IHEP-CAS

Fast Simulation of Incoherent Pair Backgrounds at the CEPC Using Conditional Flow Matching

Yanbang Tang[1] Haoyu Shi[1] Zhongjian Ma[1]

Institute of High Energy Physics,
Chinese Academy of Sciences, Beijing

July 22nd 2026

Outline

1. Background & Motivation
2. Dataset & Method
3. Results & Validation
4. Conclusion & Outlook

CEPC and Incoherent Pair Backgrounds

Circular Electron Positron Collider (CEPC)

- e^+e^- collider at $\sqrt{s} = 240\,\mathrm{GeV}$ (Higgs factory)
- Target: $> 5\,\mathrm{ab}^{-1}$, $> 10^6$ Higgs events
- Precision measurements of Higgs couplings and electroweak observables

Beam–beam interaction produces pair backgrounds:

- **$\sim$2200 particles per bunch crossing**
- Energy: $0.5\,\mathrm{MeV}$ to $100\,\mathrm{GeV}$ (5 orders of magnitude)
- Strongly forward-peaked angular distribution
- Important background for detector design and physics analysis

Three quantum electrodynamics processes

Breit-Wheeler:	$\gamma\gamma \to e^+e^-$	1.2%
Bethe-Heitler:	$e\gamma \to ee^+e^-$	18.3%
Landau-Lifshitz:	$ee \to eee^+e^-$	80.5%

Key beam parameters @ CEPC Higgs:

- $E_{\mathrm{beam}} = 120\,\mathrm{GeV}$
- $N_b = 1.3 \times 10^{11}$
- $\sigma_x/\sigma_y = 14000/36\,\mathrm{nm}$
- Crossing angle: $33\,\mathrm{mrad}$

The Simulation Bottleneck

Guinea-Pig++: the standard simulation tool

- Particle-in-cell method with full quantum electrodynamics physics
- Provides 4-momentum + vertex for every particle
- Used for ILC, CLIC, FCC-ee, CEPC design studies

Our Goal

Train a generative model that **learns** the 7-dimensional phase-space distribution from Guinea-Pig++ data and generates new samples **rapidly**

The Problem

- $\sim$**10 hours per bunch crossing** on a single central processing unit core
- (Time measured on the CEPC Higgs dataset used in this study)
- Detector studies need 10^4–10^5 bunch crossing
- 10^5 bunch crossing $\times$ 10 h = **114 CPU-years**
- Infeasible for iterative design optimization

Guinea-Pig++ ($\sim$10 h/bunch crossing)

↓ Train once

Flow Matching (**$\sim$0.13 s/bunch crossing**)

↓

$\sim$270,000$\times$ faster

From Prior Work to the Research Gap

Generative models can accelerate simulations in high energy physics:

- Already proven for calorimeters (CaloGAN, CaloFlow, CaloDiffusion)
- Extended to full detector (FlashSim) and event generation
- Our problem has the same bottleneck: Guinea-Pig++ is too slow

Why Flow Matching:

- Stable, simulation-free continuous normalizing flow training
- Optimal Transport Conditional Flow Matching: direct linear paths
- Proven on high energy physics tabular data

Research Gap

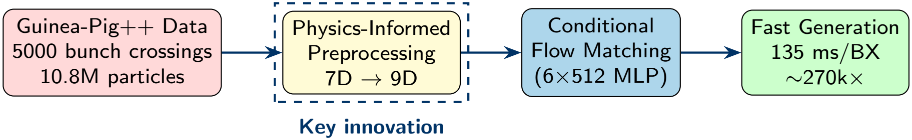
No prior work has applied generative models to beam–beam pair background simulation

— for CEPC or any lepton collider.

This Work

First application of deep generative models to incoherent pair background fast simulation

Approach Overview



Key Innovation

Physics-informed feature engineering transforms extreme distributions into learnable representations

Model Design

Single conditional model for all 3 processes, enabling cross-process information sharing

Practical Impact

100,000 bunch crossing: from 114 CPU-years to 3.7 GPU-hours

Dataset: Guinea-Pig++ CEPC Simulation

Data generation:

- 5000 independent bunch crossing, unique random seeds
- Total: $\mathbf{10.8 \times 10^6}$ particles
- Exact charge symmetry: 50% e^-, 50% e^+

Per-bunch-crossing statistics:

- Particles per crossing: 2157 ± 64
- Range: [1894, 2398]

Data split (by bunch crossing, no leakage):

4000 train / 500 val / 500 test

Raw features (7-dimensional + label):

Variable	Unit	Range
Energy E	GeV	[$\pm$0.5 MeV, $\pm$100 GeV]
β_x, β_y, β_z	—	$(-1, 1)$
x, y, z	nm	$\pm 10^7$
Process	0/1/2	BW/BH/LL

Challenges for machine learning

- Energy spans 5 orders of magnitude
- β_z concentrates at ± 1
- Extreme beam aspect ratio ($\sigma_y/\sigma_x \approx 1/400$)

Physics-Informed Preprocessing

Core idea: transform raw kinematics into physically motivated, machine-learning-friendly features

Raw variable	Transform	Rationale
Energy E	$\mathrm{sign}(E)$, $\log_{10}\lvert E\rvert$	5 orders of magnitude
β_x, β_y	$\log_{10}(p_T)$	Heavy tail
	$\sin\phi$, $\cos\phi$	Circular topology
β_z	$\Phi^{-1}(\hat{F}(\beta_z))$	Density at ± 1
x, y, z	$(\cdot - \mu)/\sigma$	Unit variance

$\Rightarrow$ 7-dimensional raw features become **9-dimensional physics-informed features** plus global standardization

Quantile Normalization for β_z

$$\beta_z \xrightarrow{\hat{F}} [0,1] \xrightarrow{\Phi^{-1}} \mathbb{R}$$

- 50,000 quantile points from training data
- Avoids artanh divergence at $\beta_z = \pm 1$
- Rank-based $\Rightarrow$ robust to extreme values

p_T/ϕ Decomposition

$(\beta_x, \beta_y) \to (\log_{10} p_T,\ \sin\phi,\ \cos\phi)$

Decouples momentum scale from direction

Optimal Transport Conditional Flow Matching Model

Optimal Transport Conditional Flow Matching

- Learn velocity field v_θ transporting $\mathcal{N}(0, I)$ to data
- **Linear interpolation path:**

$$x_t = (1 - t)\, x_0 + t\, x_1$$

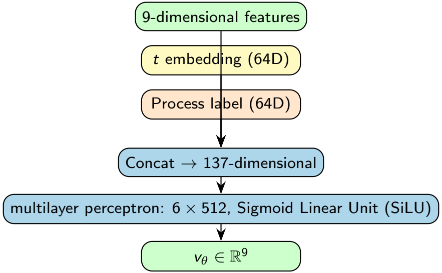
- **Training loss (mean squared error):**

$$\mathcal{L} = \mathbb{E}_{t,x_0,x_1,c}\left[\left\| v_\theta(x_t, t, c) - (x_1 - x_0)\right\|^2\right]$$

- **Generation:** solve ordinary differential equation from $t=0$ to $t=1$ using 100 midpoint (second-order Runge-Kutta) steps

Simulation-free training; no Markov chain Monte Carlo, no adversarial loss

Architecture



Parameters	1.39M
Optimizer	AdamW (lr 10^{-3})
Epochs	200
GPU	NVIDIA A800
Training time	∼80 min

Process-Conditional Generation Strategy

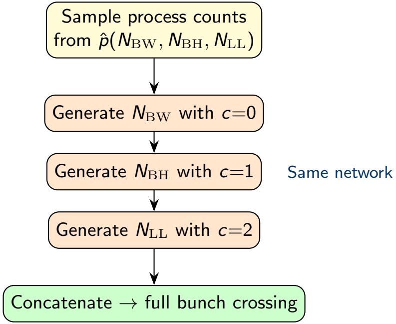
Single conditional model (Marginal Flow Matching)
for all 3 processes

- Process label → 64-dimensional learned embedding
- Network learns shared & process-specific features
- **Advantages:**
  - Parameter-efficient (one model vs. three)
  - Cross-process physics learning
  - No data starvation for rare processes (Breit-Wheeler: 1.2%)

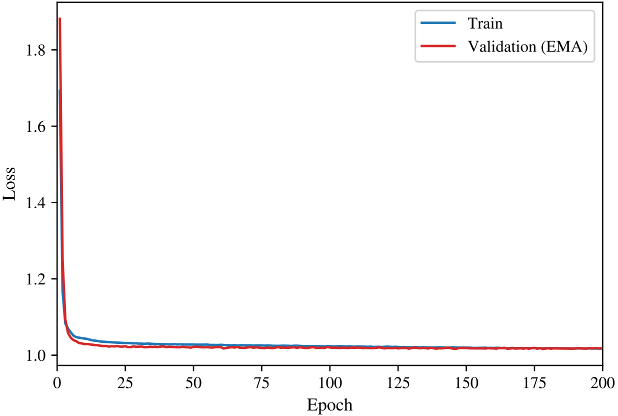
Alternative — Combined approach:

- 3 independent per-process models (no conditioning)
- → see comparison in Results

BX generation pipeline:



Training Convergence



Stable convergence, no overfitting

- All 4 model variants shown
- Validation loss tracks training loss
- Best validation loss ≈ 1.016 (epoch 146)
- exponential moving average (decay 0.999) for generation

Property	Value
Architecture	6×512 MLP, SiLU
Parameters	1.39M
Batch size	4096
Learning rate schedule	Cosine + 10-epoch warmup
GPU	NVIDIA A800 80 GB
Wall time	$\sim$80 min

Evaluation Metrics

Kolmogorov-Smirnov (KS) statistic

- Measures maximum difference between two cumulative distributions
- Range: 0 (identical) to 1 (completely different)
- Computed for each 1-dimensional marginal distribution
- **Lower KS = better agreement**

Maximum Mean Discrepancy (MMD)

- Measures distance between two distributions in the full feature space
- Uses a Gaussian (RBF) kernel
- Captures all correlations, not just 1-dimensional projections
- **Lower MMD = better overall agreement**

We compare three surrogate methods against the Guinea-Pig++ test set.

Comparison: Three Surrogate Methods

Kernel Density Estimation (KDE): a classical statistical method that places a Gaussian kernel on each training data point. Serves as a non-machine-learning baseline to contrast with our machine learning approach. Fast (6 ms/BX) but low fidelity.

Variable KS value	Kernel Density Est.	**Marginal Flow Matching**	Combined Flow Matching
$\lvert E \rvert$	0.030	**0.0009**	0.0011
β_x	0.139	**0.0012**	0.0009
β_y	0.170	**0.0014**	0.0022
β_z	0.024	**0.0132**	0.0143
x	0.094	**0.0012**	0.0011
y	0.156	**0.0117**	0.0088
z	0.055	**0.0015**	0.0017
p_T	0.297	**0.0017**	0.0015
θ	0.308	**0.0020**	0.0021
MMD	2.82×10^{-3}	2.8×10^{-5}	1.17×10^{-4}
Speed (ms/BX)	6	135	219

Marginal Flow Matching: MMD 100× better than kernel density estimation

All KS values < 0.013 (Marginal Flow Matching)

Marginal Flow Matching vs. Combined Flow Matching: Why One Model Outperforms

	Marginal FM	Combined FM
Models	1 shared	3 independent
Parameters	1.39M	$\sim$4.2M
Conditioning	Process label	None
MMD (RBF)	2.8×10^{-5}	1.17×10^{-4}
Speed (ms/BX)	135	219

Single shared model: **4.2**$\times$ better MMD, **1.6**$\times$ faster

Why the shared model outperforms

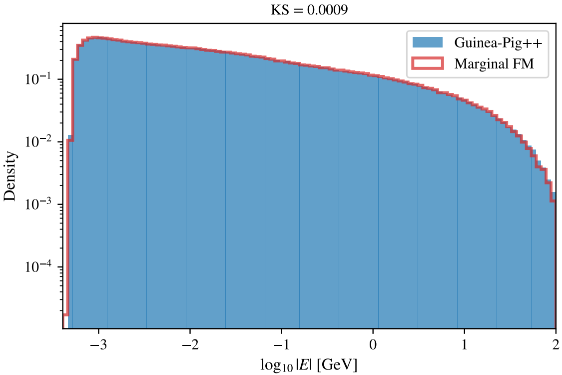
Cross-process information sharing:

- Shared spatial distributions across all processes
- Shared energy–velocity kinematic correlations
- Process identity encoded via conditional embedding
- Breit-Wheeler (1.2% of data) benefits from shared features

Practical Advantage

One model to train, validate, and deploy —
vs. three separate training pipelines

Distribution Fidelity: Energy Spectrum



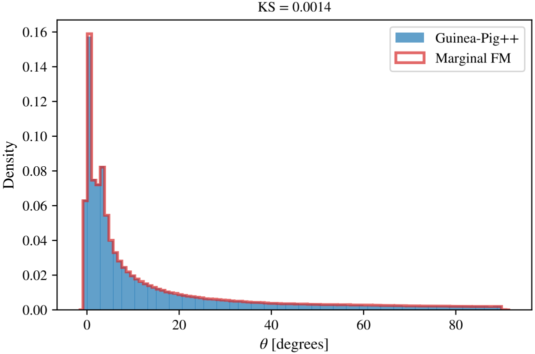
Log-scale energy spectrum

- Covers **5 orders of magnitude** ($0.5\,\mathrm{MeV}$–$100\,\mathrm{GeV}$)
- Generated and real distributions show close agreement
- KS: $|E|$ KS $= 0.0009$

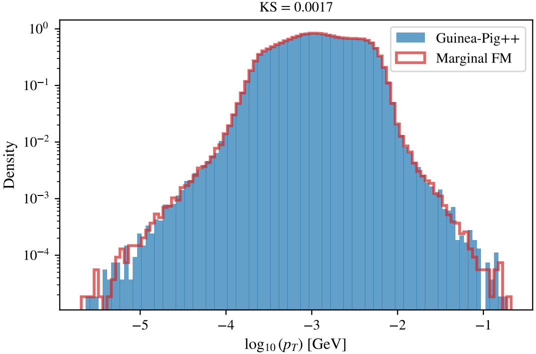
Why it works

$\log_{10}|E|$ transform compresses 5 orders of magnitude into a compact range, making the distribution learnable

Distribution Fidelity: Angular Structure



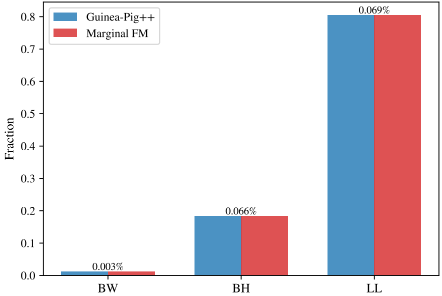
Polar angle θ: KS $= 0.0020$



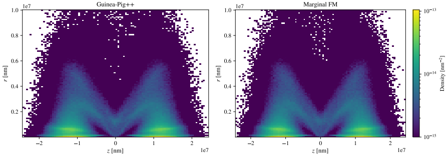
Transverse momentum p_T: KS $= 0.0017$

- θ: strong forward peak near beam axis
- Model accurately reproduces the peaked structure
- ϕ: central peak + flat wings (beam geometry)
- $(\sin\phi, \cos\phi)$ encoding avoids $\pm\pi$ discontinuity

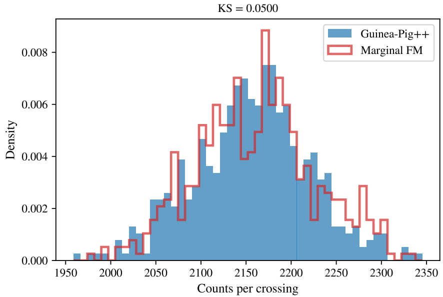
Structural Fidelity: Process Fractions & Counts



Process fraction absolute errors $< 0.1\%$



z–r vertex structure reproduced



Counts per crossing:

- Real: 2156.9 ± 63.9
- Generated: 2161.2 ± 68.2
- KS $= 0.050$, $p = 0.56$

z–r: beam interaction geometry correctly captured

Preprocessing Ablation: Every Component Matters

2×2 **factorial design:** Quantile Normalization $\times$ p_T/ϕ decomposition

	Full	$-$QN	$-p_T/\phi$	Naive
QN(β_z)	✓	—	✓	—
p_T/ϕ decomp.	✓	✓	—	—
Dimension	9	9	8	8
KS(β_z)	0.013	0.056 (4×)	0.014	0.056 (4×)
KS(β_x)	0.001	0.001	0.004 (4×)	0.004 (3×)
KS(β_y)	0.001	0.002	0.007 (5×)	0.006 (5×)
KS(p_T)	0.002	0.001	0.008 (5×)	0.006 (4×)
KS(θ)	0.002	0.008 (4×)	0.005 (3×)	0.015 (8×)

Key Finding

Both innovations are **complementary and important**:

- **QN(β_z)**: handles sharp bimodal structure at ± 1 (artanh diverges); β_z KS degrades 4× without it
- p_T/ϕ **decomp.**: decouples magnitude from azimuth; p_T KS degrades 5× without it
- **Naive (both removed)**: severe degradation — θ KS 8× worse, MMD becomes NaN

Generation Speed: Speed Comparison

Guinea-Pig++		Flow Matching
~10 hours / bunch crossing	**~270,000×** →	**~135 ms / bunch crossing** (GPU)
(single central processing unit core)		~4 s / bunch crossing (CPU)

For 100,000 bunch crossing dataset:

Guinea-Pig++	~114 CPU-years
Flow Matching (GPU)	**~3.7 hours**
Flow Matching (CPU)	**~112 hours**

What this enables:

- Rapid detector geometry optimization
- Large-scale background overlay studies
- Interactive design–evaluate cycles

Conclusions

1. **First deep generative model for beam–beam pair backgrounds**
   Novel application domain: no prior work for CEPC or any lepton collider
2. **Physics-informed preprocessing is helpful**
   Quantile normalization for β_z and p_T/ϕ decomposition
   are important — validated by ablation study
3. **Strong fidelity across all distributions**
   $\text{MMD} = 2.8 \times 10^{-5}$ ($100\times$ better than kernel density estimation)
   $|E|$ KS $= 0.0009$, θ KS $= 0.0020$, p_T KS $= 0.0017$
4. **$\sim$270,000$\times$ speedup over Guinea-Pig++**
   100,000 BX: 114 CPU-years $\rightarrow$ 3.7 GPU-hours (or 112 CPU-hours)

Flow matching: a practical, high-fidelity surrogate for beam–beam pair background simulation at future e^+e^- colliders.

Outlook & Future Directions

Near-term improvements:

- **Beam parameter conditioning**
  One model covering multiple $\sqrt{s}$ and luminosity settings
- **Flow matching distillation**
  Reduce ODE steps from 100 to $\sim$10
  $\Rightarrow$ additional 10$\times$ speedup
- **Energy-momentum conservation**
  Enforce per-bunch-crossing balance constraints

Deployment & generalization:

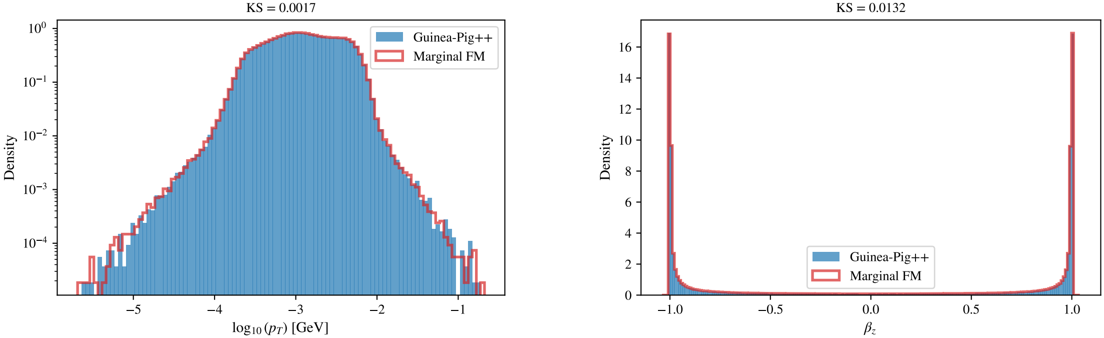
- **Key4HEP / DD4hep integration**
  Plugin for seamless Guinea-Pig++ replacement in the detector simulation framework
- **Other colliders**
  FCC-ee, CLIC, ILC — same methodology, different beam parameters
- **Uncertainty quantification**
  Ensemble or Bayesian approaches for per-sample confidence estimates

Thank You

Questions and Discussion

Yan-Bang Tang — IHEP, Chinese Academy of Sciences

Backup: Momentum & Velocity Distributions

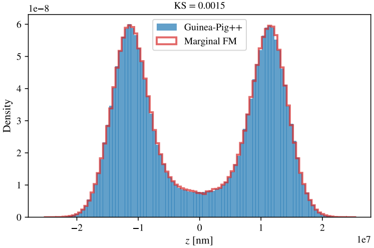


p_T distribution (log-scale): KS $= 0.0017$

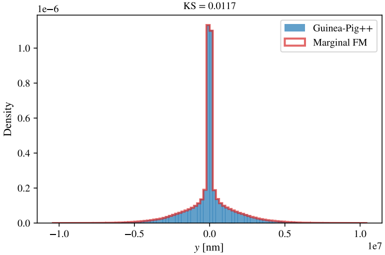
β_z distribution: KS $= 0.0132$

- p_T spans several orders of magnitude; $\log_{10} p_T$ transform enables accurate learning
- β_z concentrates near ± 1; quantile normalization maps to standard normal

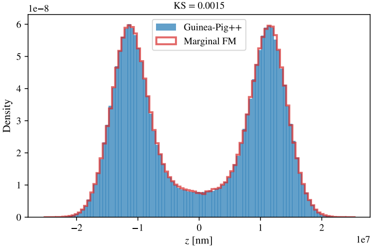
Backup: Vertex Position Distributions



x: KS $= 0.0012$



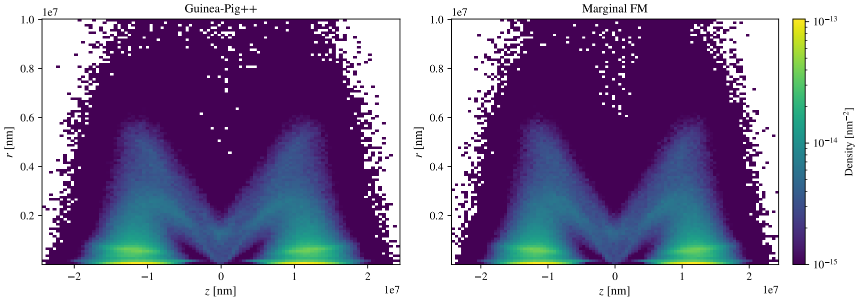
y: KS $= 0.0117$



z: KS $= 0.0015$

- y distribution is $\sim$400$\times$ narrower than x ($\sigma_y/\sigma_x \approx 1/400$)
- z shows double-peaked structure from beam interaction geometry

Backup: z–r Vertex Structure



Two-dimensional vertex distribution in (z, r) with logarithmic density scale. The spatial extent and density structure of the beam interaction region are reproduced by the Marginal Flow Matching model.