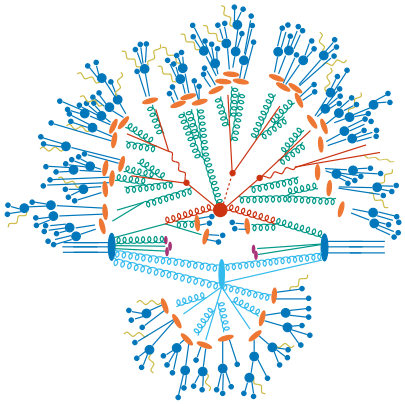
# Too good to go: Upcycling Phase-Space Points for Multijet Processes

arXiv:2608.27104

Konrad Helms
with Timo Janßen and Steffen Schumann
15th September 2026

# Introduction

- LHC $\xrightarrow{\text{upgrade}}$ HL-LHC finished by $\sim 2030$
- current MC samples statistically limited compared to HL-LHC data
- $\mathcal{M}_{ab\to X_N}$ expensive:
  - in particular high-multiplicity final states
  - need stacks: $X + 4j, X + 5j, \ldots$
  - dedicated samplers for each process' phase space
- in this talk: reduce $\mathrm{d}\Phi_N$-sampler training costs by reducing No. of $\mathcal{M}_{ab\to X_N}$ evaluations, while still being unbiased



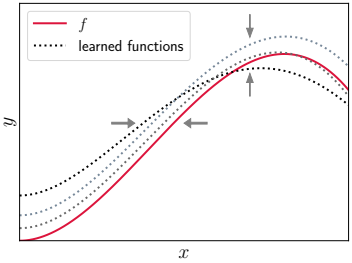
$$\hat{\sigma}_{ab\to X_N}(\hat{s}) = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_N |\mathcal{M}_{ab\to X_N}|^2$$

# Neural Importance Sampling

- importance sampling:

$$\int_{[0,1]^d} \mathrm{d}x f(x) = \int_{[0,1]^d} \mathrm{d}x \frac{f(x)}{g(x)} g(x)$$
$$\approx \frac{1}{N} \sum_{i=1}^{N} \underbrace{\frac{f(x_i)}{g(x_i)}}_{\equiv w_i} \text{ with } x_i \sim g$$

- $g$ approximates $f$ well:
  $\iff f/g \simeq \text{const.} \iff$ spread of weights minimal
- *neural* importance sampling: [Müller et al., arXiv:1808.03856]
  $g(\cdot) = g(\cdot, \boldsymbol{\theta})$ with learnable parameters $\boldsymbol{\theta}$
- for us: $g$ is Normalizing Flow




- in HEP, see e.g.:
  arXiv:2001.05478,
  arXiv:2604.03511,
  arXiv:2401.09069

slide on intuition

# Continuous Normalizing Flows & Flow Matching

[Lipman et al. arXiv:2210.02747]

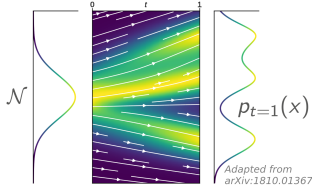
- $\left.\begin{array}{rcl} \boldsymbol{x}_0 & \sim & \mathcal{N} \\ \boldsymbol{x}_1 & \sim & \text{data} \end{array}\right\}$ linear interpolation: $\boldsymbol{x}(t) = t\boldsymbol{x}_1 + (1-t)\boldsymbol{x}_0$ with $t \in [0,1]$
- $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{x}_1 - \boldsymbol{x}_0$
- model target during training: $\boldsymbol{v}_\theta(\boldsymbol{x}, t) = \boldsymbol{x}_1 - \boldsymbol{x}_0$

sampling - solving an ODE:

$$\Rightarrow \int_{t=0}^{t=1} \mathrm{d}\tau \frac{\mathrm{d}}{\mathrm{d}\tau}\boldsymbol{x}(\tau) = \int_0^1 \mathrm{d}\tau\, \boldsymbol{v}_\theta(\boldsymbol{x}, \tau) \Rightarrow \boldsymbol{x}(t=1) = \boldsymbol{x}(t=0) + \int_0^1 \mathrm{d}\tau\, \boldsymbol{v}_\theta(\boldsymbol{x}, \tau)$$

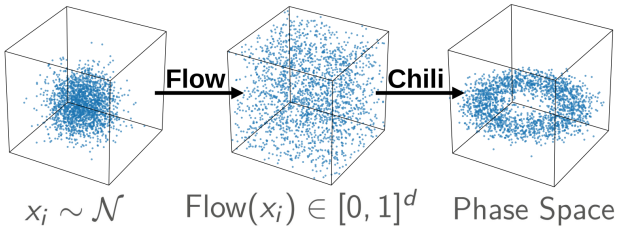
evaluating densities - solving an ODE:

- $\frac{\mathrm{d}\log(p_t(\boldsymbol{x}))}{\mathrm{d}t} = -\nabla \cdot \boldsymbol{v}_\theta(\boldsymbol{x}, t)$
  $\Rightarrow \log(p_{t=1}(\boldsymbol{x}_1)) = \log(\mathcal{N}(\boldsymbol{x}_0)) - \int_0^1 \mathrm{d}\tau\, \nabla \cdot \boldsymbol{v}_\theta(\boldsymbol{x}, \tau)$



# Workflow: A Chained Mapping

- expert knowledge about integrand, implemented through phase-space mapping
- dimensionality: $d = 3N - 4 + 2$
- Flow: $[0,1]^d \to [0,1]^d$
- CHILI [Bothmann et al., arXiv:2302.10449]: $[0,1]^d \to \mathrm{d}x_a \mathrm{d}x_b \mathrm{d}\Phi_N$



# CHILI Phase-Space Parameterization

- want to integrate
  $\sigma_{pp\to X_N} = \sum_{a,b} \int_0^1 \mathrm{d}x_a \int_0^1 \mathrm{d}x_b \times \text{PDFs} \times \hat{\sigma}_{ab\to X_N}(x_a, x_b, \mu_R, \mu_F)$ with
  $\hat{\sigma}_{ab\to X_N}(\hat{s}) = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_N |\mathcal{M}_{ab\to X_N}|^2$
- with Lorentz-invariant phase-space element $\mathrm{d}\Phi_N$, parameterized in $\mathrm{d}p_\perp^2, \mathrm{d}y, \mathrm{d}\varphi$
- 4-momentum conservation implemented through combination with light-cone momentum fractions $\mathrm{d}x_a$ and $\mathrm{d}x_b$
- coordinate nesting, not literal inclusion:

$$\mathrm{d}x_a\,\mathrm{d}x_b\,\mathrm{d}\Phi_{N+1} = \left[\mathrm{d}x_a\,\mathrm{d}x_b\,\mathrm{d}\Phi_N\right] \times \mathrm{d}\Phi_1$$
with
$$z_{N+1} = \left(z_N, p_{\perp,1}^2, y_1, \varphi_1\right)$$

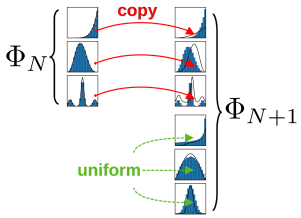
# The New Training Method

- in this talk: case study $d\bar{d} \to e^+e^- + n \cdot g$, in paper also $gg \to t\bar{t} + n \cdot g$
- evaluation of $\mathcal{M}$ expensive $\Rightarrow$ generating training data is expensive
- we propose new training strategy that
  - requires significantly less $\mathcal{M}$ evaluations to reach benchmark performance
  - yields models of superior performance compared to benchmark trained on similar amount of training data
- the idea:
  1. use $N$-particle phase space sampler to generate $\mathrm{d}\Phi_N$ points (no $\mathcal{M}$ needed)
  2. augment well-sampled $\mathrm{d}\Phi_N$ dataset to $\mathrm{d}\Phi_{N+1}$
     $\Rightarrow$ $\mathrm{d}\Phi_{N+1}$ dataset now better matches integrand than sampled uniformly
  3. train $(N+1)$-particle phase space sampler
  4. repeat for next higher particle multiplicity

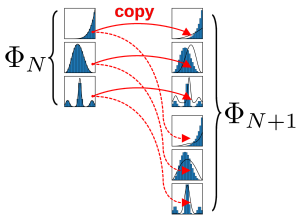
# Phase-Space Augmentation: Dimensions

- assuming efficient $\mathrm{d}\Phi_N$ sampler exists:

$$\mathbf{z}_N = \begin{bmatrix} z_N^{(1)} \\ z_N^{(2)} \\ \vdots \\ z_N^{(d_N)} \end{bmatrix}_{\Phi_N} \xrightarrow{\text{copy}} \begin{bmatrix} z_N^{(1)} \\ z_N^{(2)} \\ \vdots \\ z_N^{(d_N)} \\ * \\ \vdots \\ * \end{bmatrix}_{\Phi_{N+1}}$$




- uniform sampling of extra dimensions




- fill extra dimensions with previous particle's phase space points

# Phase-Space Augmentation: Conditions

- distributions conditional on the helicity configurations of the external particles
- extra gluon in $X + n \cdot g \to X + (n+1)g$ has helicity state $+1$ or $-1$

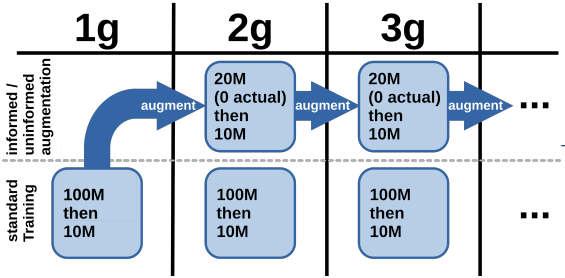
  $\Longrightarrow$ number of helicity combinations on $\mathrm{d}\Phi_{N+1}$ doubles w.r.t. $\mathrm{d}\Phi_N$
- condition augmentation:

$$\left[\boldsymbol{z}_N, c\right] \longrightarrow \begin{cases} \left[(\boldsymbol{z}_N, p^2_{\perp,n+1}, y_{n+1}, \varphi_{n+1}),\ c\right] \\ \left[(\boldsymbol{z}_N, p^2_{\perp,n+1}, y_{n+1}, \varphi_{n+1}),\ c + \text{No. helicity cond. on } \mathrm{d}\Phi_N\right] \end{cases}$$

- e.g.: from 32 conditions on $\mathrm{d}\Phi_N$ we create 32+32 conditions on $\mathrm{d}\Phi_{N+1}$, etc.

# Build-up of the Training Stacks

- for multijet processes: can build a stack of consecutive augmented trainings
- training speed-ups accumulate over time




1. for $X + 1g$: sample $\sim$ 100M uniform points $\rightarrow$ evaluate $\mathcal{M}$ to calculate MC weight $w$
2. (iteratively) train $X + 1g$ phase-space sampler with phase-space points and weights

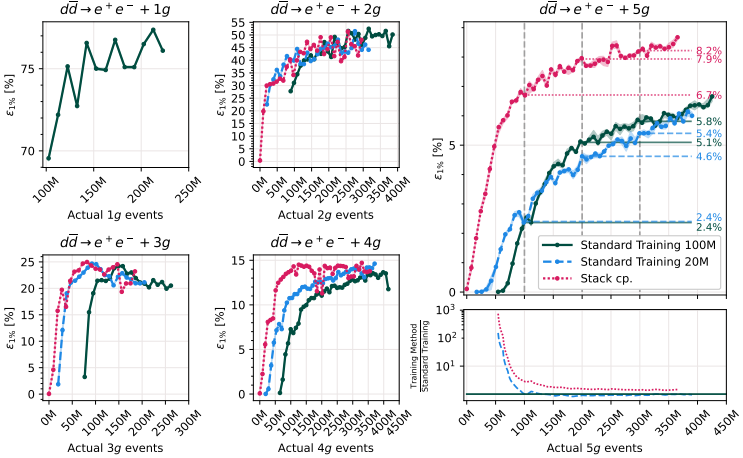
$\rightarrow$ iterative training: train model $\rightarrow$ generate phase-space points $\rightarrow$ evaluate $\mathcal{M} \rightarrow$ repeat

3. augment $X + 1g \rightarrow X + 2g$
4. repeat

# Results #1

$d\bar{d} \to e^+e^- + n \cdot g$

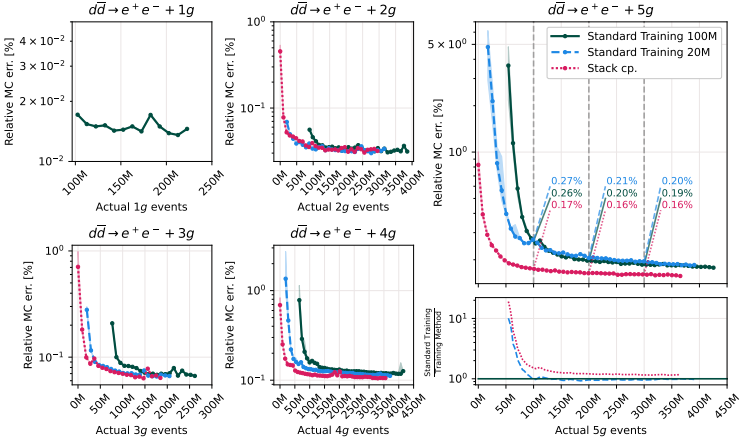
- augmentation vs. standard training
- performance metric: unweighting efficiency $\epsilon = \langle w \rangle / \max(w)$ (larger is better)



# Results #2

$d\bar{d} \to e^+e^- + n \cdot g$

- augmentation vs. standard training
- performance metric: relative MC integration error $\sigma_{\text{error}}/\sigma$ (smaller is better)



# Conclusion and Outlook

- augmentation can realize significant resource savings
  - up to 80% in $\mathcal{M}$ evaluation cost
  - gain largest for highest dimensional problem: $\Rightarrow$ fraction of well-initialized phase-space dimensions for $n$ gluons: $\frac{4+3n}{7+3n}$
  - only used to initialize proposal distribution $\rightarrow$ exact event weighting preserves unbiased MC estimates
- augmentation applicable also in other contexts
  - agnostic to sampler, can be used for VEGAS, coupling flows etc.
  - other areas:
    $\sigma^{\text{NLO}} = \int \mathrm{d}\Phi_N [B(\Phi_N) + V(\Phi_N)] + \int \mathrm{d}\Phi_{N+1} R(\Phi_{N+1})$
- check out: arXiv:2608.27104

# Thank you for your attention

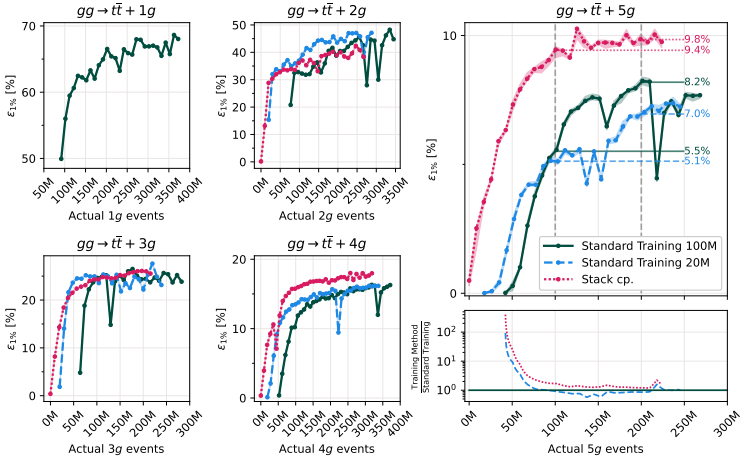
Georg-August-Universität Göttingen
E-Mail: konrad.helms@theorie.physik.uni-goettingen.de

# Backup

# Results #3

$gg \to t\bar{t} + n \cdot g$

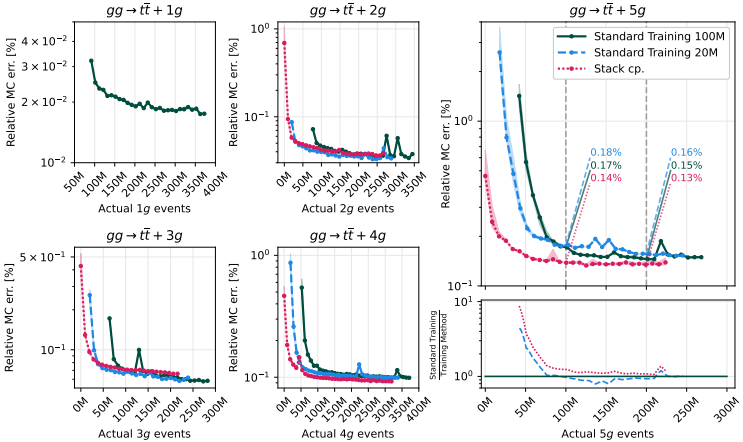
- augmentation vs. standard training
- performance metric: unweighting efficiency $\epsilon = \langle w \rangle / \max(w)$ (larger is better)



# Results #4

$gg \to t\bar{t} + n \cdot g$

- augmentation vs. standard training
- performance metric: relative MC integration error $\sigma_{\text{error}}/\sigma$ (smaller is better)



# Results #5

## $d\bar{d} \to e^+e^- + n \cdot g$ and $gg \to t\bar{t} + n \cdot g$

0.1% unweighting efficiency and Kish effective sample size: $\frac{N_{\text{eff.}}}{N_{\text{events}}} = \frac{\langle w \rangle^2}{\langle w^2 \rangle}$

(both: larger is better)