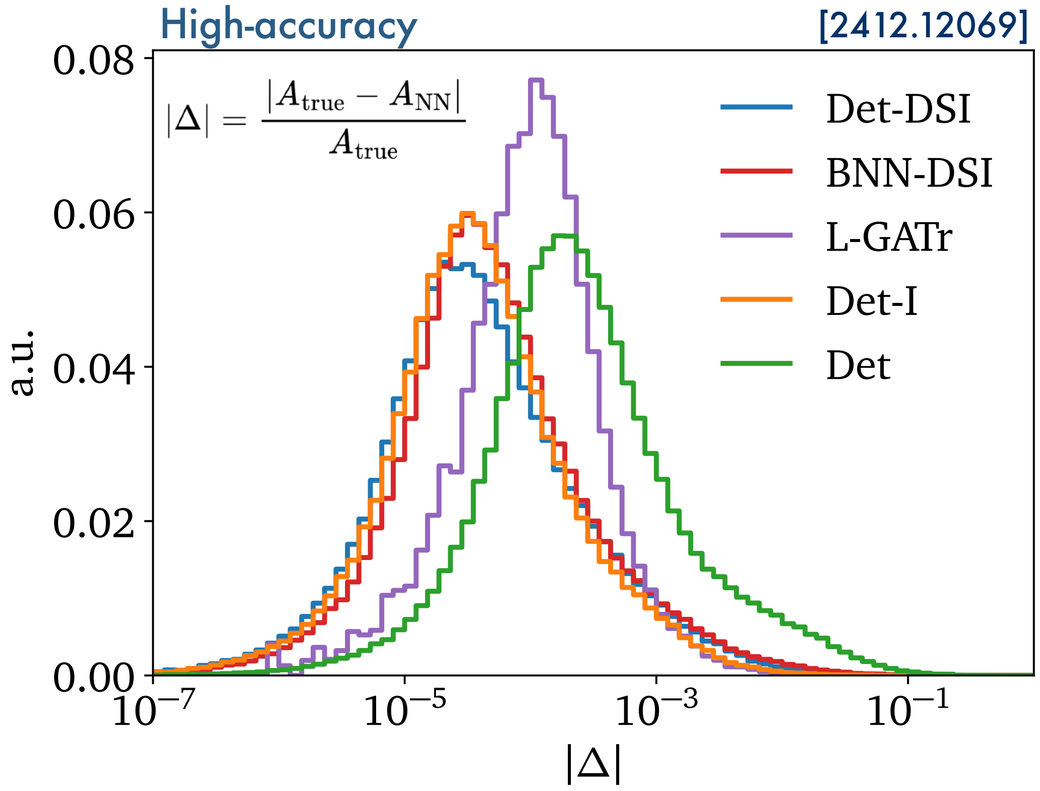
Amplitude Surrogates

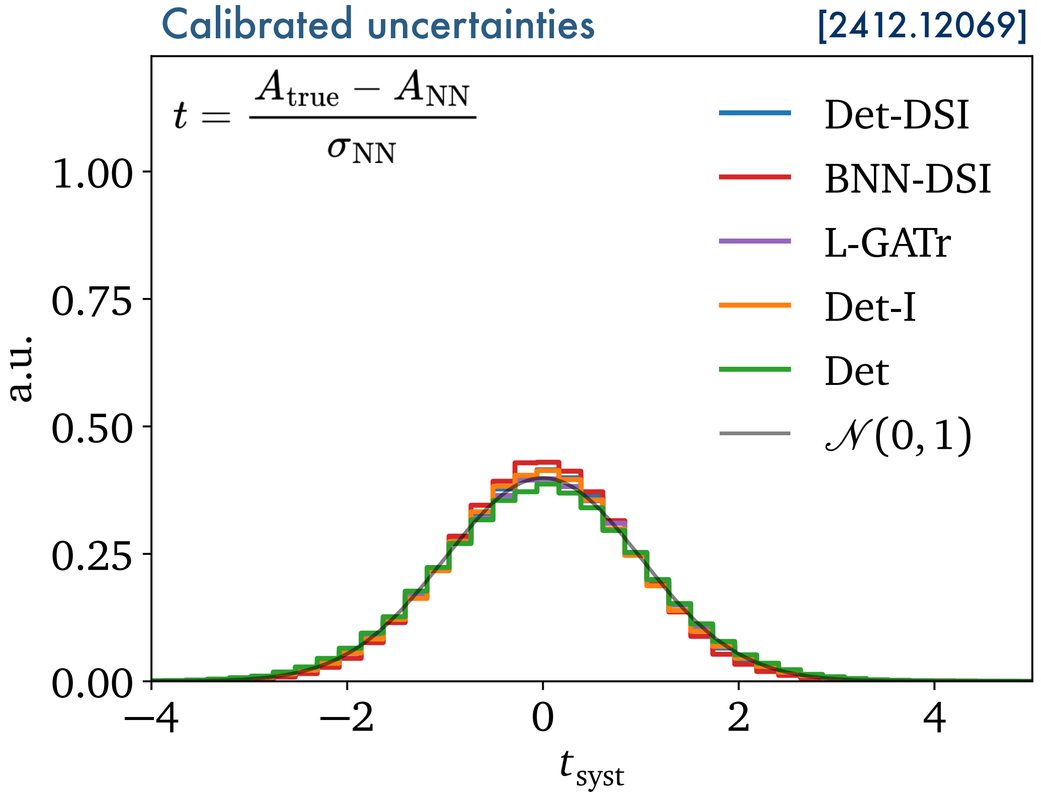
Precise and calibrated ML amplitudes



- Better architectures = better results

Precise and calibrated ML amplitudes

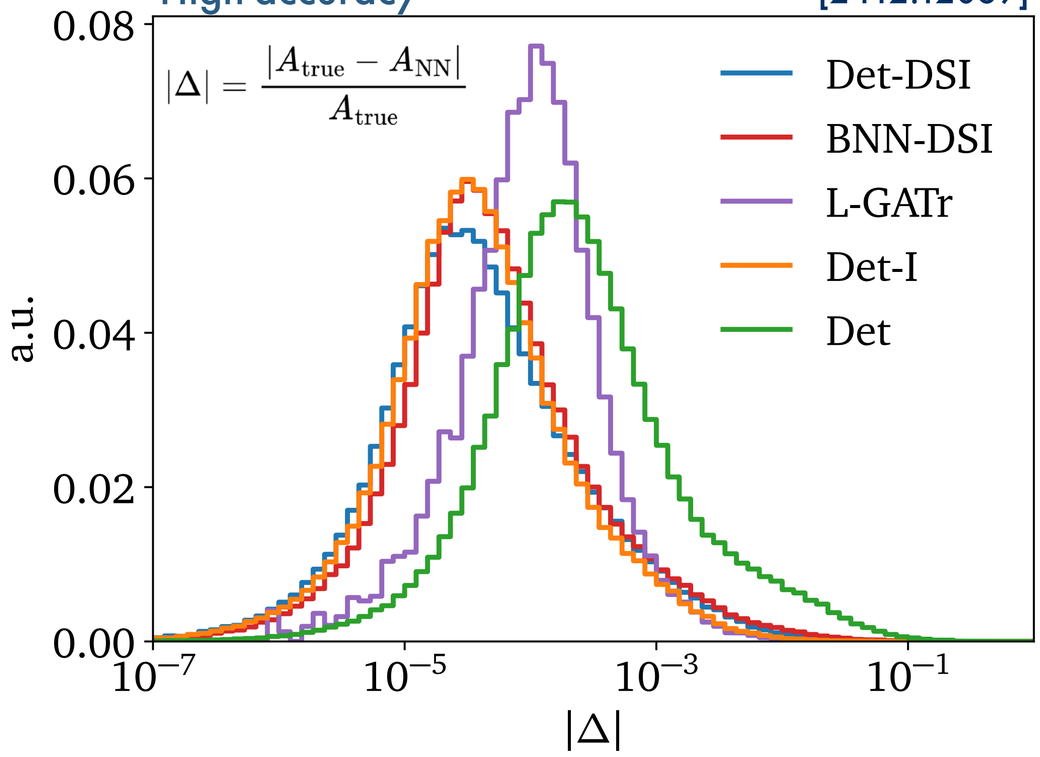




- Better architectures = better results
- Well calibrated uncertainties

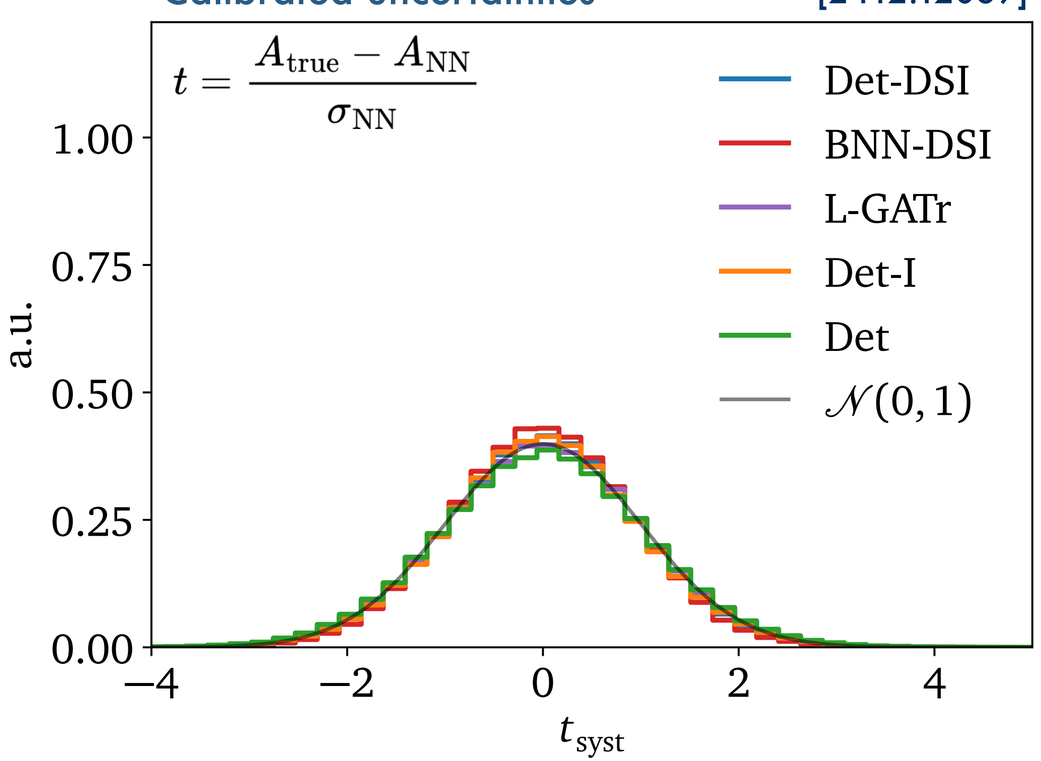
Precise and calibrated ML amplitudes

High-accuracy [2412.12069]



- Better architectures = better results

Calibrated uncertainties [2412.12069]



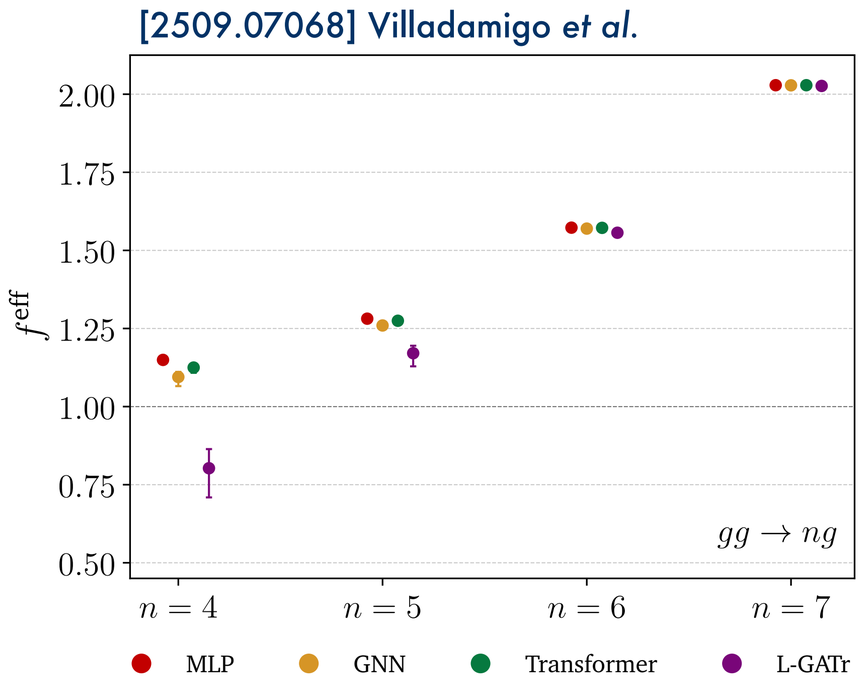
- Well calibrated uncertainties

→ Proof-of-concept works!

Making surrogates work at LO

Double-stage unweighting

[2509.07068] Villadamigo *et al.*

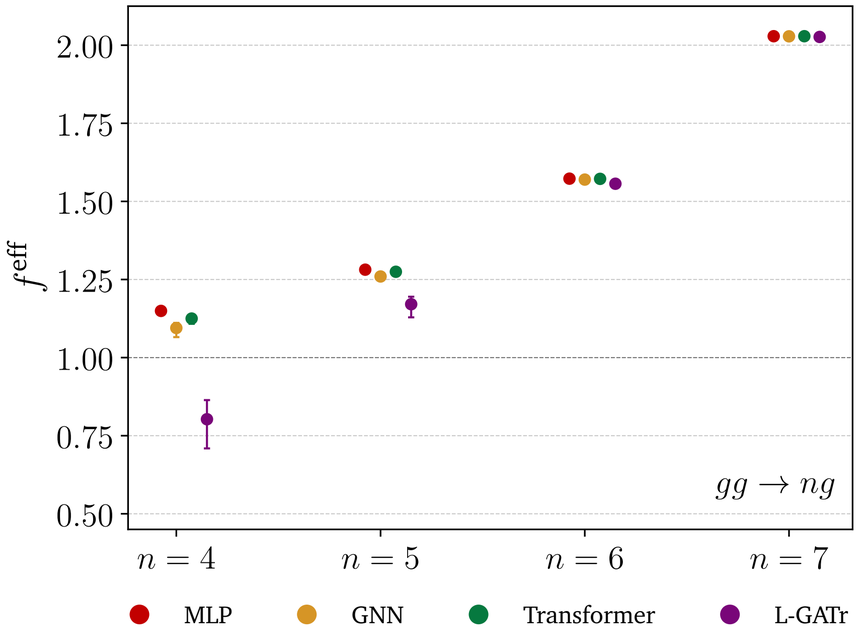


- Gains of up to factor 2 for large n
- Reweighing with truth → unbiased result

Making surrogates work at LO

Double-stage unweighting

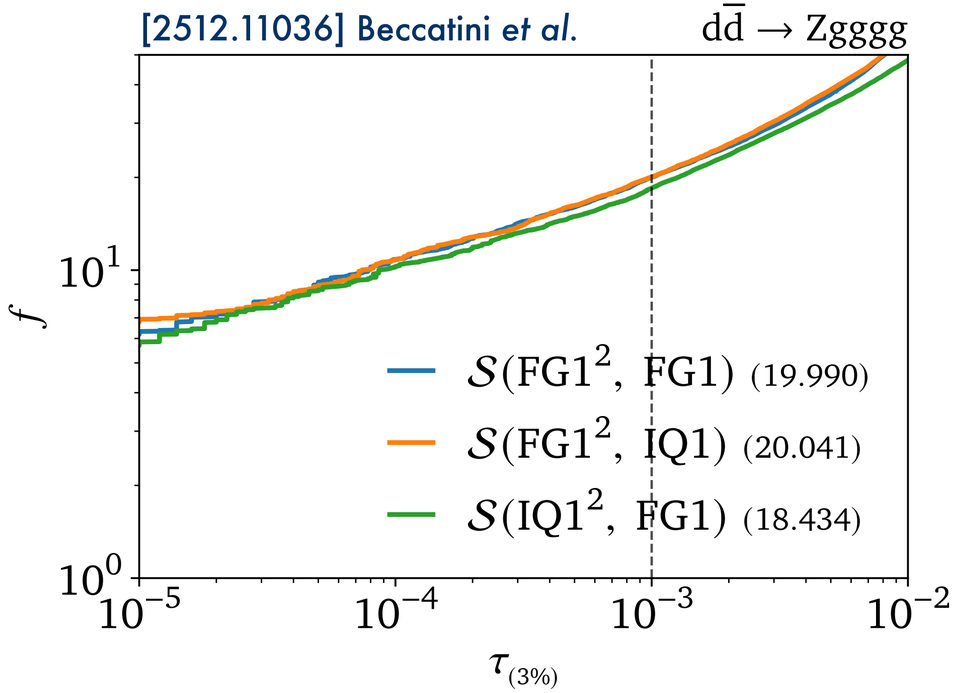
[2509.07068] Villadamigo *et al.*



- Gains of up to factor 2 for large n
- Reweighing with truth → unbiased result

Fixed-precision integration

[2512.11036] Beccatini *et al.*



- Larger gains of up to factor 20
- No reweighting→ biased but controlled!

The MadNIS Framework

Event generation in MadGraph

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i(x) \frac{w(x)}{p_i(x)} \right\rangle_{x\sim p_i(x)}$$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure,...
Analytic mappings + refine with *Vegas*

Event generation in MadGraph

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i(x) \frac{w(x)}{p_i(x)} \right\rangle_{x\sim p_i(x)}$$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Channel mappings

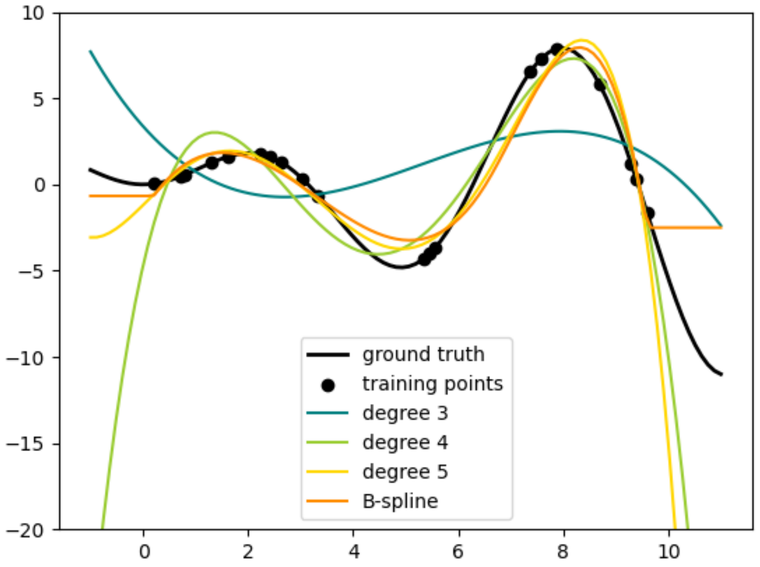
MadGraph: use amplitude structure,...
Analytic mappings + refine with *Vegas*

Use ML to improve?

A brief intro to generative AI

“All the impressive achievements of deep learning amount to just curve fitting” — Judea Pearl

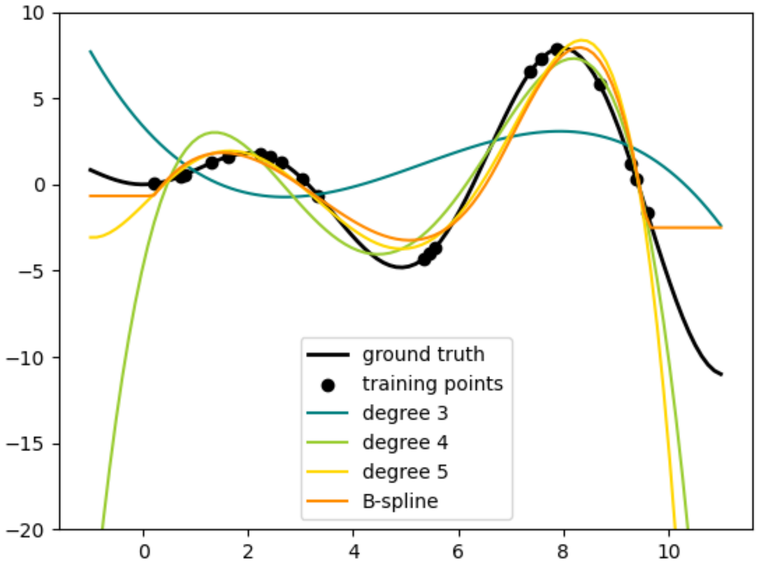
From this



A brief intro to generative AI

“All the impressive achievements of deep learning amount to just curve fitting” — Judea Pearl

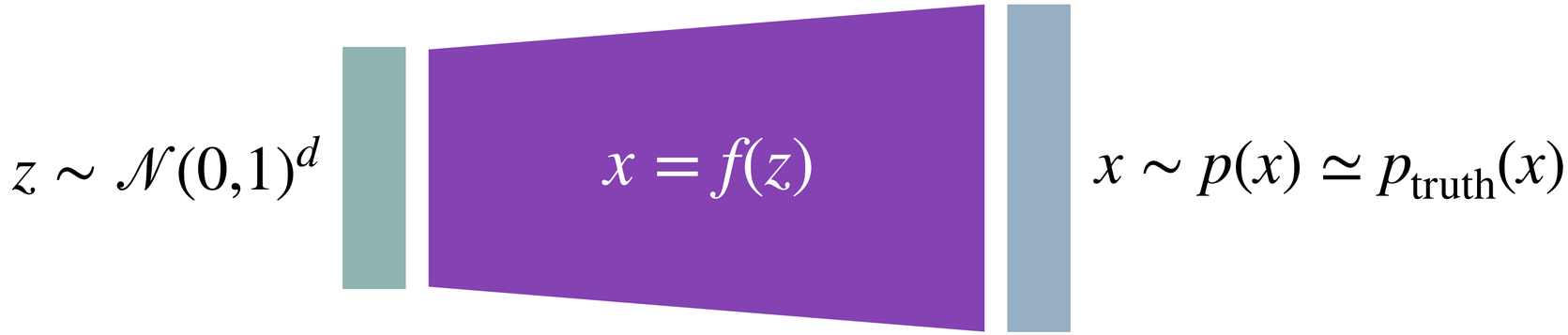
From this



Deep Neural Networks

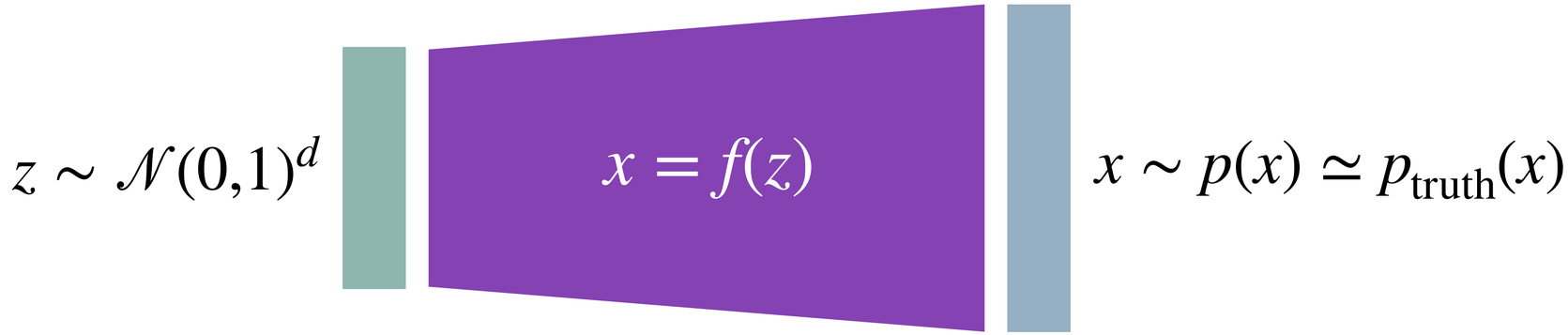
To this

A brief intro to generative AI



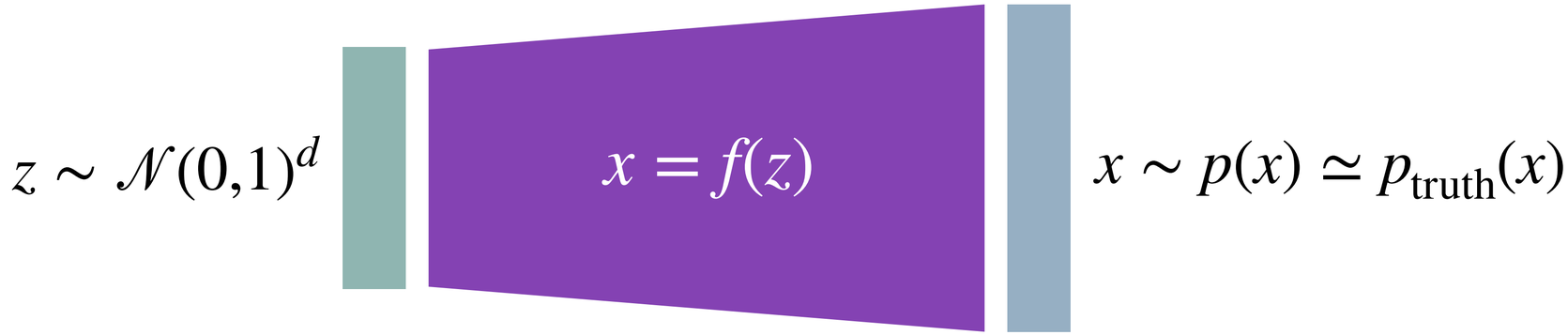
- A model $f(z)$ that can produce samples $x \sim p(x)$ starting from random noise z

A brief intro to generative AI



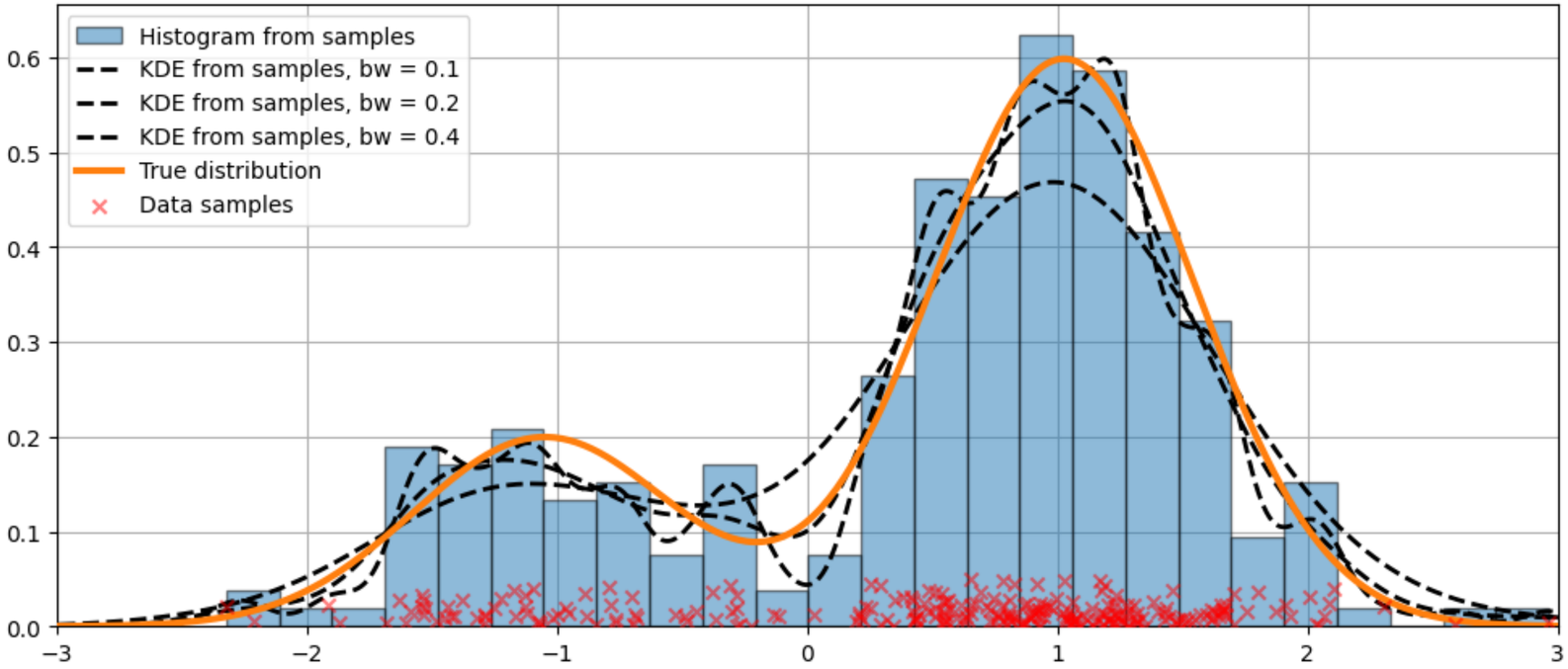
- A model $f(z)$ that can produce samples $x \sim p(x)$ starting from random noise z
- The distribution $p_{\text{truth}}(x)$ is usually given as:
  - ***explicit*** as function (e.g. $\mathrm{d}\sigma \propto$ differential cross-section)
  - ***implicit*** via a set of training data $\{x\} \sim p_{\text{data}}(x)$

A brief intro to generative AI

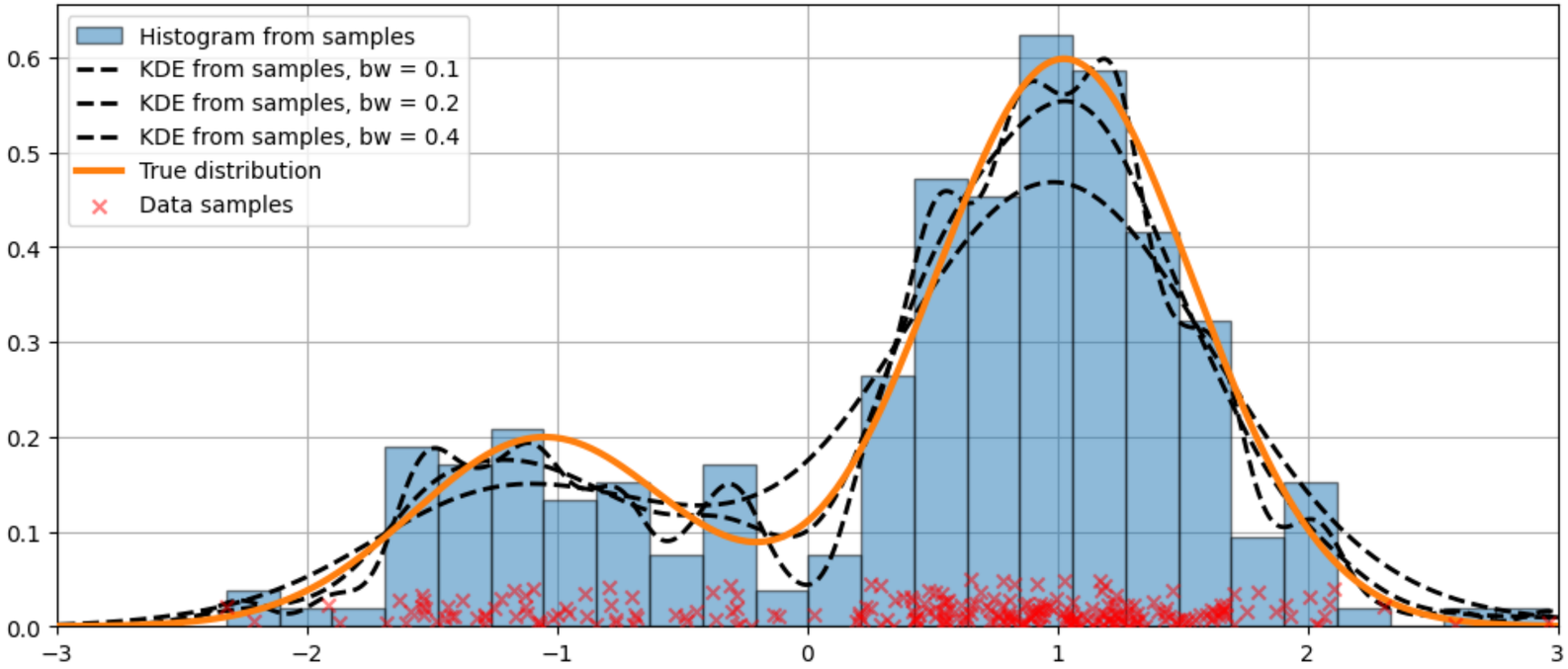


- A model $f(z)$ that can produce samples $x \sim p(x)$ starting from random noise z
- The distribution $p_{\text{truth}}(x)$ is usually given as:
  - ***explicit*** as function (e.g. $\mathrm{d}\sigma \propto$ differential cross-section)
  - ***implicit*** via a set of training data $\{x\} \sim p_{\text{data}}(x)$
    - ChatGPT: $\text{word} \sim p(\text{word} \,|\, \text{previous words})$
    - Midjourney: $\text{image} \sim p(\text{image} \,|\, \text{caption})$

Generative models and density estimation



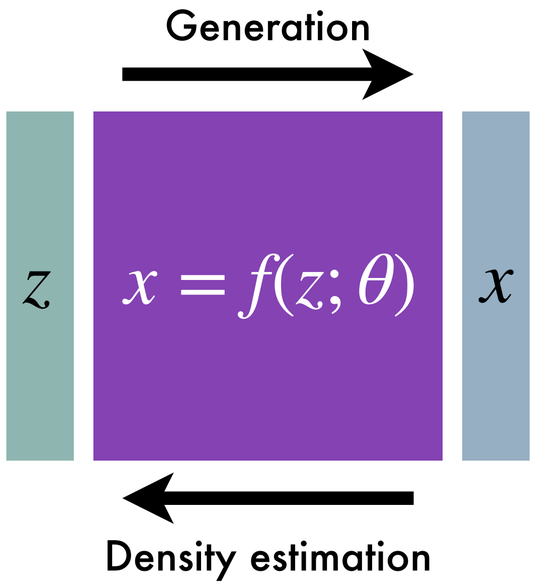
Generative models and density estimation



- Generative models closely connected to ***density estimation***
  - **Generative model:** sample from $p(x)$
  - **Density estimation:** estimate $p(x)$

Example: Normalizing flows

$$z \sim p_{\text{base}}(z) = \mathcal{N}(0,1)^d$$



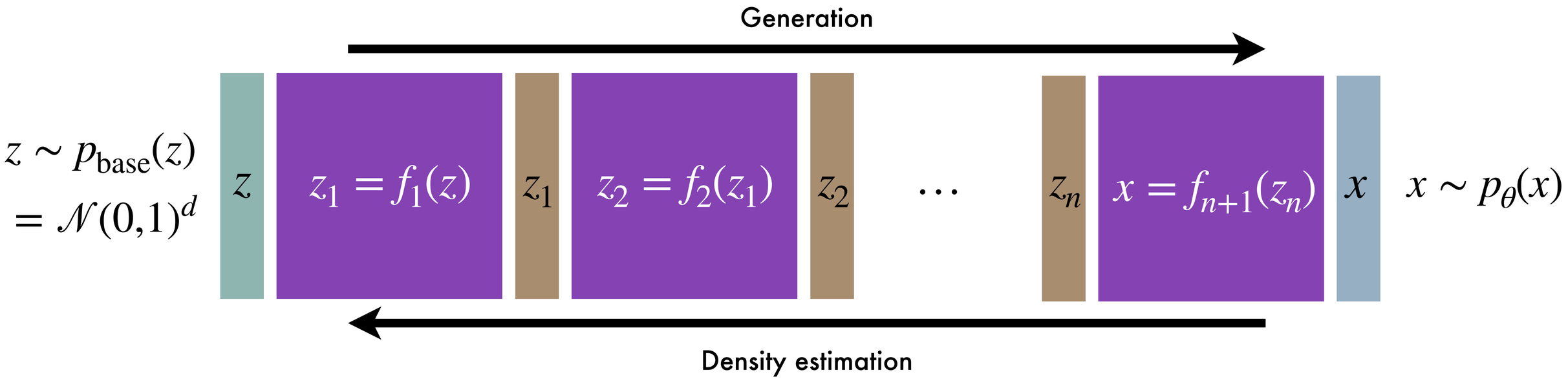
$$x \sim p_\theta(x)$$

Powerful class of ***density estimator*** that are also ***generative models***

- Family of invertible maps parametrized by neural networks $\log p_\theta(x) = \log p_{\text{base}}(z = \bar{f}_\theta(x)) \left| \frac{\partial z}{\partial x} \right|$
- (Usually) train with maximum likelihood objective $L = -\sum_{x_i \in \text{data}} \log p_\theta(x_i)$

(need tractable Jacobian!)

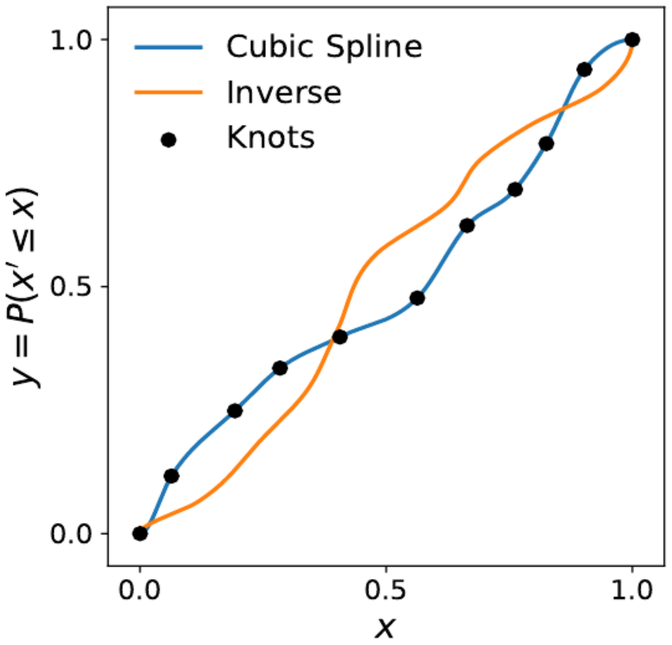
Example: Normalizing flows



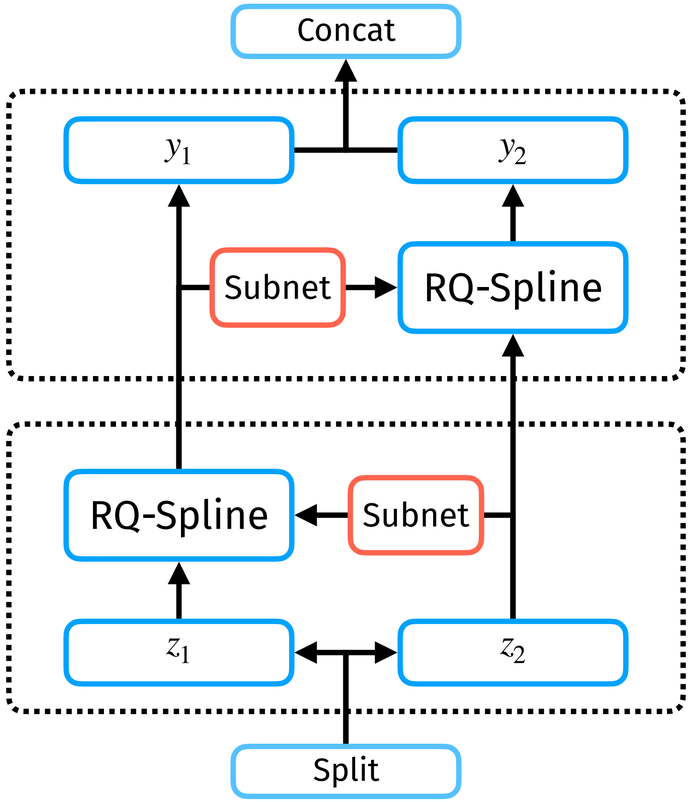
Powerful class of ***density estimator*** that are also ***generative models***

- Family of invertible maps parametrized by neural networks $\log p_\theta(x) = \log p_{\text{base}}(z = \bar{f}_\theta(x)) \left| \frac{\partial z}{\partial x} \right|$
- (Usually) train with maximum likelihood objective $L = -\sum_{x_i \in \text{data}} \log p_\theta(x_i)$ (need tractable Jacobian!)
- ***Compose multiple blocks for greater expressivity***

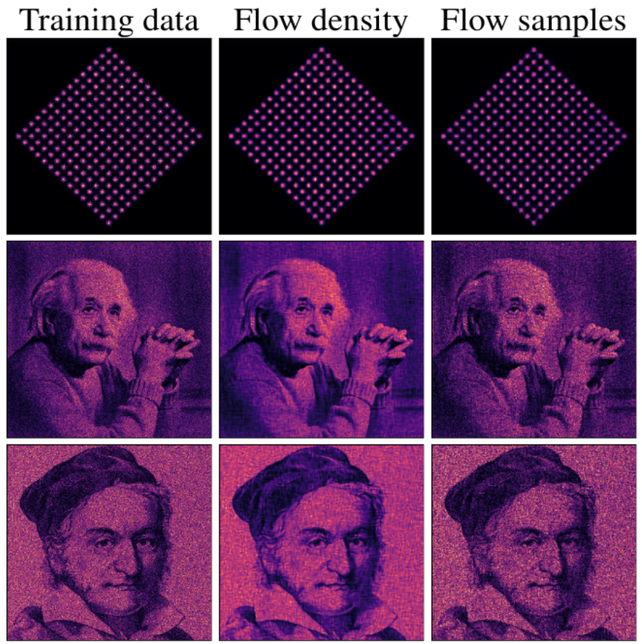
Spline coupling blocks



Model CDF as RQ spline: learn knot positions with neural networks



Model correlations by splitting inputs and using sub-networks



Learns probability distributions with high detail and accuracy

[Durkan et al, 1906.04032] [Durkan et al, 1906.02145]

Event generation in MadGraph + MadNIS

+ MadNIS

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i(x) \frac{w(x)}{p_i^\theta(x)} \right\rangle_{x\sim p_i^\theta(x)}$$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Learned channel mappings

MadGraph: use amplitude structure,...
Analytic mappings + ~~refine with *Vegas*~~

refine with *Normalizing Flow*

[Gao *et al,* 2001.05486+2001.10028] [Bothmann *et al,* 2001.05478+2506.18987]
[Winterhalder *et al,* 2112.09145] [Deutschmann *et al,* 2401.09069] [Janssen *et al.,* 2505.13608]

Event generation in MadGraph + MadNIS

+ MadNIS

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $\mathrm{d}\sigma/\mathrm{d}x$

$$I = \sum_i \left\langle \alpha_i^{\xi}(x) \frac{w(x)}{p_i^{\theta}(x)} \right\rangle_{x\sim p_i^{\theta}(x)}$$

Learned channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

$$\alpha_i(x) \to \alpha_i^{\xi}(x) = \alpha_i^{\mathrm{MG}}(x) \cdot K_i^{\xi}(x)$$

parametrize with *neural network*

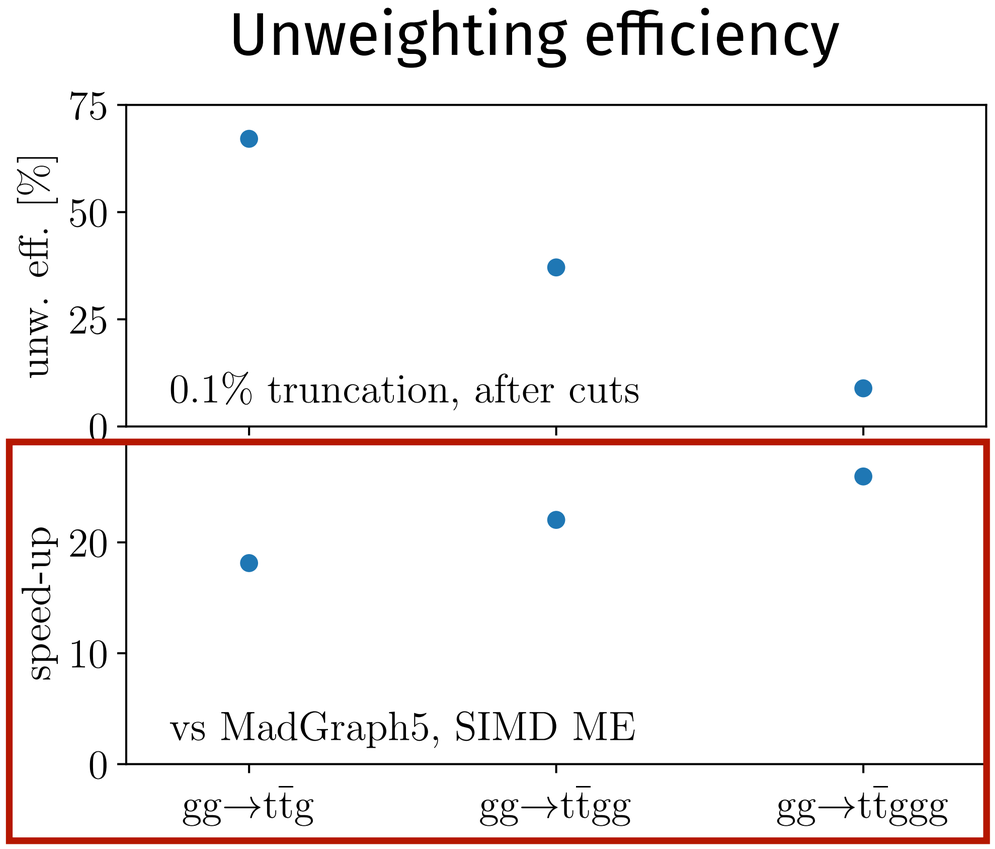
Learned channel mappings

MadGraph: use amplitude structure,...

Analytic mappings + ~~refine with *Vegas*~~

refine with *Normalizing Flow*

MadGraph + MadNIS at LO



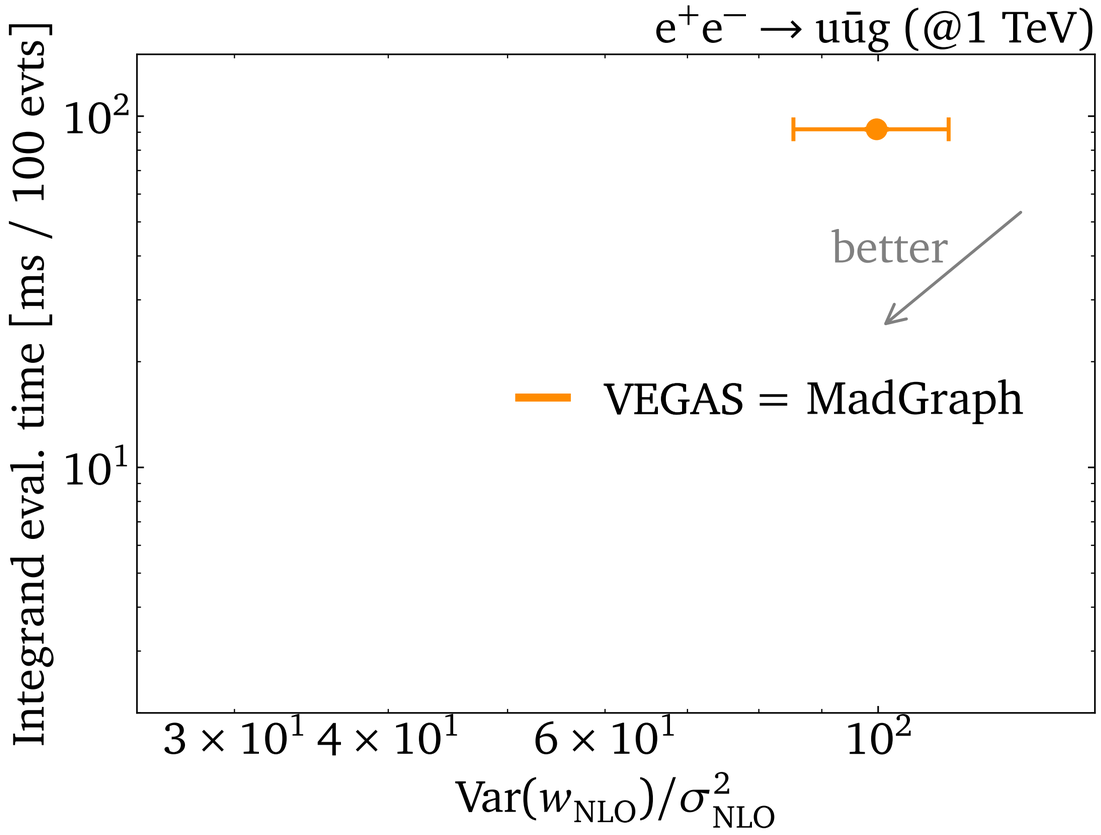
- improved unweighting efficiency
  → **large speed-up across processes**
- will become default in many cases
  → starting at $2 \to 3$ or $2 \to 4$

(preliminary)

MadNIS@LO [2212.06172, 2311.01548, 2408.01486]

Putting it all together at NLO

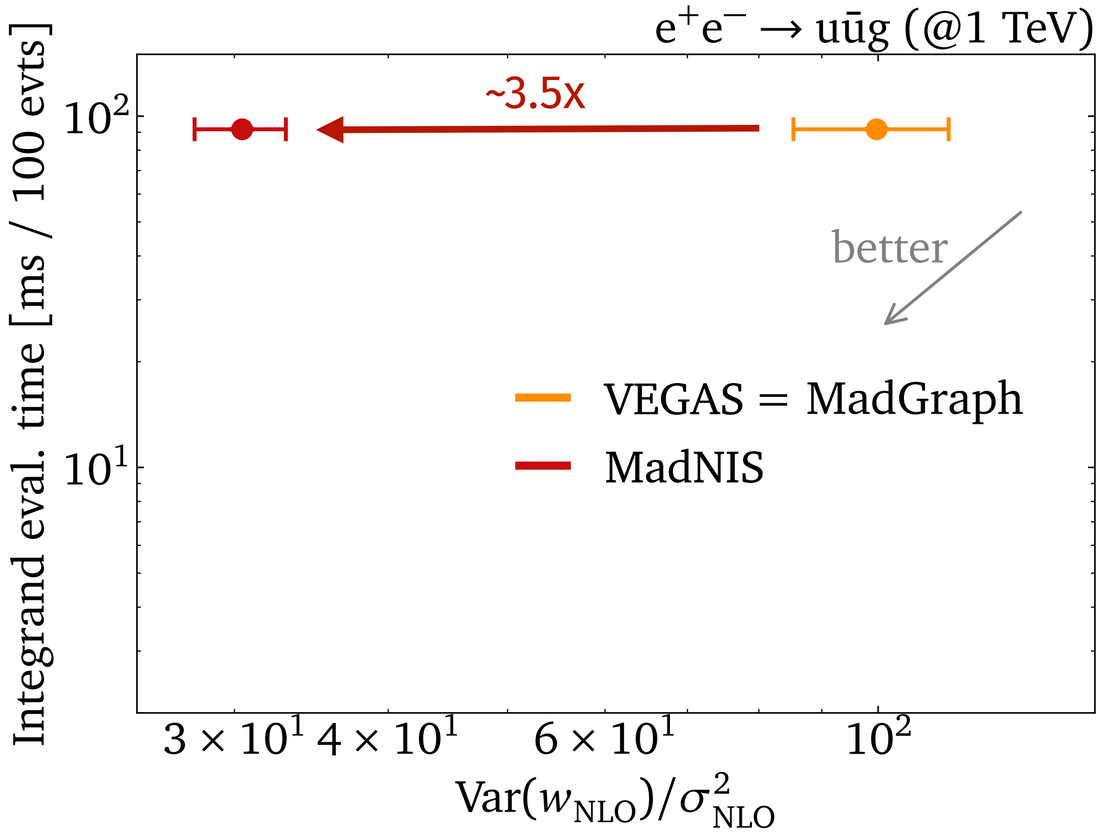
MadNIS meets surrogates at NLO



MadNIS@NLO [2603.22407] De Crescenzo et al.

NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

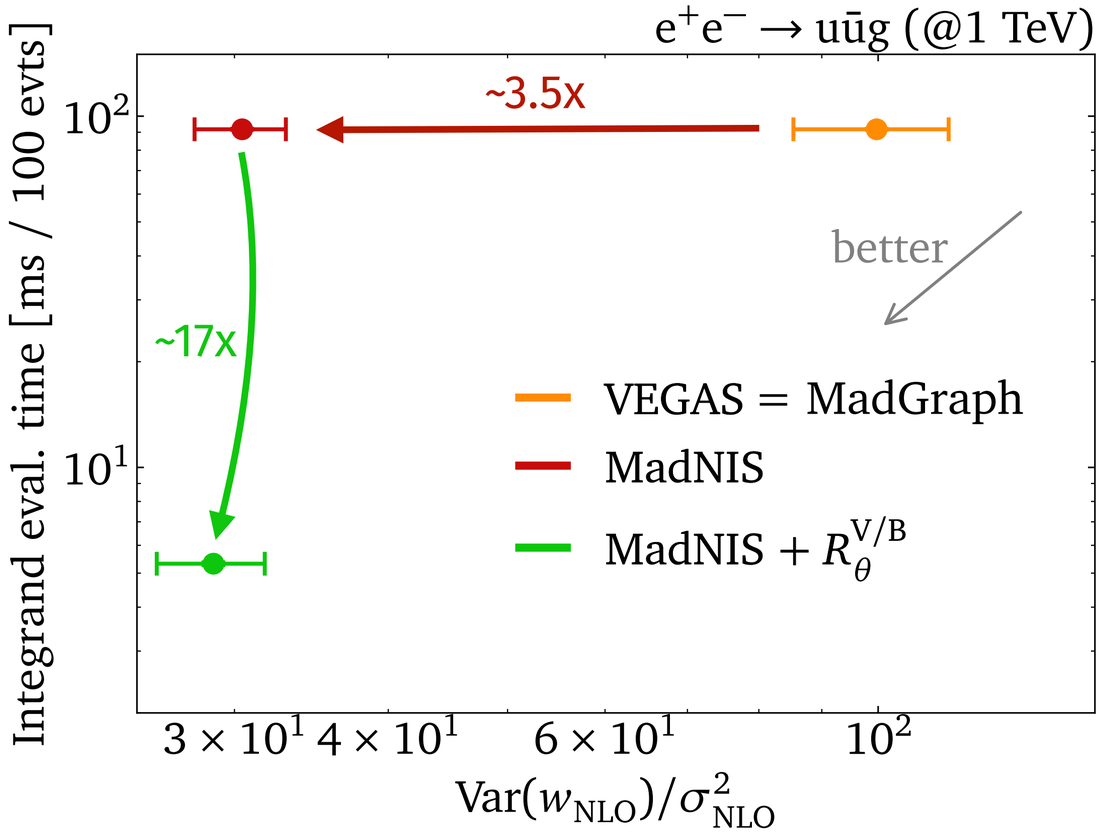
MadNIS meets surrogates at NLO



MadNIS@NLO [2603.22407] De Crescenzo et al.

NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

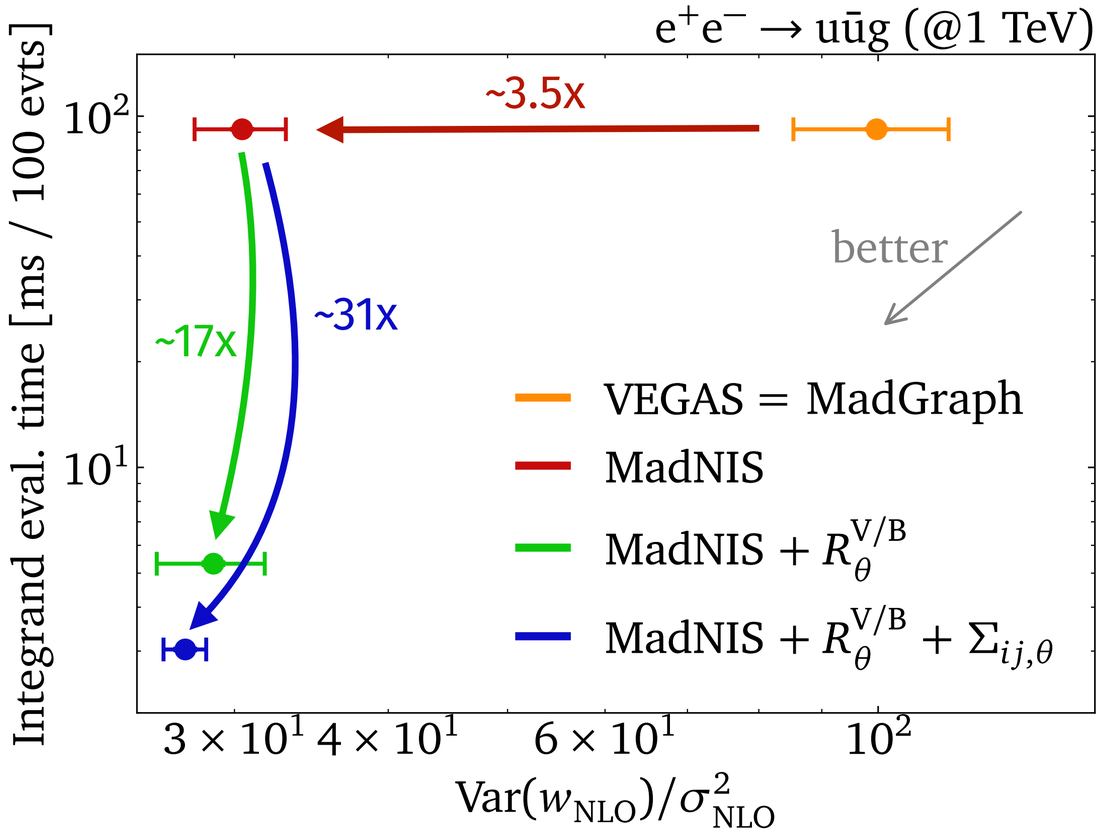
MadNIS meets surrogates at NLO



MadNIS@NLO [2603.22407] De Crescenzo et al.

NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

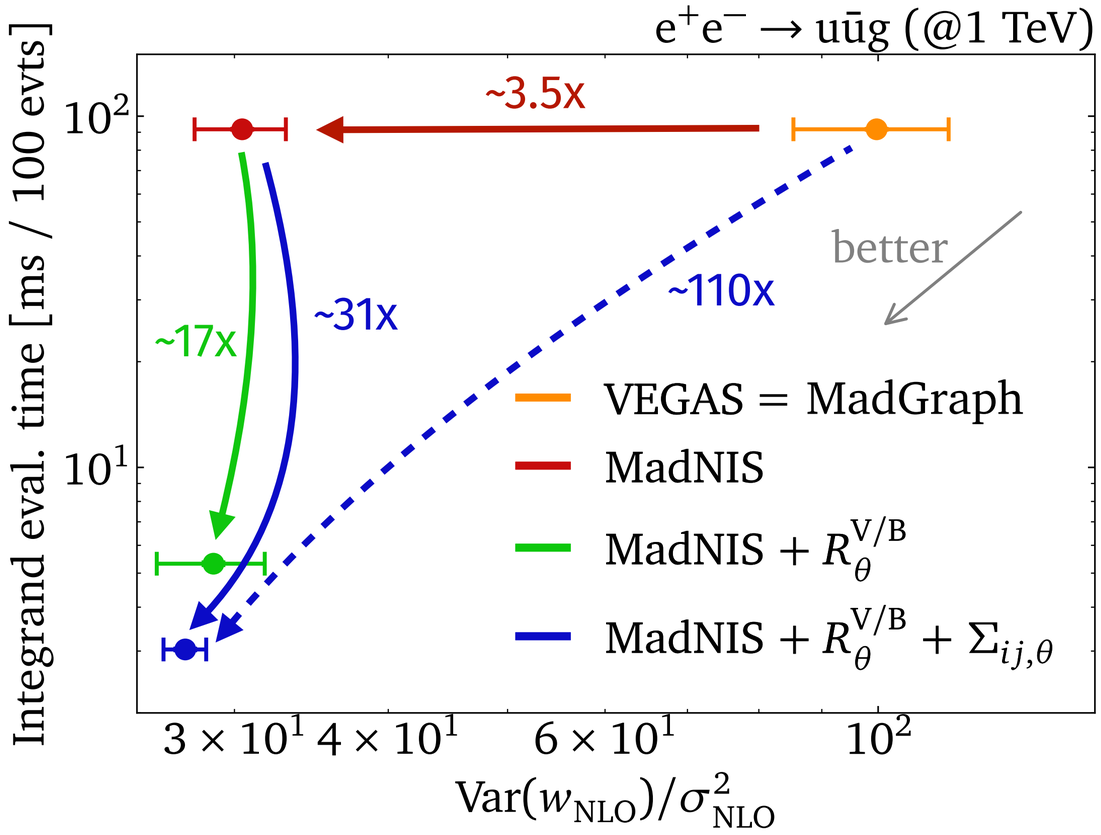
MadNIS meets surrogates at NLO



MadNIS@NLO [2603.22407] De Crescenzo et al.

NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

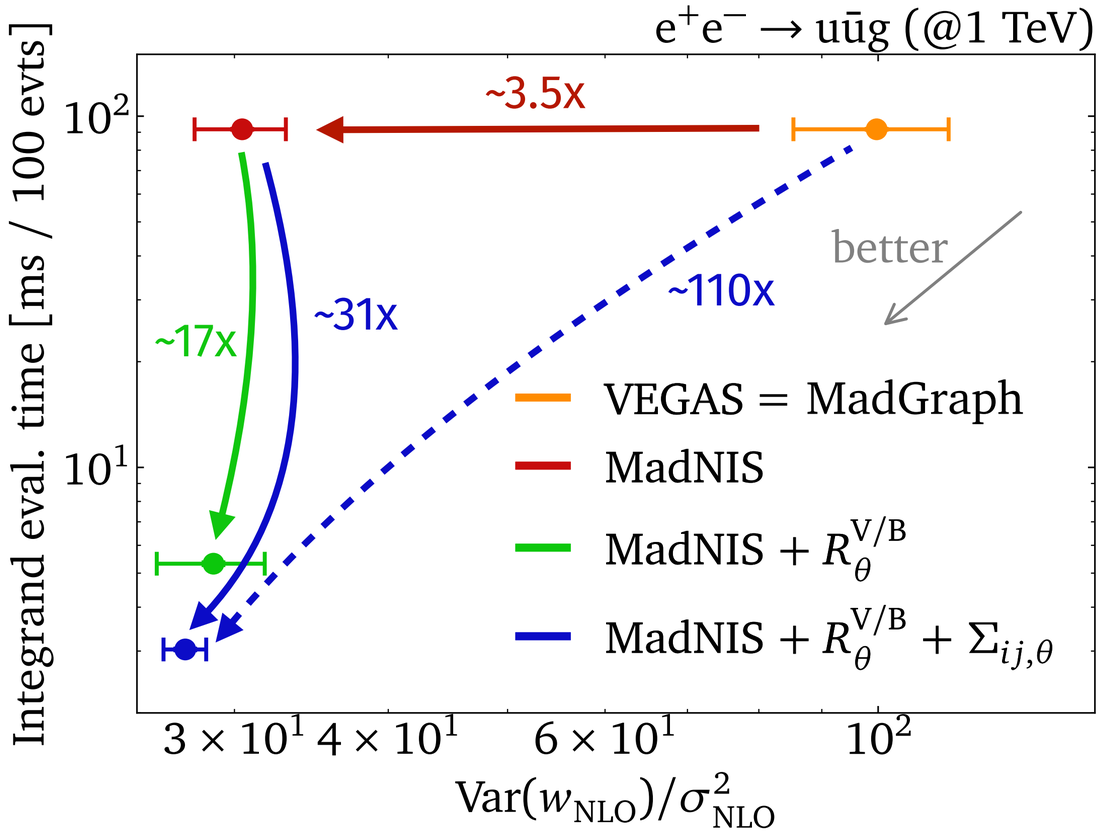
MadNIS meets surrogates at NLO

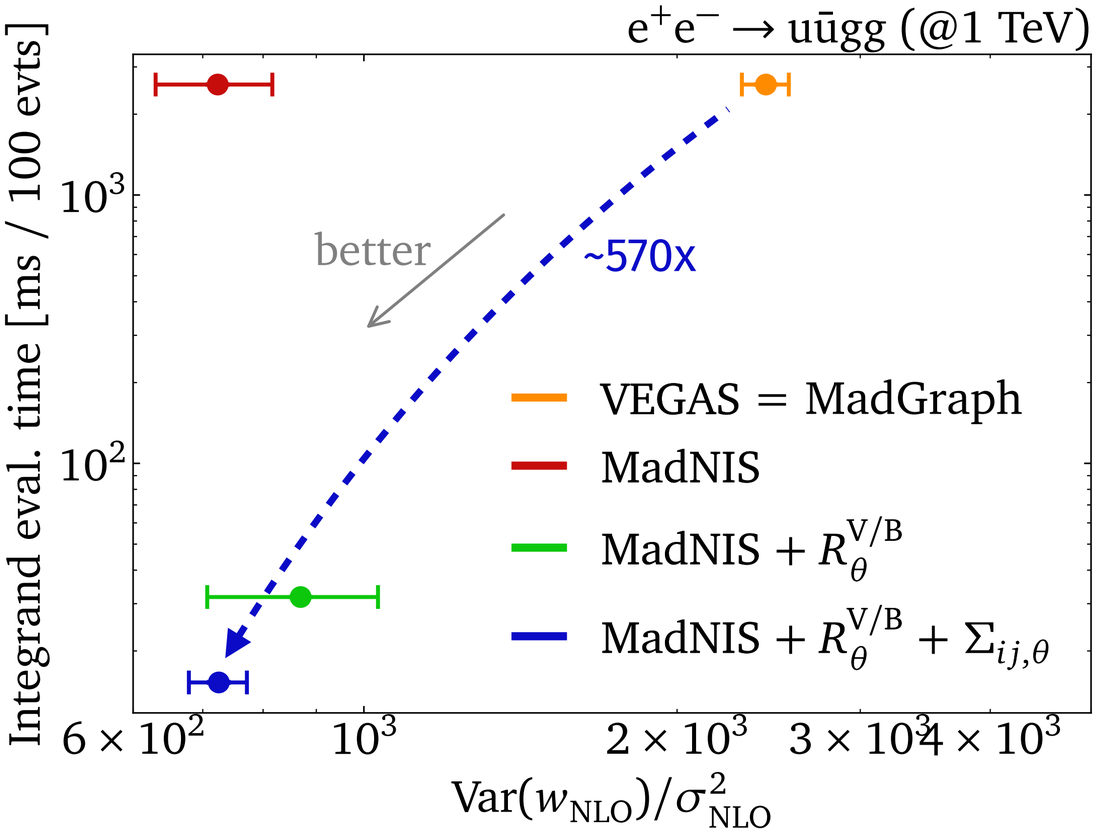


MadNIS@NLO [2603.22407] De Crescenzo et al.

NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

MadNIS meets surrogates at NLO





MadNIS@NLO [2603.22407] De Crescenzo et al.

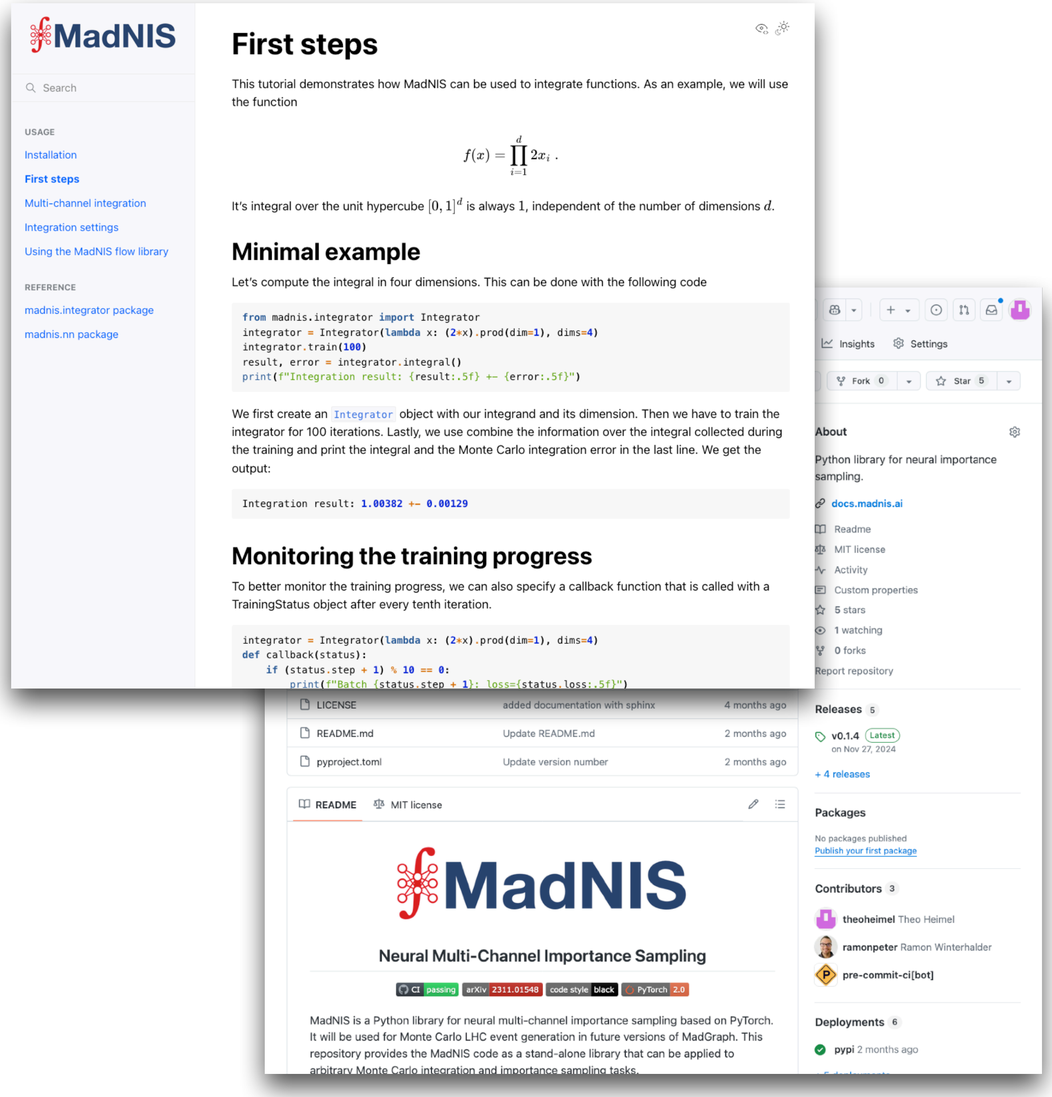
NIS@NLO [Bothmann *et al*, 2001.05478]
NIS@NNLO [Janssen *et al.*, 2505.13608]

Standalone Python module

- **MadNIS** as a Python package
  → apply to your own integration tasks
- From simple single-channel integrals to complex multi-channel setups
- Easy-to-use implementation of normalizing flows

https://madnis.ai/

```
pip install madnis
```



Summary

- Two orthogonal approaches to use ML in event generation

  → fast **matrix element surrogates**

  → better proposal distribution with **neural importance sampling**

- **Normalizing flows** learn invertible transformation with **known Jacobian**

  → drop-in replacement for VEGAS

- Improve multi-channel decomposition with NN

  → especially useful for processes with large interference terms

- Large speed-ups compared to VEGAS for wide range of processes

Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. **Parton showers**
9. Multijet merging

Parton shower

Goal: turn a hard partonic event into a more realistic event with soft and collinear radiation.

- The Parton shower ***adds explicit branchings***:

$$q \to qg, \quad g \to gg, \quad g \to q\bar{q}, \ldots$$

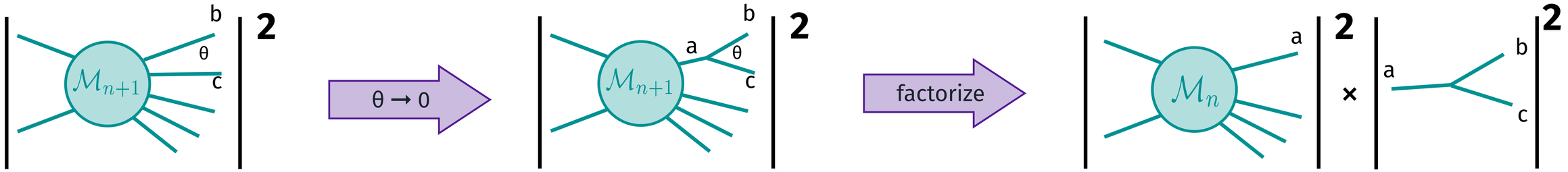
- But, if unitary, it does ***not add a new inclusive contribution***:

$$\mathrm{d}\sigma_{\mathrm{LO}} \longrightarrow \mathrm{d}\sigma_{\mathrm{LO}} \left[P_0 + P_1 + P_2 + \cdots\right], \quad P_0 + P_1 + P_2 + \cdots = 1$$

- The Born cross section is redistributed into exclusive event classes: 0 emissions, 1 emissions, 2 emissions, ...

- What ***changes are exclusive observables***:
  jet multiplicities, transverse momenta, event shapes, etc.

Collinear factorization



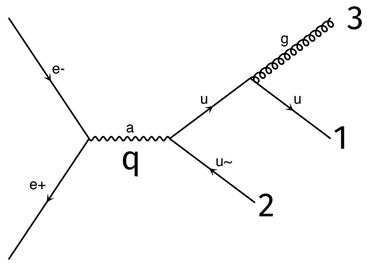
- Limit θ → 0: contribution from a nearly ***on-shell parent parton*** dominates
- Branching happens on very long time scale compared to hard process
- Can’t change picture set up by hard process
  → process must ***factorize*** as (hard process) × (branching probability)
  → the leading collinear behavior is ***universal***!

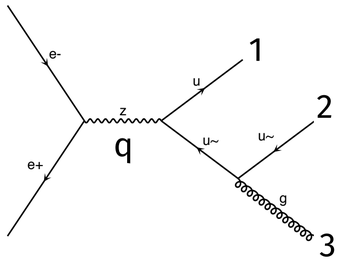
$$\frac{1}{(p_b + p_c)^2} \simeq \frac{1}{2E_b E_c(1 - \cos\theta)}$$

soft and collinear divergencies

Factorization example

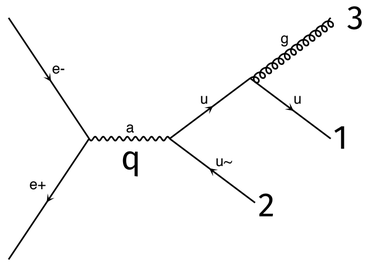
$$e^+e^- \to q\bar{q}g$$

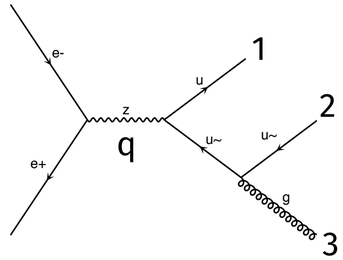




Factorization example

$$e^+e^- \to q\bar{q}g$$



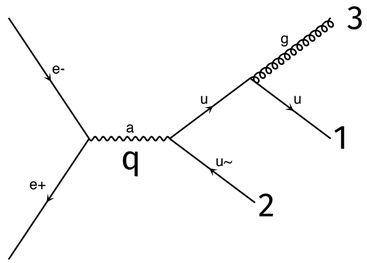


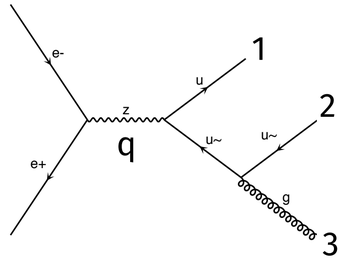
$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

$$x_1 = 2k_1 \cdot q/q^2 = 2E_q/\sqrt{s}$$
$$x_2 = 2k_2 \cdot q/q^2 = 2E_{\bar{q}}/\sqrt{s}$$
$$x_3 = 2k_3 \cdot q/q^2 = 2E_g/\sqrt{s}$$
$$x_1 + x_2 + x_3 = 2$$

Factorization example

$$e^+e^- \to q\bar{q}g$$





$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

$$x_1 = 2k_1 \cdot q/q^2 = 2E_q/\sqrt{s}$$
$$x_2 = 2k_2 \cdot q/q^2 = 2E_{\bar{q}}/\sqrt{s}$$
$$x_3 = 2k_3 \cdot q/q^2 = 2E_g/\sqrt{s}$$
$$x_1 + x_2 + x_3 = 2$$

Divergent at $x_1 = 1$ and $x_2 = 1$

- **soft** divergencies
- **collinear** divergencies

$$1 - x_1 = \frac{x_2 x_3}{2}(1 - \cos\theta_{23})$$
$$1 - x_2 = \frac{x_1 x_3}{2}(1 - \cos\theta_{13})$$

Factorization example

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

$$x_1 = 2k_1 \cdot q/q^2 = 2E_q/\sqrt{s}$$
$$x_2 = 2k_2 \cdot q/q^2 = 2E_{\bar{q}}/\sqrt{s}$$
$$x_3 = 2k_3 \cdot q/q^2 = 2E_g/\sqrt{s}$$
$$x_1 + x_2 + x_3 = 2$$

Factorization example

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

$$x_1 = 2k_1 \cdot q/q^2 = 2E_q/\sqrt{s}$$
$$x_2 = 2k_2 \cdot q/q^2 = 2E_{\bar{q}}/\sqrt{s}$$
$$x_3 = 2k_3 \cdot q/q^2 = 2E_g/\sqrt{s}$$
$$x_1 + x_2 + x_3 = 2$$

Change variables to x_3 and $\cos\theta_{13}$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_3\,\mathrm{d}\cos\theta_{13}} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \left(\frac{2}{\sin^2\theta_{13}} \frac{1+(1-x_3)^2}{x_3} - x_3 \right)$$

Factorization example

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)}$$

$$x_1 = 2k_1 \cdot q/q^2 = 2E_q/\sqrt{s}$$
$$x_2 = 2k_2 \cdot q/q^2 = 2E_{\bar{q}}/\sqrt{s}$$
$$x_3 = 2k_3 \cdot q/q^2 = 2E_g/\sqrt{s}$$
$$x_1 + x_2 + x_3 = 2$$

Change variables to x_3 and $\cos\theta_{13}$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_3\,\mathrm{d}\cos\theta_{13}} = \sigma_0 C_F \frac{\alpha_s}{2\pi} \left(\frac{2}{\sin^2\theta_{13}} \frac{1+(1-x_3)^2}{x_3} - x_3 \right)$$

Collinear limit:
Integral splits into two parts
→ radiate from jet 1 or 2

$$\begin{aligned}
\frac{2\,\mathrm{d}\cos\theta_{13}}{\sin^2\theta_{13}} &= \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{13}}{1+\cos\theta_{13}} \\
&\simeq \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{23}}{1-\cos\theta_{23}} \\
&\simeq \frac{\mathrm{d}\theta_{13}^2}{\theta_{13}^2} + \frac{\mathrm{d}\theta_{23}^2}{\theta_{23}^2}
\end{aligned}$$

Factorization example

Change variables to x_3 and $\cos\theta_{13}$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_3\,\mathrm{d}\cos\theta_{13}} = \sigma_0 C_F \frac{\alpha_s}{2\pi}\left(\frac{2}{\sin^2\theta_{13}}\,\frac{1+(1-x_3)^2}{x_3} - x_3\right)$$

Collinear limit:
Integral splits into two parts
→ radiate from jet 1 or 2

$$\begin{aligned}\frac{2\,\mathrm{d}\cos\theta_{13}}{\sin^2\theta_{13}} &= \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{13}}{1+\cos\theta_{13}} \\ &\simeq \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{23}}{1-\cos\theta_{23}} \\ &\simeq \frac{\mathrm{d}\theta_{13}^2}{\theta_{13}^2} + \frac{\mathrm{d}\theta_{23}^2}{\theta_{23}^2}\end{aligned}$$

Factorization example

Change variables to x_3 and $\cos\theta_{13}$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_3\,\mathrm{d}\cos\theta_{13}} = \sigma_0 C_F \frac{\alpha_s}{2\pi}\left(\frac{2}{\sin^2\theta_{13}}\,\frac{1+(1-x_3)^2}{x_3} - x_3\right)$$

Collinear limit:
Integral splits into two parts
→ radiate from jet 1 or 2

$$\begin{aligned}\frac{2\,\mathrm{d}\cos\theta_{13}}{\sin^2\theta_{13}} &= \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{13}}{1+\cos\theta_{13}} \\ &\simeq \frac{\mathrm{d}\cos\theta_{13}}{1-\cos\theta_{13}} + \frac{\mathrm{d}\cos\theta_{23}}{1-\cos\theta_{23}} \\ &\simeq \frac{\mathrm{d}\theta_{13}^2}{\theta_{13}^2} + \frac{\mathrm{d}\theta_{23}^2}{\theta_{23}^2}\end{aligned}$$

Energy fraction z
→ **Generic formula**

$$\mathrm{d}\sigma = \sigma_0 \sum_{\text{jets}} C_F \frac{\alpha_s}{2\pi}\frac{\mathrm{d}\theta^2}{\theta^2}\,\mathrm{d}z\,\frac{1+(1-z)^2}{z}$$

Collinear factorization

Universal factorization in the collinear limit θ → 0

$$|\mathcal{M}_{n+1}|^2 \, \mathrm{d}\Phi_{n+1} \simeq |\mathcal{M}_n|^2 \, \mathrm{d}\Phi_n \times \frac{\mathrm{d}t}{t} \, \mathrm{d}z \, \frac{\mathrm{d}\phi}{2\pi} \, \frac{\alpha_s}{2\pi} \, P_{a\to bc}(z)$$

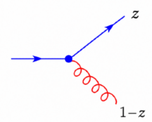
- t is the evolution parameter
- z is the energy fraction
- α_s is evaluated at scale t
- $P_{a\to bc}$ is the splitting kernel (controls soft behavior)

Splitting kernels

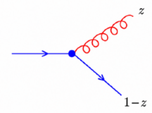
$$|\mathcal{M}_{n+1}|^2 \, \mathrm{d}\Phi_{n+1} \simeq |\mathcal{M}_n|^2 \, \mathrm{d}\Phi_n \times \frac{\mathrm{d}t}{t} \, \mathrm{d}z \, \frac{\mathrm{d}\phi}{2\pi} \, \frac{\alpha_s}{2\pi} \, P_{a\to bc}(z)$$

Four splitting functions depending on the type of branching (Altarelli-Parisi)

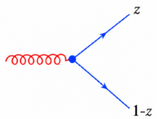
$$P_{q\to qg}(z) = C_F \left[\frac{1+z^2}{1-z}\right]$$



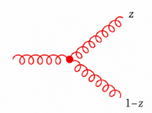
$$P_{q\to gq}(z) = C_F \left[\frac{1+(1-z)^2}{z}\right]$$



$$P_{g\to q\bar{q}}(z) = T_R \left[z^2+(1-z)^2\right]$$



$$P_{g\to gg}(z) = C_A \left[\frac{z}{1-z} + \frac{1-z}{z} + z(1-z)\right]$$



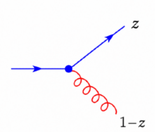
$$C_F = \frac{4}{3}, \qquad C_A = 3, \qquad T_R = \frac{1}{2}.$$

Splitting kernels

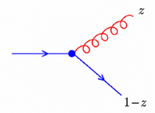
$$|\mathcal{M}_{n+1}|^2 \, \mathrm{d}\Phi_{n+1} \simeq |\mathcal{M}_n|^2 \, \mathrm{d}\Phi_n \times \frac{\mathrm{d}t}{t} \, \mathrm{d}z \, \frac{\mathrm{d}\phi}{2\pi} \, \frac{\alpha_s}{2\pi} \, P_{a\to bc}(z)$$

Four splitting functions depending on the type of branching (Altarelli-Parisi)

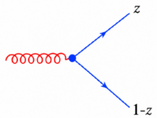
$$P_{q\to qg}(z) = C_F \left[\frac{1+z^2}{1-z}\right]$$



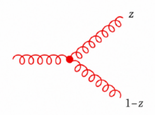
$$P_{q\to gq}(z) = C_F \left[\frac{1+(1-z)^2}{z}\right]$$



$$P_{g\to q\bar{q}}(z) = T_R \left[z^2+(1-z)^2\right]$$



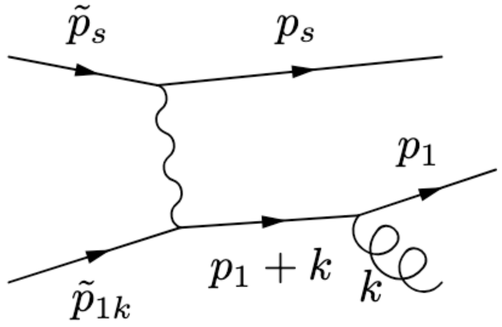
$$P_{g\to gg}(z) = C_A \left[\frac{z}{1-z} + \frac{1-z}{z} + z(1-z)\right]$$



$$C_F = \frac{4}{3}, \qquad C_A = 3, \qquad T_R = \frac{1}{2}.$$

Comments

- There are soft divergences in z=1 and z=0
- Gluons radiate the most
- All soft divergences are gluon related
- Need additional parton to conserve momentum → introduce spectator (Catani-Seymour)



Sudakov form factor

What is the ***probability of no emission***?

$$\mathcal{P}_{\text{no-branching}}(t_i) = 1 - \mathcal{P}_{\text{branching}}(t_i) = 1 - \frac{\delta t}{t_i} \frac{\alpha_s}{2\pi} \int \mathrm{d}z \, \hat{P}(z)$$

Sudakov form factor

What is the ***probability of no emission***?

$$\mathcal{P}_{\text{no-branching}}(t_i) = 1 - \mathcal{P}_{\text{branching}}(t_i) = 1 - \frac{\delta t}{t_i}\,\frac{\alpha_s}{2\pi}\int \mathrm{d}z\,\hat{P}(z)$$

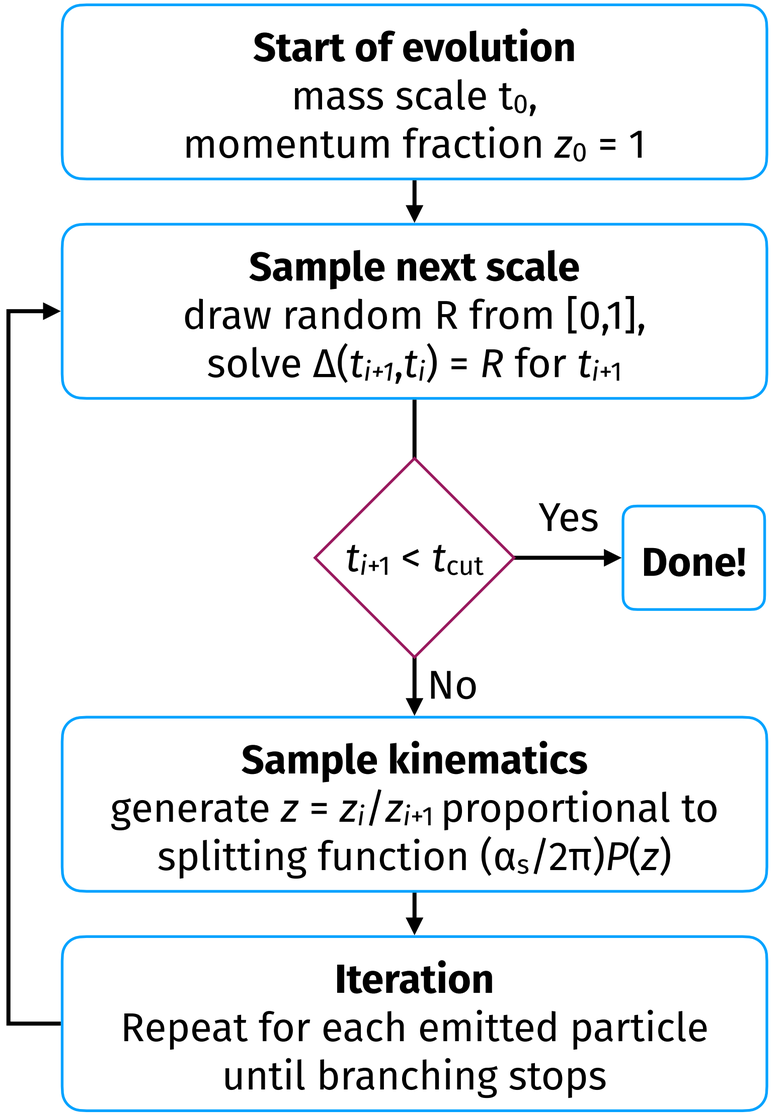
Probability of no emission between two scales

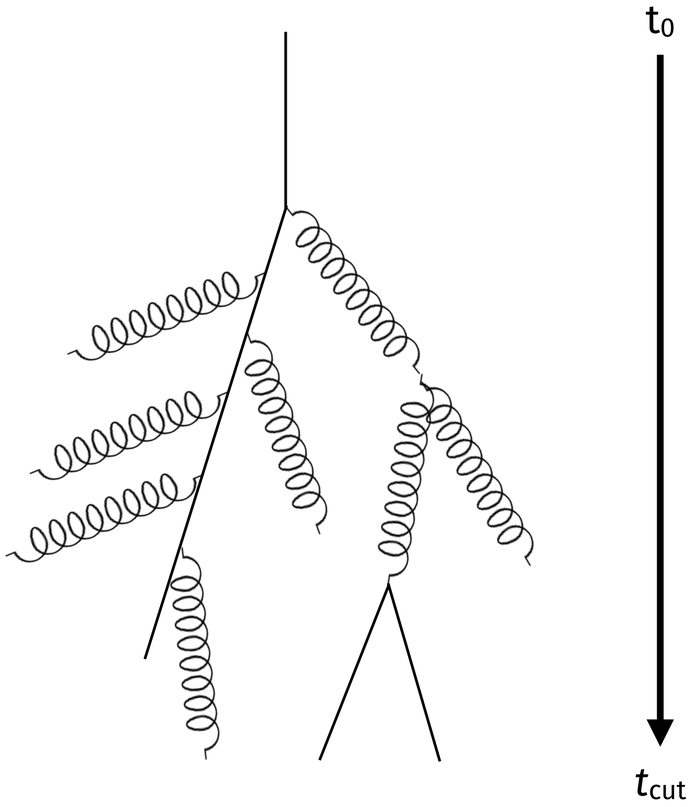
$$\begin{aligned}
\Delta(Q^2,t) \equiv \mathcal{P}_{\text{no-branching}}(Q^2,t) &= \lim_{N\to\infty}\prod_{i=0}^{N}\left[1 - \frac{\delta t}{t_i}\,\frac{\alpha_s}{2\pi}\int \mathrm{d}z\,\hat{P}(z)\right] \\
&\simeq \lim_{N\to\infty}\exp\left[-\sum_{i=0}^{N}\frac{\delta t}{t_i}\,\frac{\alpha_s}{2\pi}\int \mathrm{d}z\,\hat{P}(z)\right] \\
&\simeq \exp\left[-\int_t^{Q^2}\frac{\mathrm{d}t'}{t'}\int \mathrm{d}z\,\frac{\alpha_s}{2\pi}\,\hat{P}(z)\right] \equiv \exp\left[-\int_t^{Q^2}\mathrm{d}p(t')\right].
\end{aligned}$$

→ Sudakov form factor $\Delta(Q^2,t)$ with property

$$\Delta(t_1,t_3) = \Delta(t_1,t_2)\Delta(t_2,t_3)$$

Parton shower algorithm





Remarks

- Lot of freedom in the choice of evolution parameter
  → examples: virtuality, transverse momentum, splitting angle…
  → equivalent in collinear limit, but may differ in soft limit

- So far, we only looked at final-state showers
  → need to account for parton densities in initial-state shower

$$|\mathcal{M}_{n+1}|^2 \, \mathrm{d}\Phi_{n+1} \simeq |\mathcal{M}_n|^2 \, \mathrm{d}\Phi_n \times \frac{\mathrm{d}t}{t} \, \mathrm{d}z \, \frac{\mathrm{d}\phi}{2\pi} \, \frac{\alpha_s}{2\pi} \, P_{a\to bc}(z) \, \frac{f_a(x/z,t)}{f_b(x,t)}$$

Summary

- Splittings described by ***universal splitting kernels***
- Sudakov form factor: non-branching probability between two scales

$$\Delta(Q^2, t) \simeq \exp\left[-\int_t^{Q^2} \frac{\mathrm{d}t'}{t'} \int \mathrm{d}z \, \frac{\alpha_s}{2\pi} \hat{P}(z)\right] \equiv \exp\left[-\int_t^{Q^2} \mathrm{d}p(t')\right]$$

- Modeled as ***chain of emissions***

  → each interaction has its own scale for α_s

  → various choices for evolution parameter

Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. **Multijet merging**

Matrix Element vs Parton Shower

Matrix element

1. Fixed order calculation
2. Computationally expensive
3. Limited number of particles
4. Valid when partons are ***hard and well separated***
5. Quantum interference correct
6. Needed for multi-jet description

Parton shower

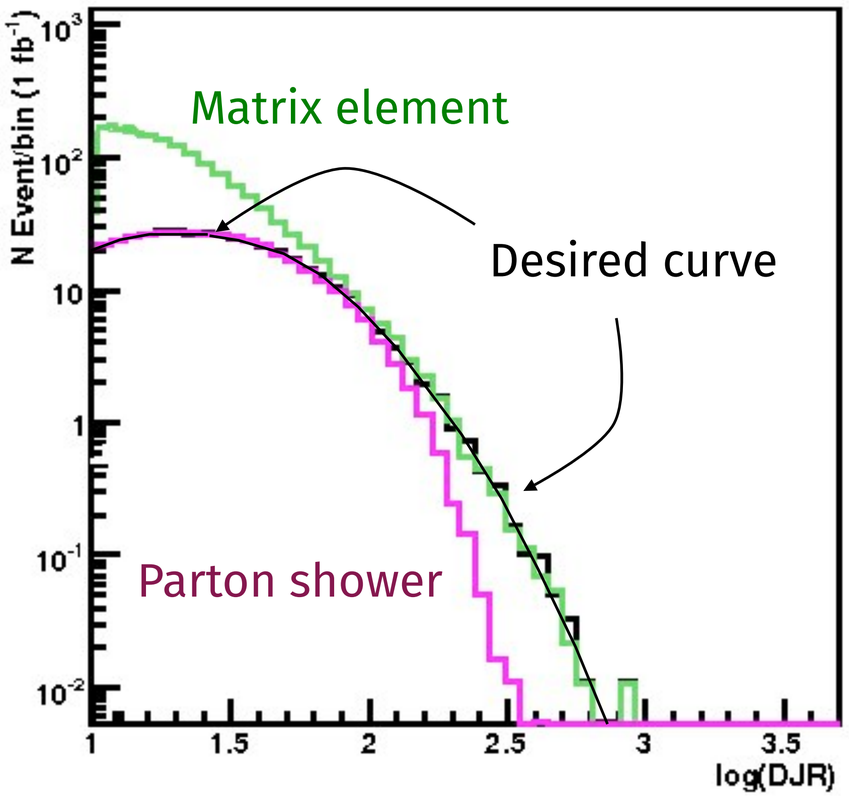
1. Resums logs to all orders
2. Computationally cheap
3. No limit on particle multiplicity
4. Valid when partons are ***collinear and/or soft***
5. Interference via angular ordering
6. Needed for hadronization

Approaches are complementary: merge them!

Challenge: avoid double counting, ensure smooth distributions

Goals

- Regularization of matrix element divergence
- Correction of the parton shower for hard momenta
- Smooth jet distributions

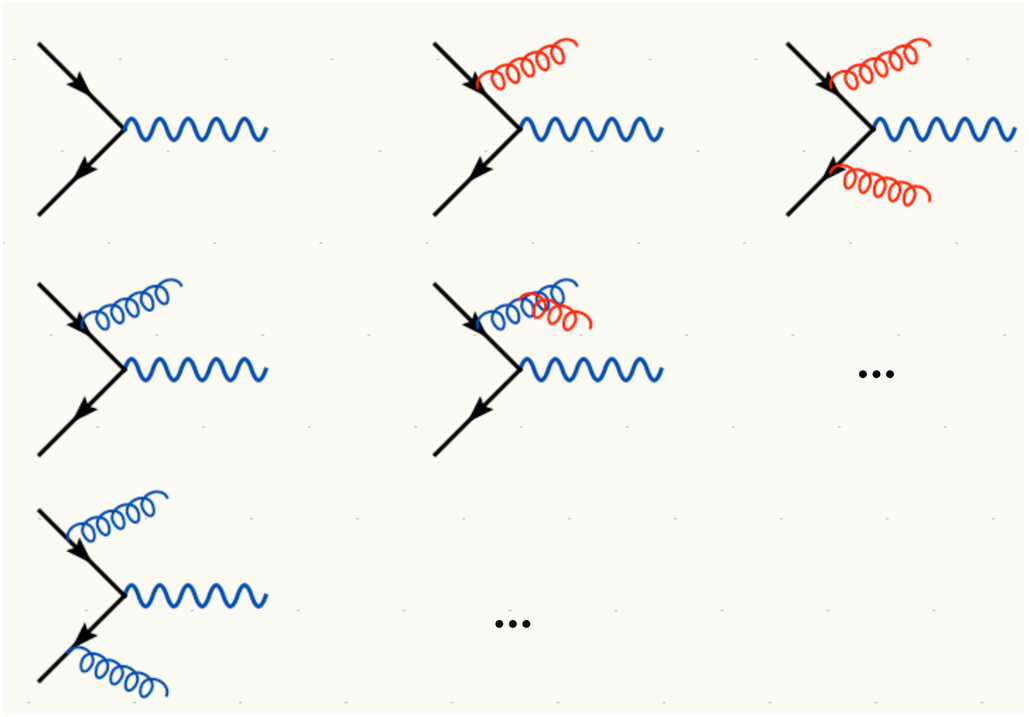


Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Double counting

PS jets →

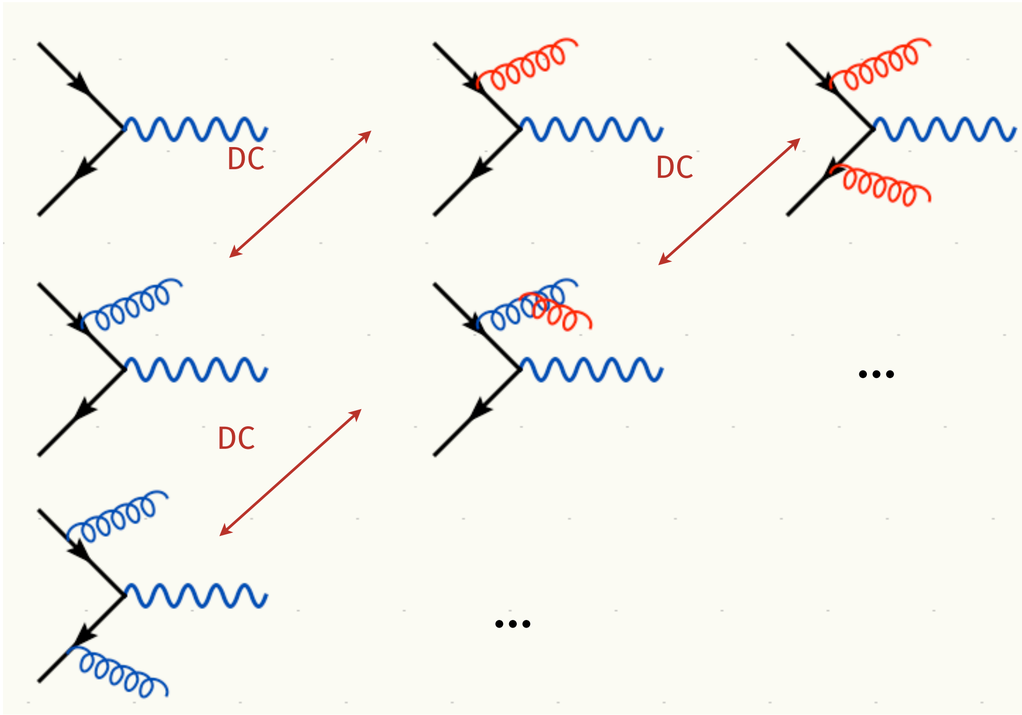
ME jets ↓



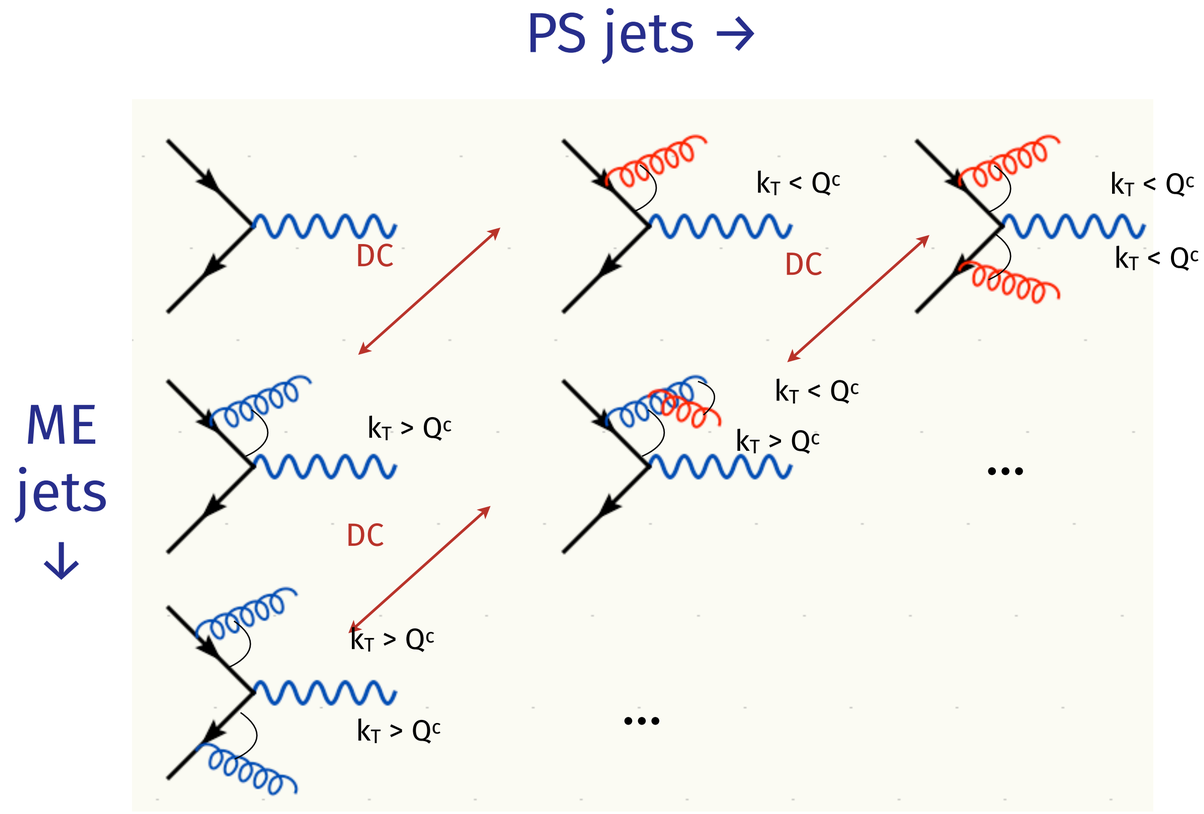
Double counting

PS jets →

ME jets ↓



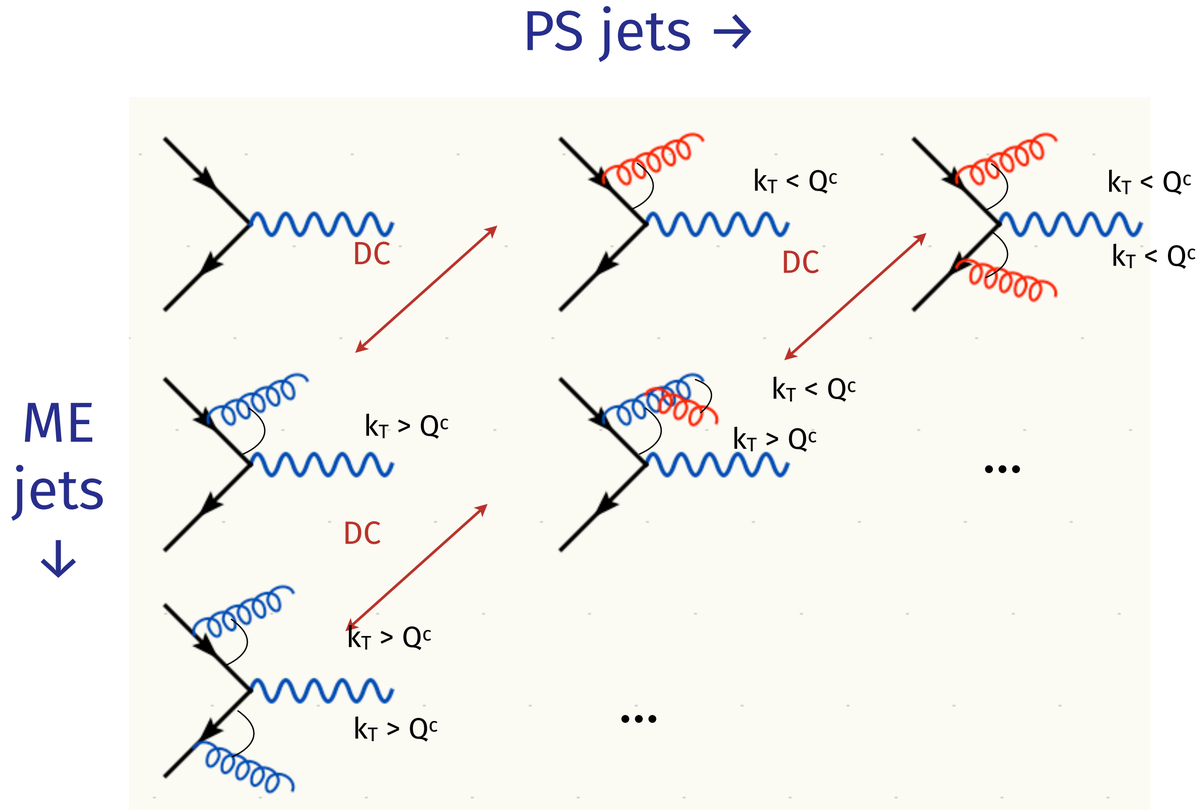
Double counting



Solution to double counting: phase space cut between the two.

PS below cutoff, ME above cutoff.

Double counting

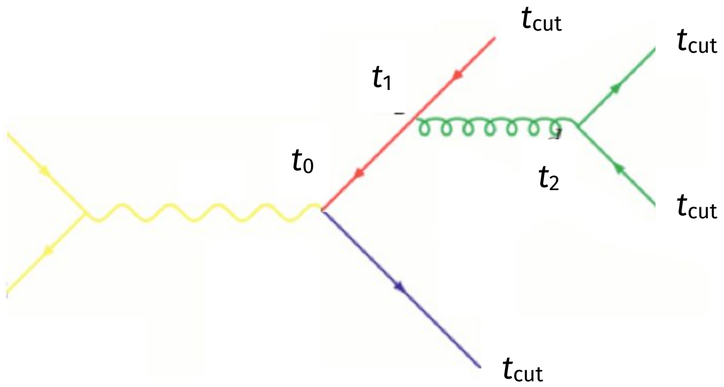


Solution to double counting: phase space cut between the two.

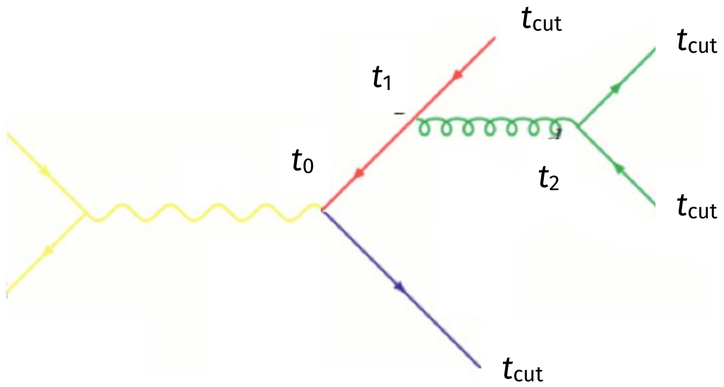
PS below cutoff, ME above cutoff.

Next problem: how can we ensure smooth distribution close to cut-off?

Merging ME and PS

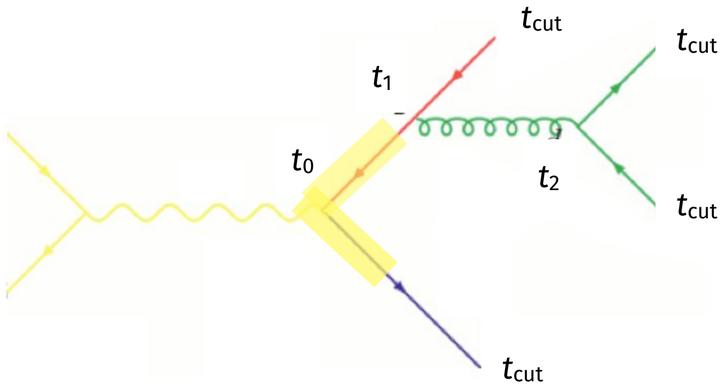


Merging ME and PS



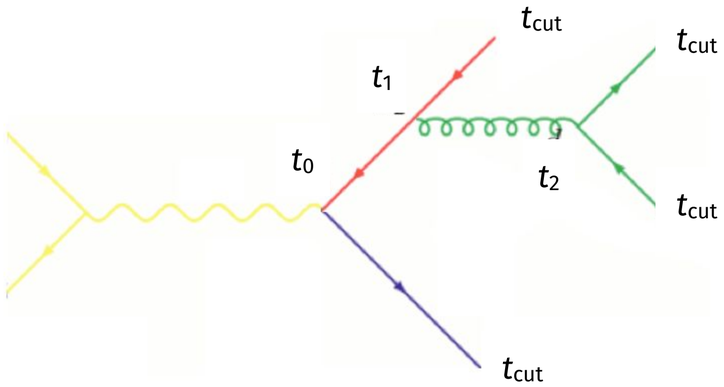
$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

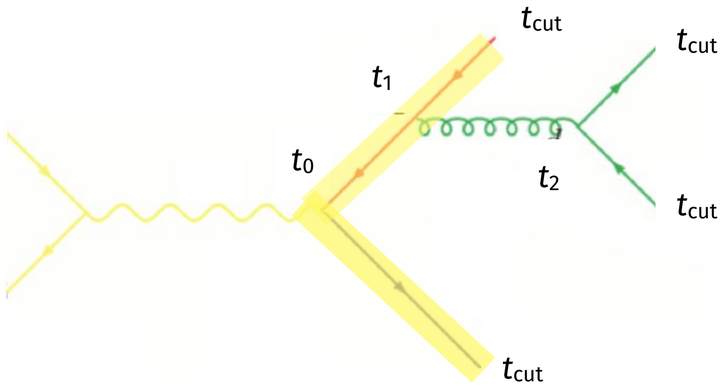
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1) (\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

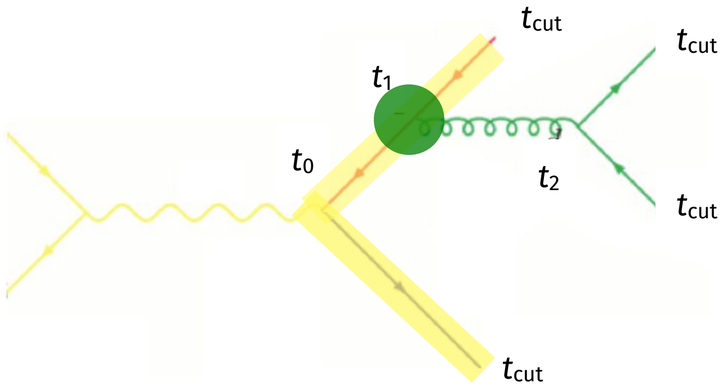
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1)(\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

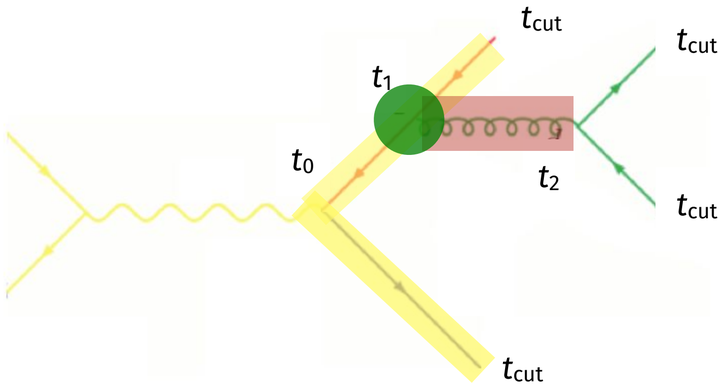
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1) (\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

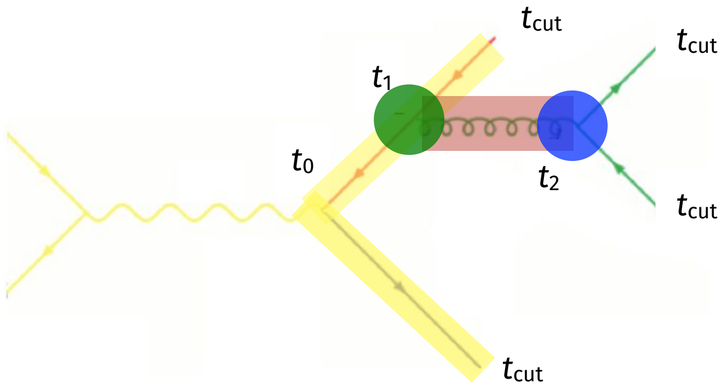
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1) (\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

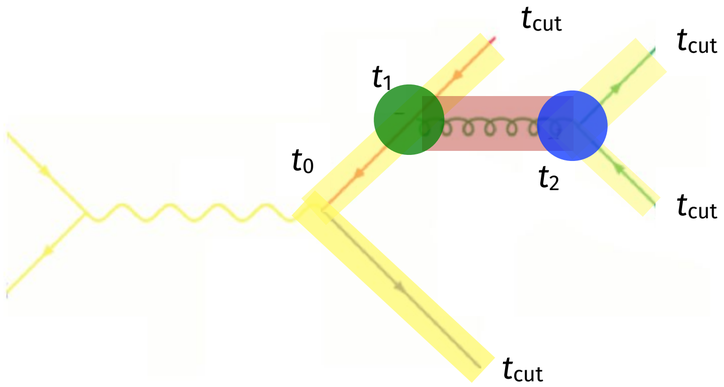
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1) (\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

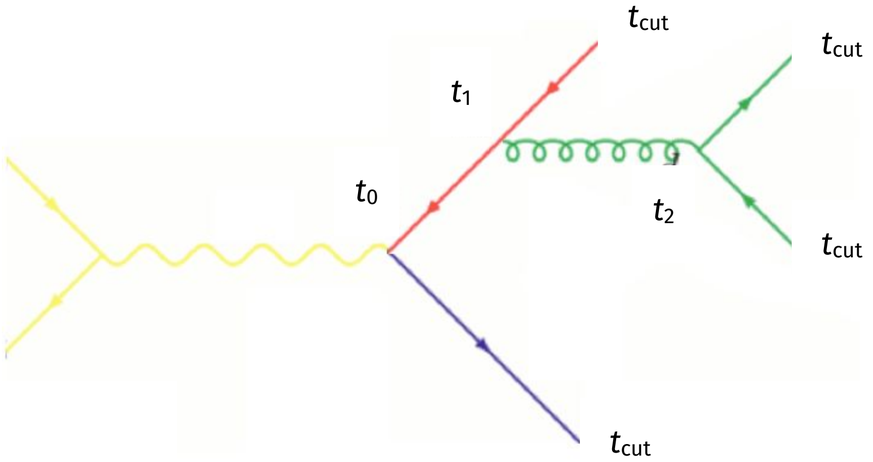
Merging ME and PS



$$(\Delta_q(t_1, t_0))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z)$$

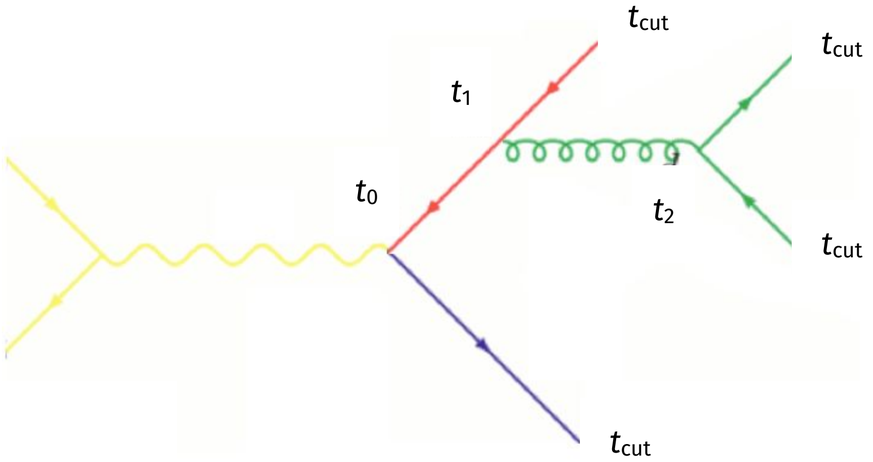
$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1) (\Delta_q(t_{\text{cut}}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

Merging ME + PS



$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1)(\Delta_q(\text{cut}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

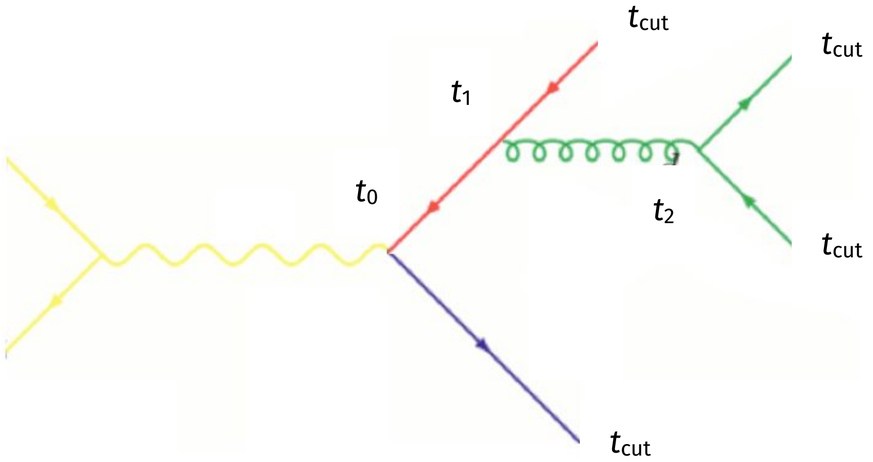
Merging ME + PS



$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1)(\Delta_q(\text{cut}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

Corresponds to matrix element, but
with α_s evaluated at scale of each splitting

Merging ME + PS



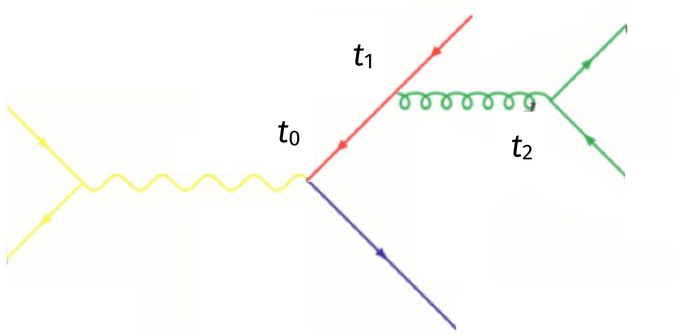
$$(\Delta_q(t_{\rm cut}, t_0))^2 \Delta_g(t_2, t_1)(\Delta_q({\rm cut}, t_2))^2 \frac{\alpha_s(t_1)}{2\pi} P_{gq}(z) \frac{\alpha_s(t_2)}{2\pi} P_{qg}(z')$$

Sudakov suppression due to not allowing additional radiation above the scale t_{cut}

Corresponds to matrix element, but with α_s evaluated at scale of each splitting

Merging ME + PS

Steps for equivalent treatment of matrix element



$$|\mathcal{M}|^2(\hat{s}, p_3, p_4, ...)$$

1. Cluster the event using some clustering algorithm
   → corresponding “parton shower history”
2. Reweight α_s in each clustering vertex with the clustering scale

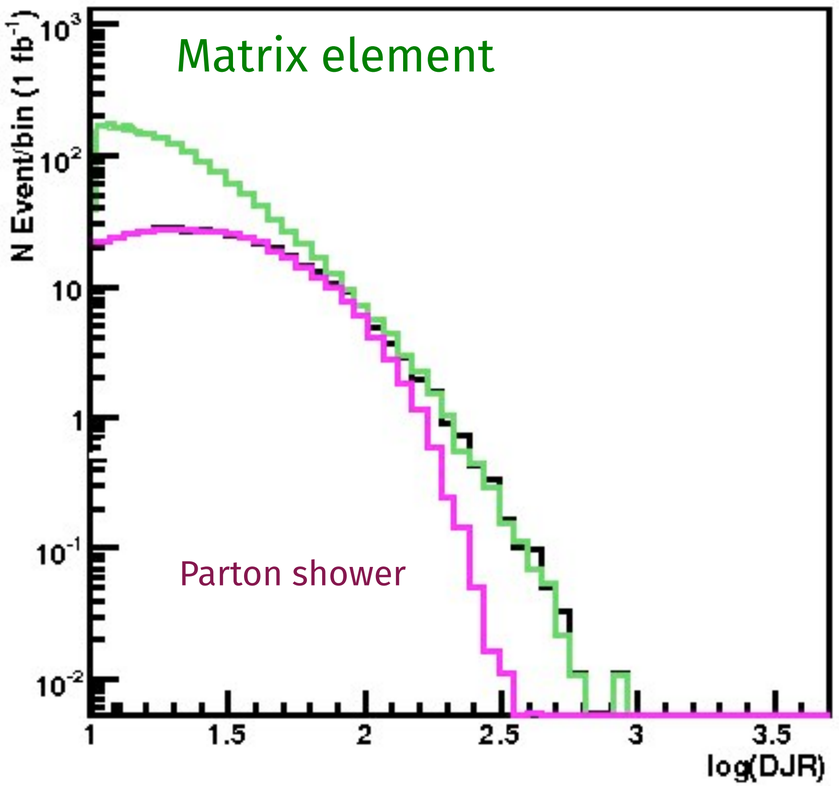
$$|\mathcal{M}|^2 \to |\mathcal{M}|^2 \frac{\alpha_s(t_1)}{\alpha_s(t_0)} \frac{\alpha_s(t_2)}{\alpha_s(t_0)}$$

3. Use some algorithm to apply the equivalent Sudakov suppression

$$(\Delta_q(t_{\text{cut}}, t_0))^2 \Delta_g(t_2, t_1)(\Delta_q(_{\text{cut}}, t_2))^2$$

Back to the “matching goals”

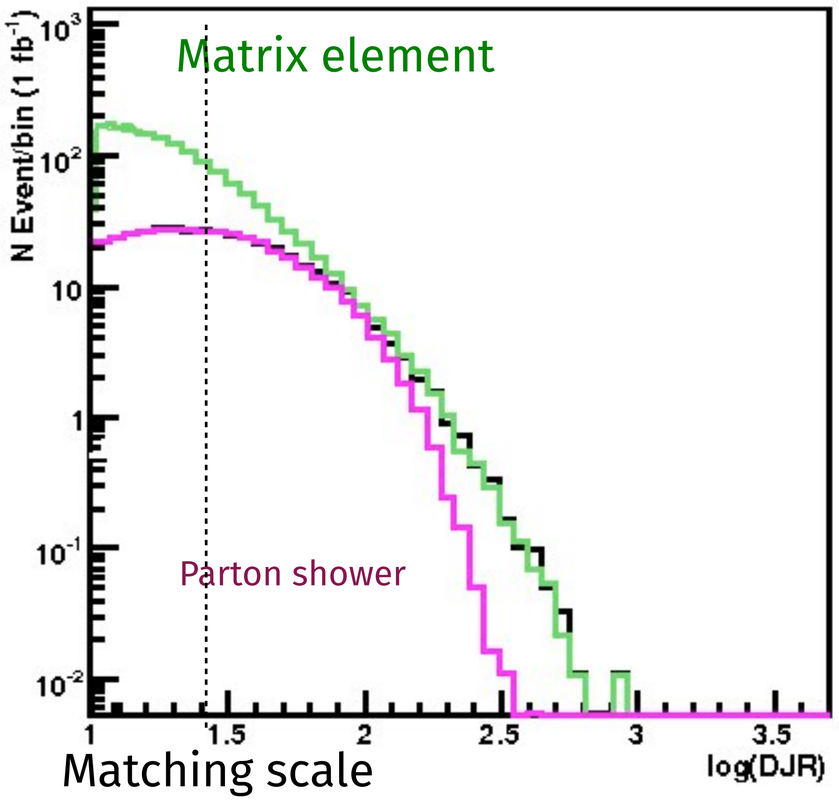
- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Back to the “matching goals”

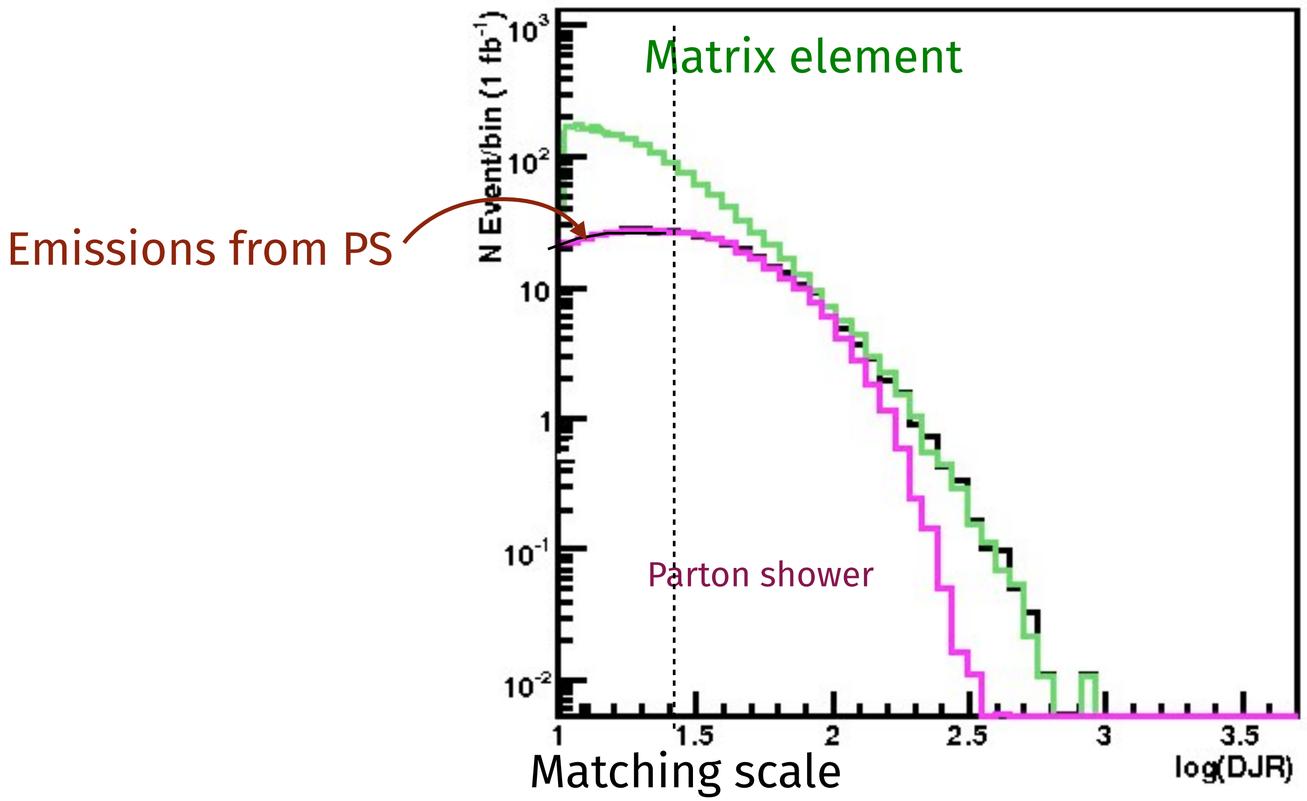
- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Back to the “matching goals”

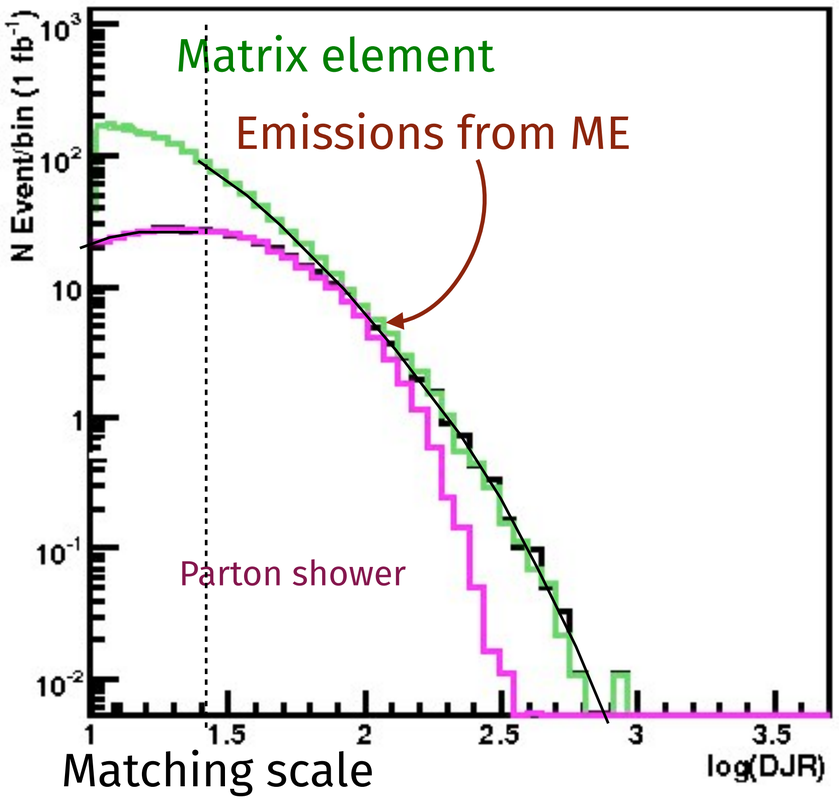
- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Back to the “matching goals”

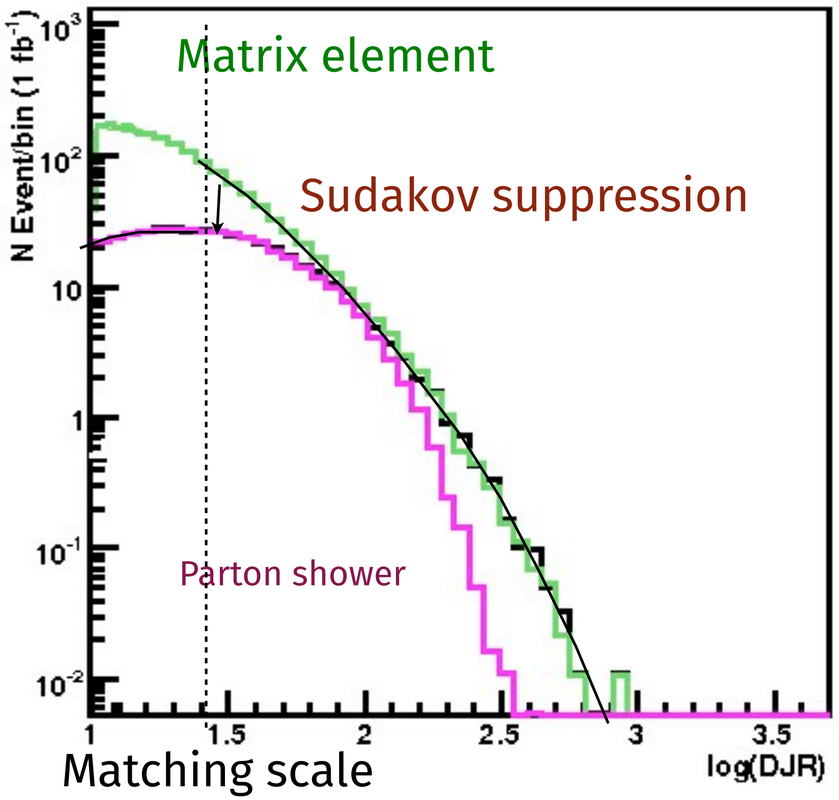
- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Back to the “matching goals”

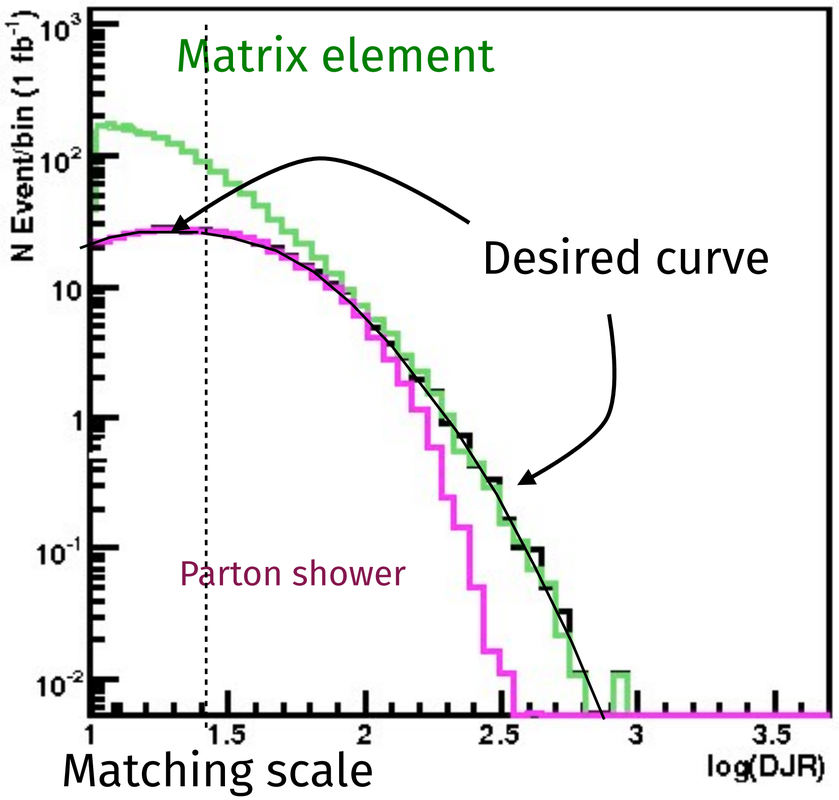
- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Back to the “matching goals”

- Regularization of matrix element divergence
- Correction of the parton shower for large momenta
- Smooth jet distributions



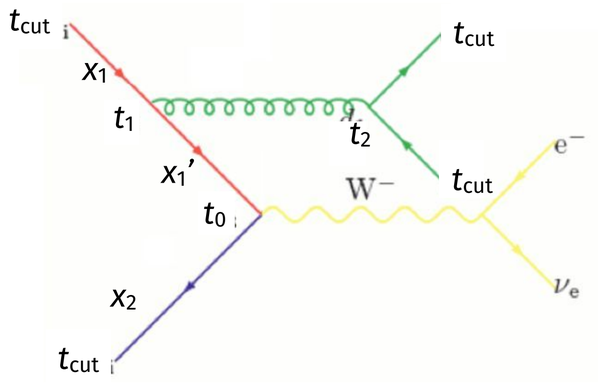
Example:
2nd QCD radiation jet
in top pair production at the LHC,
using MadGraph + Pythia

Matching for ISR

To include ISR:

have to **add PDF reweighting factor** like in parton shower

$$(\Delta_{Iq}(t_{\rm cut},t_0))^2\Delta_g(t_2,t_1)(\Delta_q(t_{\rm cut},t_2))^2\frac{\alpha_s(t_1)}{2\pi}\frac{P_{gq}(z)}{z}\frac{f_q(x_1,t_1)}{f_q(x_1',t_1)}\frac{\alpha_s(t_2)}{2\pi}P_{qg}(z')$$
$$\times\hat{\sigma}_{q\bar{q}\to e\nu}(\hat{s},...)f_q(x_1',t_0)f_{\bar{q}}(x_2,t_0)$$



Matching for ISR

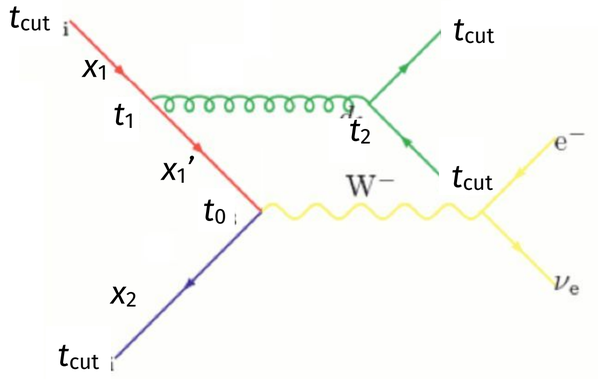
To include ISR:

have to **add PDF reweighting factor** like in parton shower

$$(\Delta_{Iq}(t_{\rm cut},t_0))^2\Delta_g(t_2,t_1)(\Delta_q(t_{\rm cut},t_2))^2\frac{\alpha_s(t_1)}{2\pi}\frac{P_{gq}(z)}{z}\frac{f_q(x_1,t_1)}{f_q(x_1',t_1)}\frac{\alpha_s(t_2)}{2\pi}P_{qg}(z')$$

$$\times\hat{\sigma}_{q\bar{q}\to e\nu}(\hat{s},...)f_q(x_1',t_0)f_{\bar{q}}(x_2,t_0)$$

ME with α_s evaluated at the scale of each splitting



Matching for ISR

To include ISR:

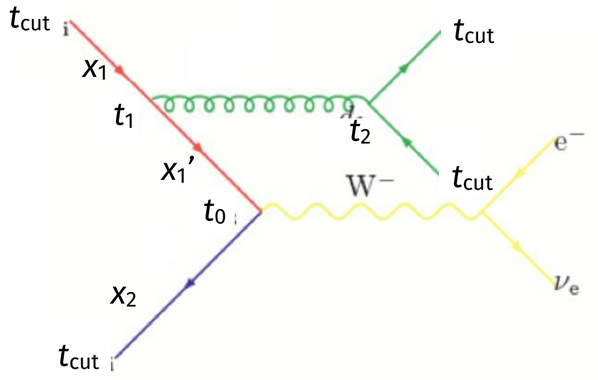
have to **add PDF reweighting factor** like in parton shower

$$(\Delta_{Iq}(t_{\rm cut},t_0))^2\Delta_g(t_2,t_1)(\Delta_q(t_{\rm cut},t_2))^2\frac{\alpha_s(t_1)}{2\pi}\frac{P_{gq}(z)}{z}\frac{f_q(x_1,t_1)}{f_q(x_1',t_1)}\frac{\alpha_s(t_2)}{2\pi}P_{qg}(z')$$

$$\times\hat{\sigma}_{q\bar{q}\to e\nu}(\hat{s},...)f_q(x_1',t_0)f_{\bar{q}}(x_2,t_0)$$

ME with α_s evaluated at the scale of each splitting

PDF reweighting



Matching for ISR

To include ISR:

have to **add PDF reweighting factor** like in parton shower

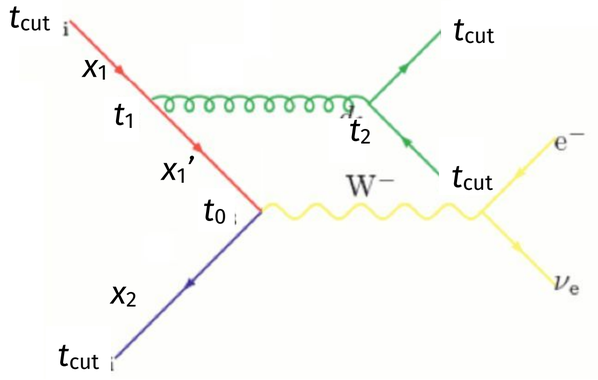
$$(\Delta_{Iq}(t_{\rm cut},t_0))^2\Delta_g(t_2,t_1)(\Delta_q(t_{\rm cut},t_2))^2\frac{\alpha_s(t_1)}{2\pi}\frac{P_{gq}(z)}{z}\frac{f_q(x_1,t_1)}{f_q(x_1',t_1)}\frac{\alpha_s(t_2)}{2\pi}P_{qg}(z')$$

$$\times\hat{\sigma}_{q\bar{q}\to e\nu}(\hat{s},...)f_q(x_1',t_0)f_{\bar{q}}(x_2,t_0)$$

ME with α_s evaluated at the scale of each splitting

PDF reweighting

Sudakov suppression due to non-branching above scale $t_{\rm cut}$



Matching schemes

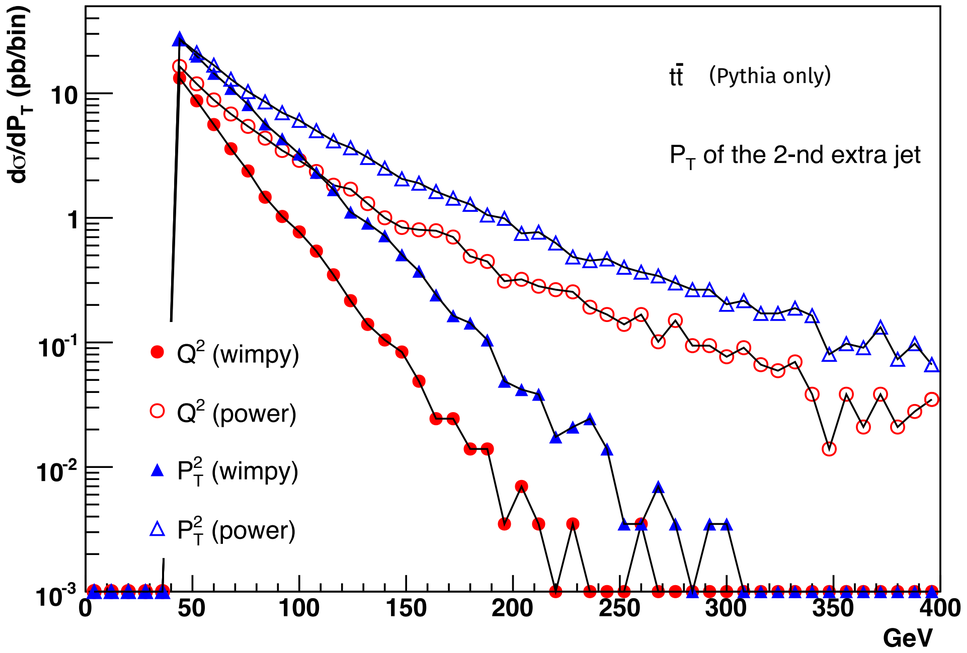
- Have a number of choices in the matching procedure
  1. The clustering scheme used to determine the parton shower history of the ME event
  2. What to use for the scale of hard emission
  3. How to divide the phase space between parton showers and matrix elements

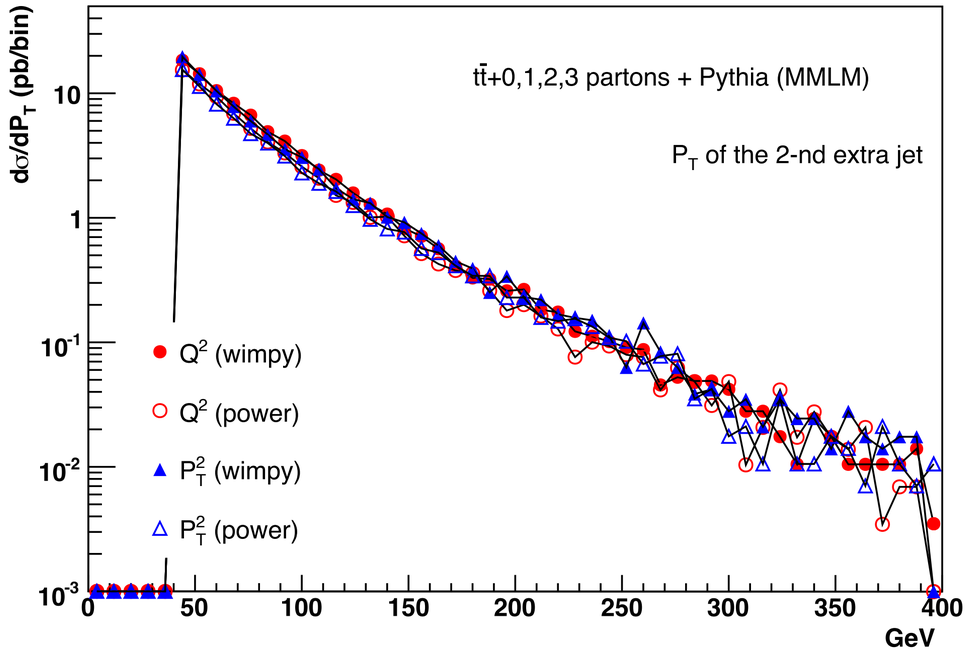
- Three general schemes available in the literature
  1. CKKW scheme [Catani,Krauss,Kuhn,Webber 2001; Krauss 2002]
  2. Lönnblad scheme (or CKKW-L) [Lönnblad 2002]
  3. MLM scheme [Mangano unpublished 2002; Mangano et al. 2007]

PS alone vs matched samples

In soft-collinear approximation of parton shower: parameters are used to tune the result
→ Large variation in results
→ small prediction power

Matched sample: behavior at high p_T is dominated by the matrix element
→ differences become irrelevant



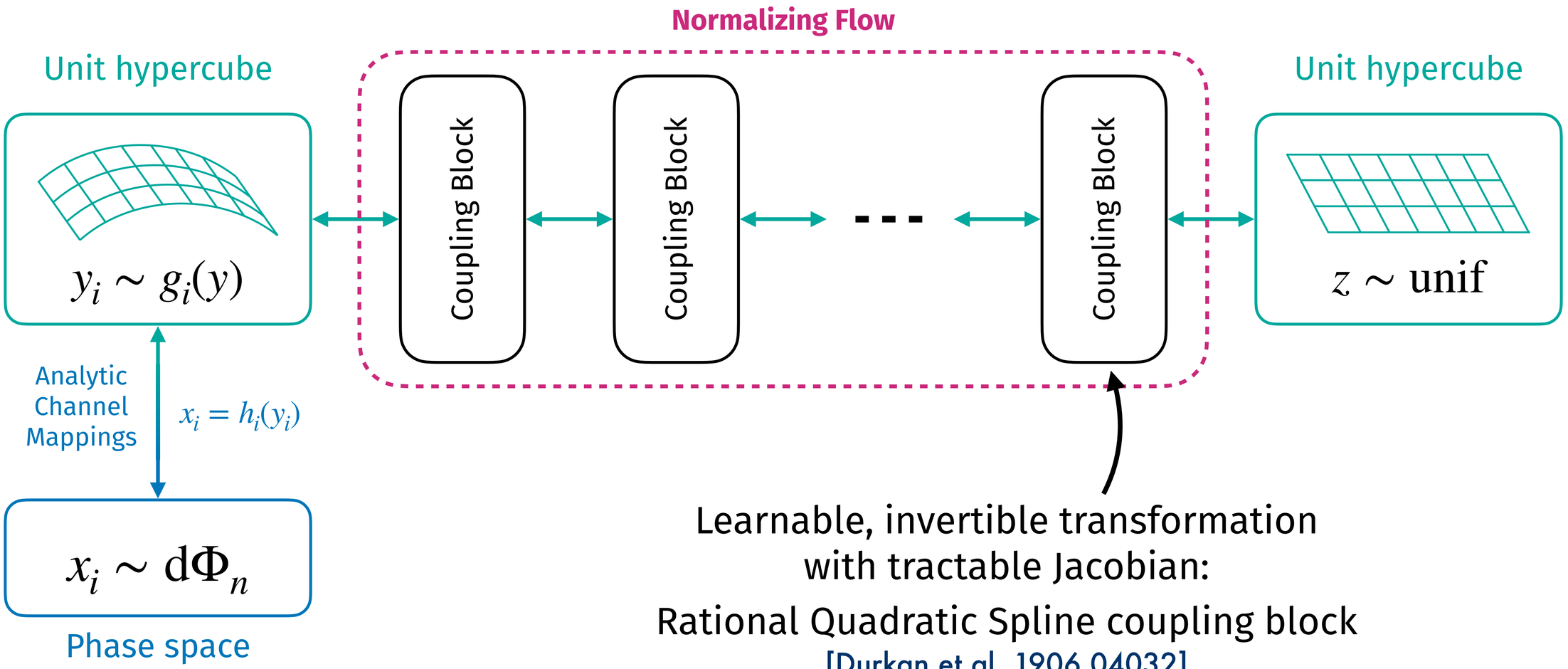


Summary

- Matrix element and parton shower are complimentary
  → matrix element: valid when partons are ***hard and well separated***
  → parton shower: valid when partons are ***collinear and/or soft***
- need to ***avoid double counting***
  → introduce ***cut-off: PS below, ME above***
- Matching: ensure smooth distribution at cut-off
  → obtain shower history by clustering
  → reweight α_s and apply Sudakov suppression

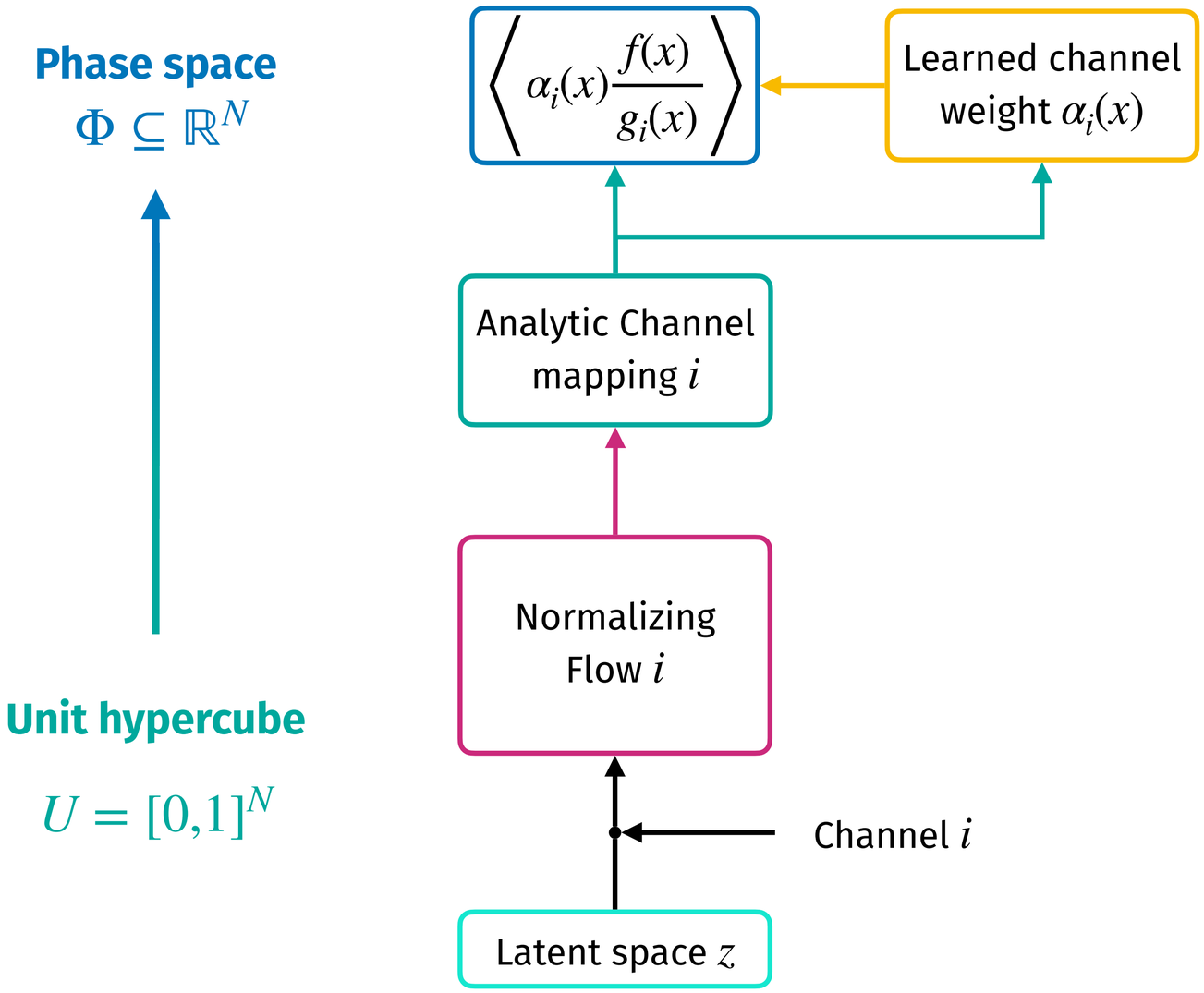
Appendix

Neural importance sampling



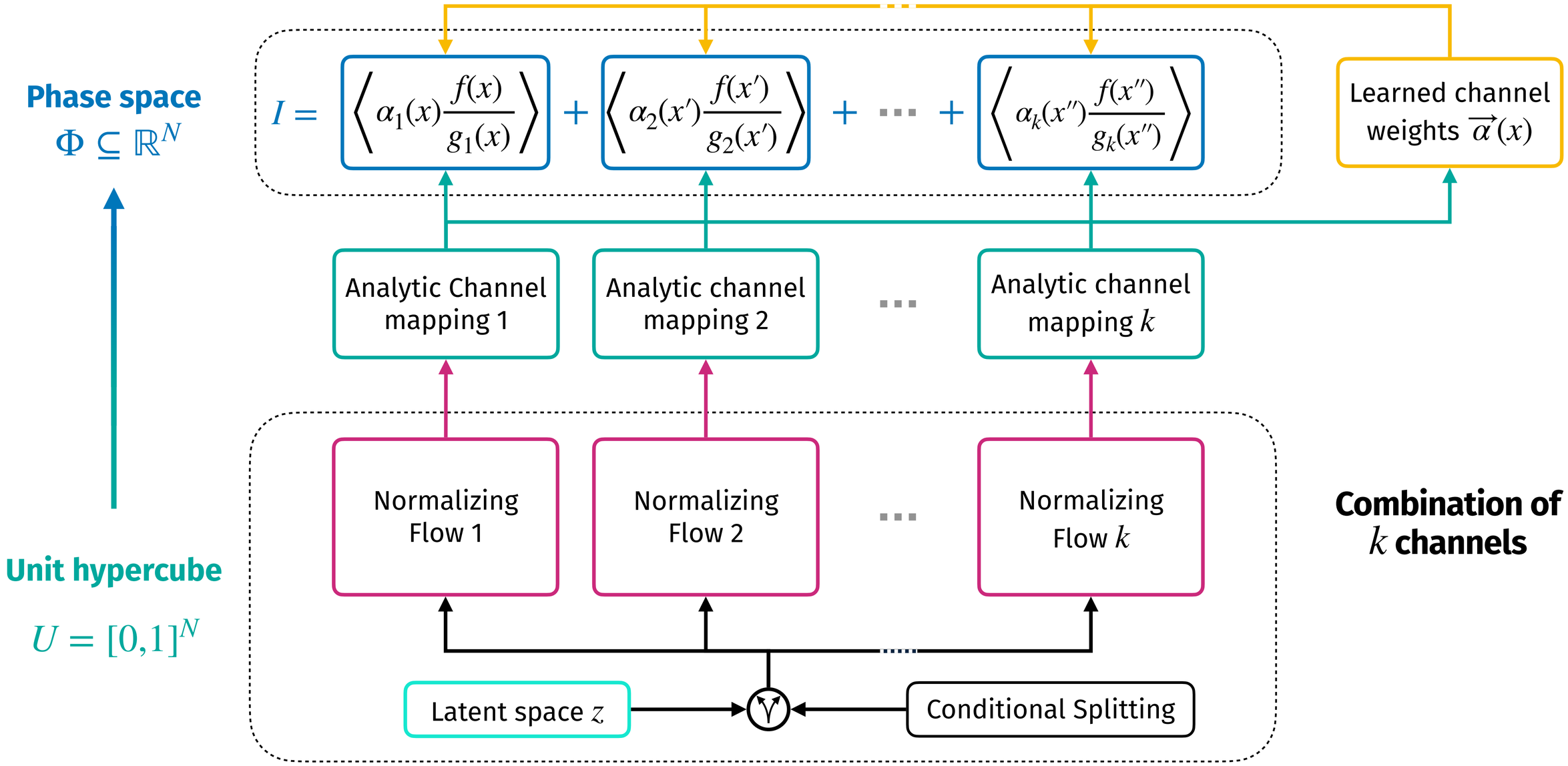
[Gao *et al,* 2001.05486+2001.10028] [Bothmann *et al,* 2001.05478+2506.18987]
[Winterhalder *et al,* 2112.09145] [Deutschmann *et al,* 2401.09069] [Janssen *et al.,* 2505.13608]

MadNIS: Neural importance sampling

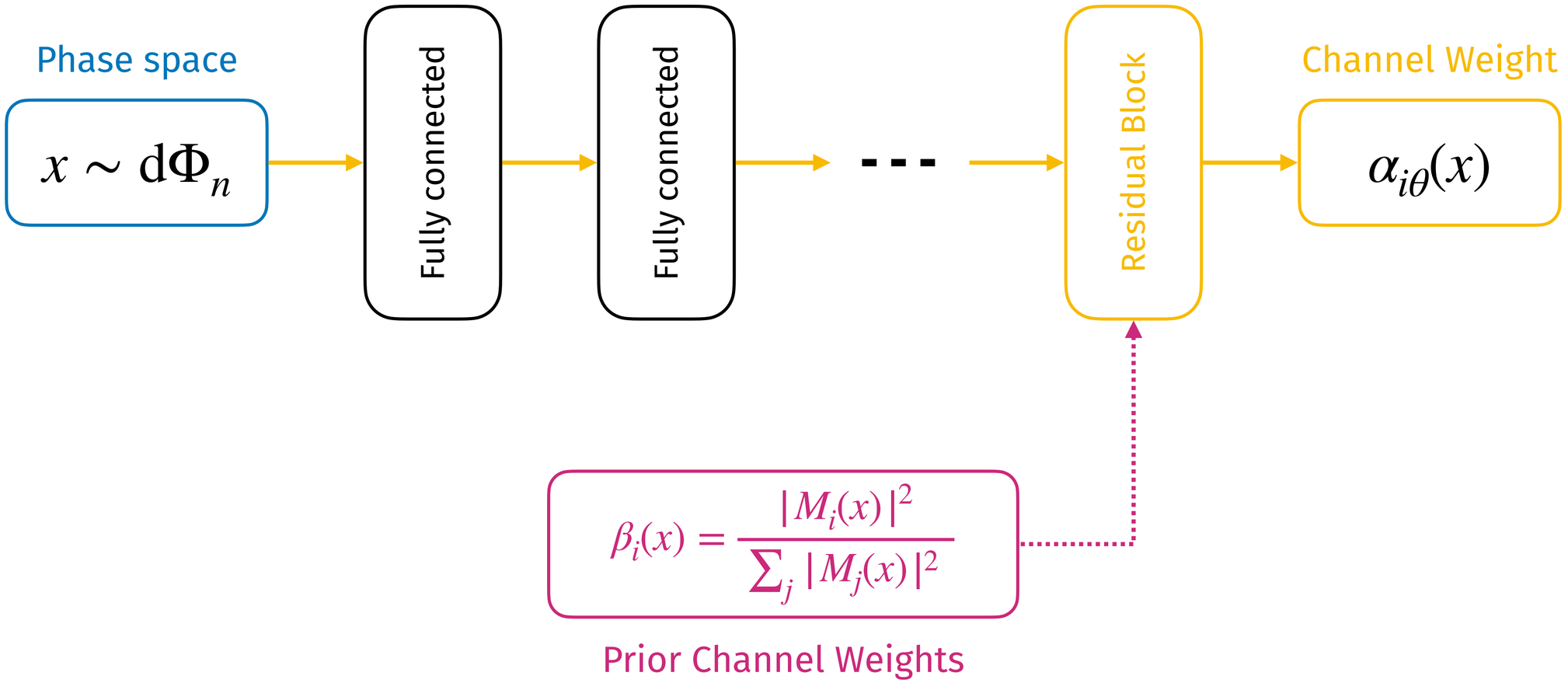


Single channel i

MadNIS: Neural importance sampling



Neural channel weights



Neural channel weights

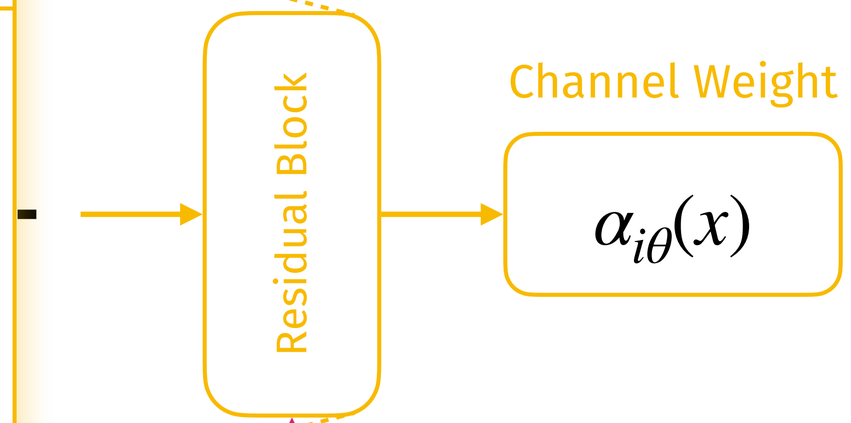
Residual Block

Add prior

$$\alpha_{i\theta} = \beta_i(x) \exp \Delta_{i\theta}(x)$$

Normalization

$$\alpha_{i\theta}(x) \rightarrow \hat{\alpha}_{i\theta}(x) = \frac{\beta_i(x) \exp \Delta_{i\theta}(x)}{\sum_j \beta_j(x) \exp \Delta_{i\theta}(x)}$$



$$\beta_i(x) = \frac{|M_i(x)|^2}{\sum_j |M_j(x)|^2}$$

Prior Channel Weights

Loss function

Training objective:
Minimize total variance

$$\sigma_{\text{tot}}^2 = N \sum_i \frac{\sigma_i^2}{N_i} \qquad \text{with}$$

$$\sigma_i^2 = \text{Var}\left(\alpha_i(x) \frac{f(x)}{g_i(x)} \right)_{x \sim g_i(x)}$$

Optimal MC weights depend on N_i

↓

assume choice of N_i during training:
use stratified sampling

$$N_i = N \frac{\sigma_i}{\sum_k \sigma_k}$$

MadNIS loss function

$$\mathscr{L} = \sigma_{\text{tot}}^2 = \sum_{i,k} \sigma_i \, \sigma_k$$