ÖAW AUSTRIAN ACADEMY OF SCIENCES

MARIETTA BLAU INSTITUTE FOR PARTICLE PHYSICS

Efficient Event Generation for High-Multiplicity LHC Processes: An End-to-End GPU Workflow with Normalizing Flows

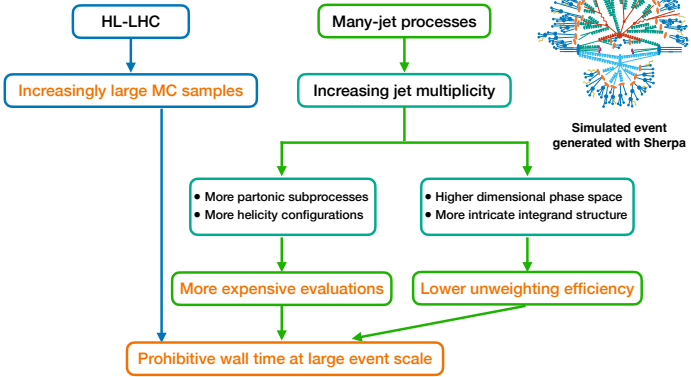
Daohan Wang

Marietta Blau Institute for Particle Physics (MBI), Austrian Academy of Sciences (OeAW)

Based on 2608.21338

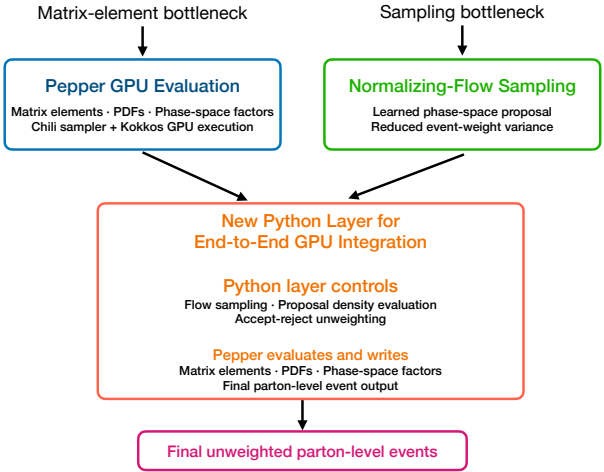
with Enrico Bothmann, Joshua Isaacson, Claudius Krause,
Carla J. López-Zurita and Maximilian Spannring

Many-Jet Generation Faces Two Coupled Bottlenecks



more events × higher cost per unweighted event → prohibitive wall time

A New Python Layer Integrates Pepper and Learned Sampling



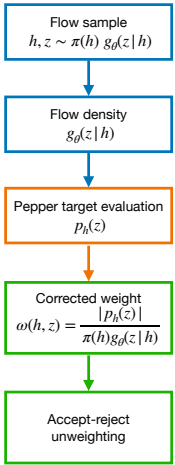
Fast Density Evaluation Matters for Event Production

Every proposed event needs both
a **sample** and a **proposal density**

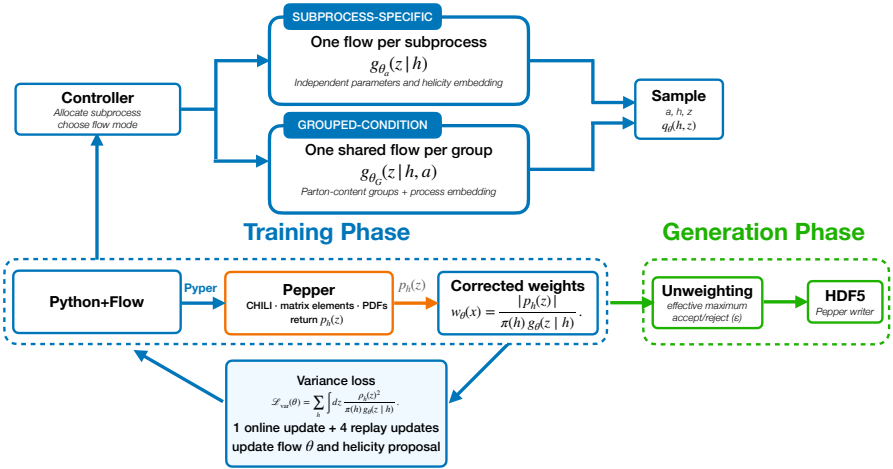
Coupling flows provide **fast sampling**
and **exact density evaluation**

The goal is not only proposal quality,
but **end-to-end wall time** for complete processes.

The flow changes only the proposal,
exact corrected weights keep the physics **unbiased**.

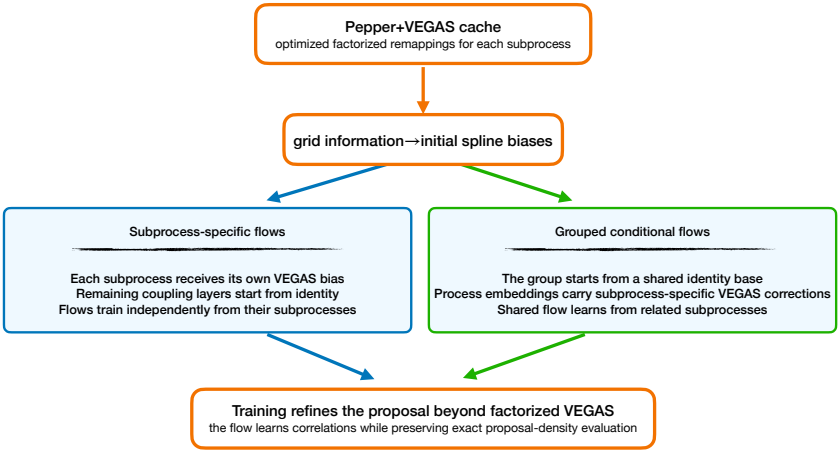


The End-to-End GPU Workflow

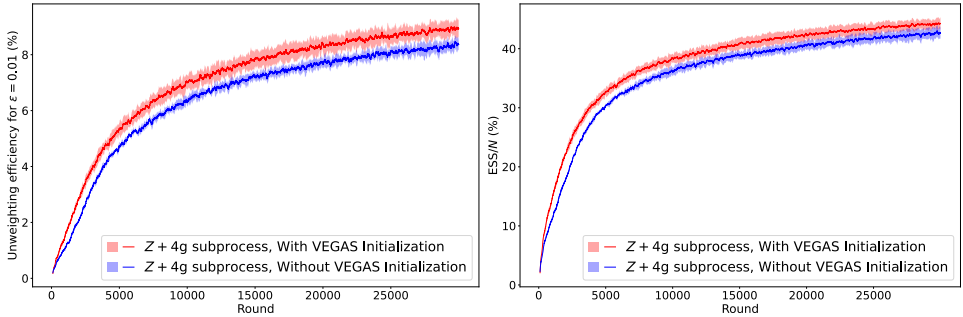


VEGAS Initialization Gives the Flow a Structured Start

Convert the optimized Pepper+VEGAS grid into initial flow biases, then refine by training



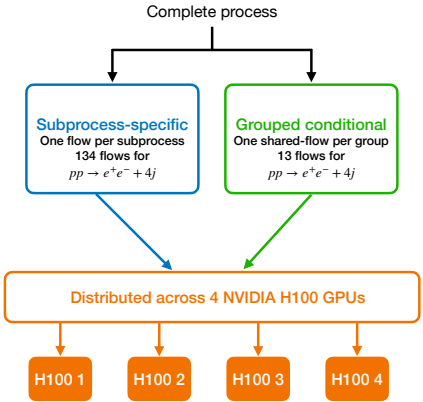
VEGAS Initialization Raises Sampling Efficiency Throughout Training



- Representative $pp \to e^+e^- + 4g$ subprocess, averaged over five trainings
- With VEGAS initialization, both acceptance and ESS/N are higher across the full training range.
- The shaded bands show run-to-run variation; the gain is a systematic upward shift

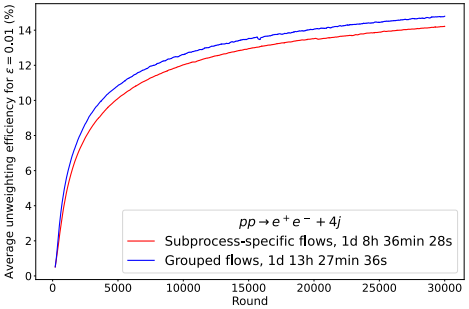
Scaling Flows to Complete Many-Jet Processes

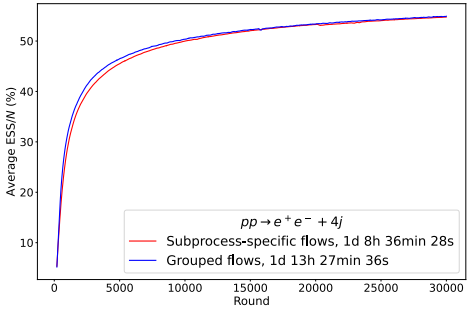
Process	Subprocesses	Groups
$pp \to e^+e^- + 4j$	134	13
$pp \to e^+e^- + 5j$	168	15
$pp \to t\bar{t} + 4j$	20	10
$pp \to 4j$	20	10
$pp \to 5j$	24	11



Disjoint model sets · No gradient synchronization · vmap parallelization · torch.compile

Training Improves the Proposal Quality

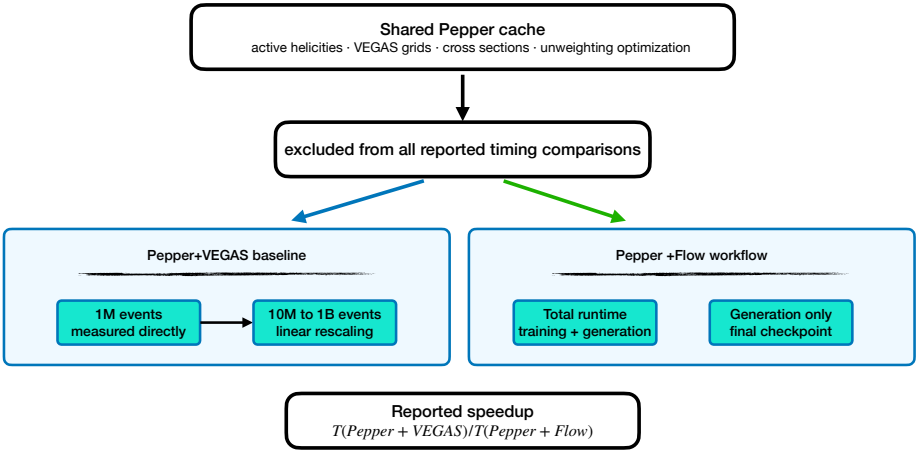




Same per-subprocess training statistics for both setups
Higher acceptance and ESS/N mean lower corrected-weight fluctuations

Speedups Compare Complete Event-Generation Workflows

The same Pepper cache is used for both paths; cache construction is common preprocessing



Checkpoint Selection Balances Training and Generation

$$\widehat{T}_{\text{total}}(c; N_{\text{evt}}) = T_{\text{train}}(c) + \frac{N_{\text{evt}}}{10^6}\, T_{\text{gen}}(c; 10^6)$$

Choose checkpoint c minimizing the estimated total runtime.

1. Save
checkpoints every
50 rounds

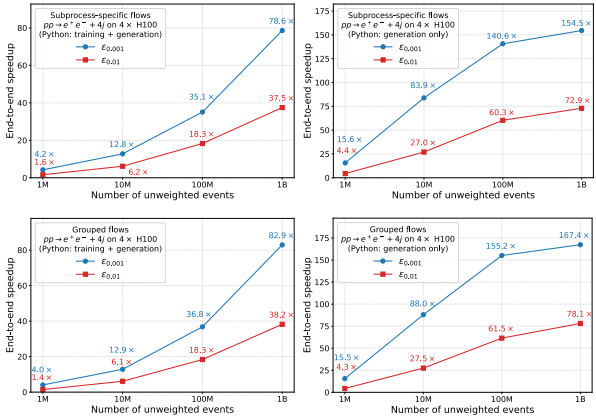
2. Estimate
training cost +
projected generation cost

3. Select
later checkpoints when
generation gains pay off

Flow setup	ε	Selected checkpoint round			
		10^6	10^7	10^8	10^9
Subprocess-specific	10^{-3}	100	400	900	**3600**
Subprocess-specific	10^{-2}	50	200	600	**1500**
Grouped conditional	10^{-3}	100	250	750	**3150**
Grouped conditional	10^{-2}	50	150	450	**1250**

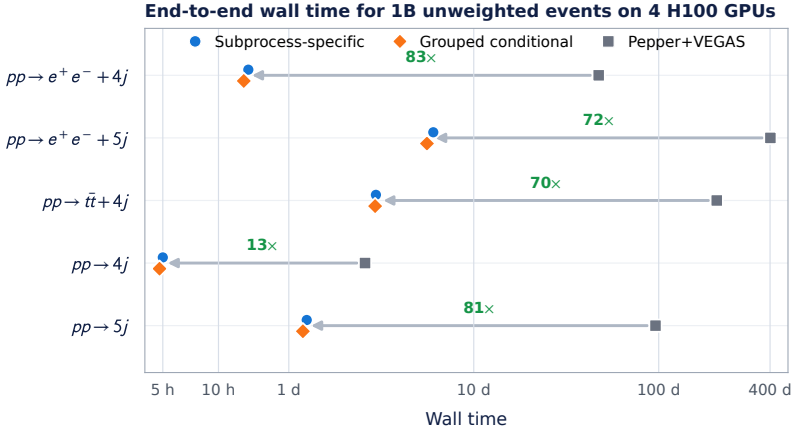
Best proposal quality and best end-to-end runtime are not the same target.

Learned Proposals Pay Off at Large Event Scale



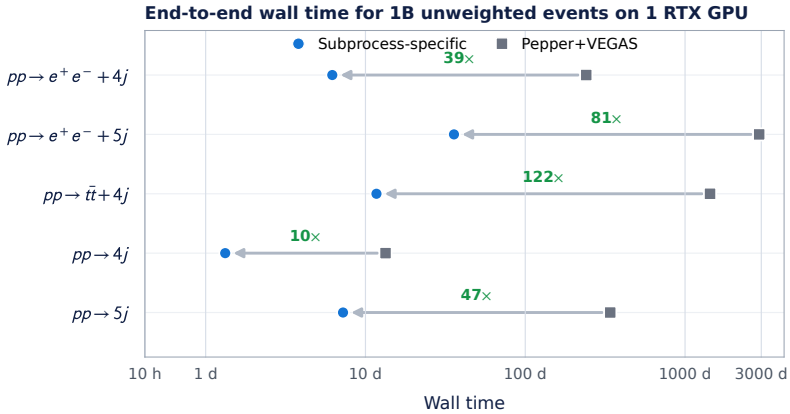
Both flow strategies become increasingly advantageous as the requested event sample grows

End-to-end Speedups at One Billion Events



Pepper+Flows runtimes include flow training and event generation. Pepper+VEGAS runtimes at 1B events are extrapolated from 1M events.

End-to-end Speedups at One Billion Events



Pepper+Flows runtimes include flow training and event generation. Pepper+VEGAS runtimes at 1B events are extrapolated from 1M events.

Takeaways

A **new Python layer** enables an **end-to-end GPU-resident workflow** that integrates normalizing-flow sampling, PEPPER GPU evaluation, and accept–reject unweighting.

VEGAS-initialized, helicity-conditioned flows reduce corrected-weight fluctuations and scale to **all subprocesses of five many-jet processes**.

On $4\times$ H100 GPUs, learned proposals provide **end-to-end speedups of up to** $83\times$ at 10^9 unweighted events. For $pp \to e^+e^- + 4j$, the wall time is reduced from 47.4 d to 13.7 h.