# Search for CME with Transformers

Aihong Tang

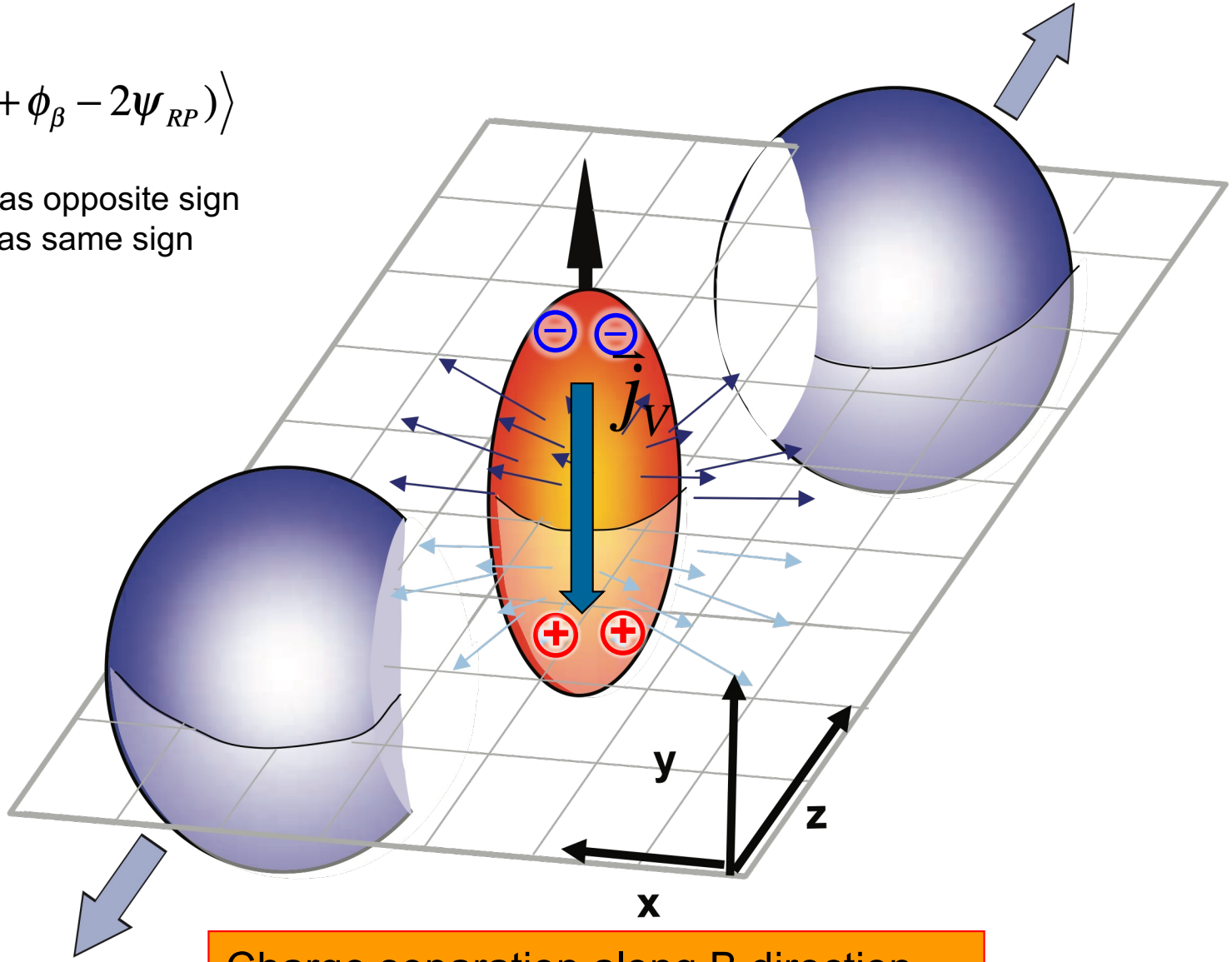
Grateful to Ivan Kisel and his group for encouragement and early guidance in AI methods

# Chiral Magnetic Effect

$$\gamma_{SS} < \gamma_{OS}$$

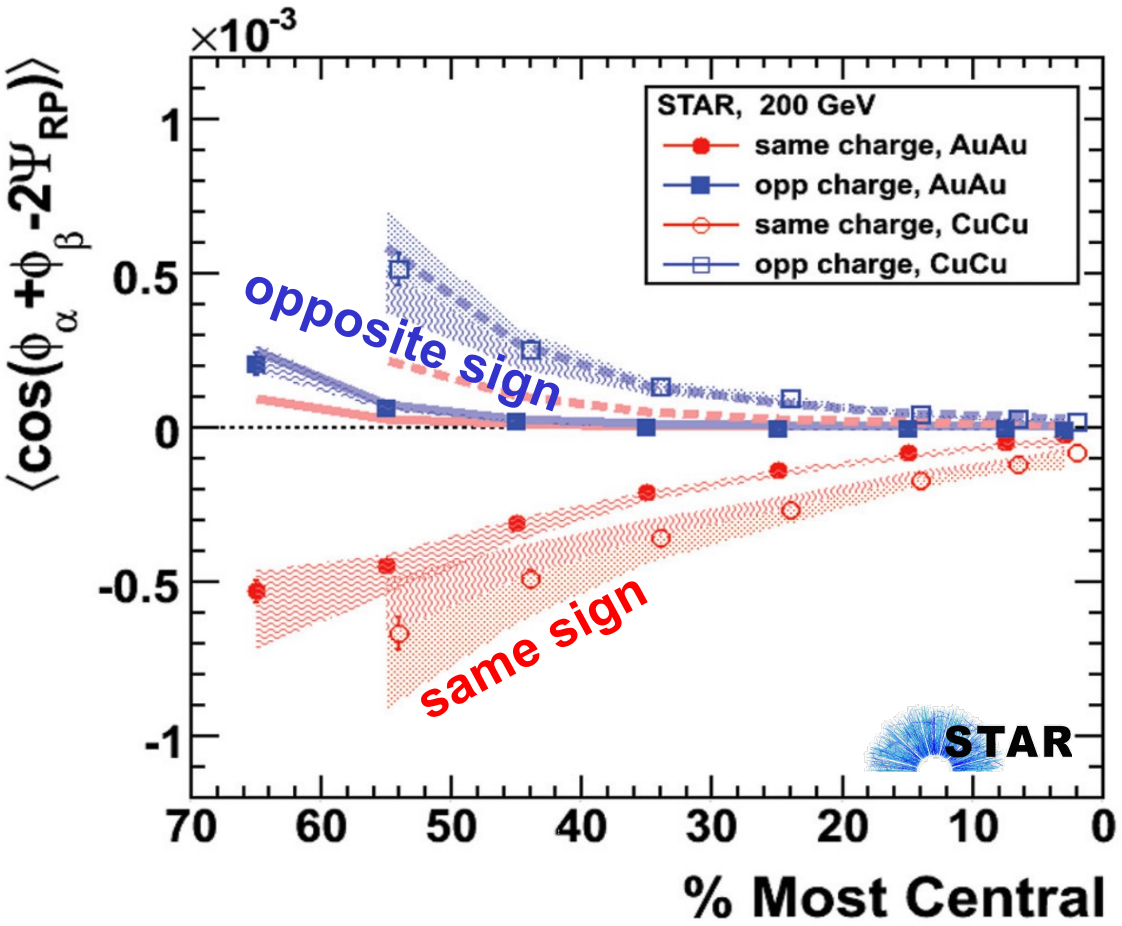
$$\gamma \equiv \left\langle \cos(\phi_\alpha + \phi_\beta - 2\psi_{RP}) \right\rangle$$

os : α and β has opposite sign
ss : α and β has same sign




Charge separation along B direction.
Dipole effect, flips event by event.

# Chiral Magnetic Effect




$$\gamma_{SS} < \gamma_{OS}$$

$$\gamma \equiv \left\langle \cos(\phi_\alpha + \phi_\beta - 2\psi_{RP}) \right\rangle$$

$\Delta\gamma$, $\Delta\delta$, and $\kappa$.
Event Shape Engineering,
Event Shape Selection,
Signed Balance Function,
PP/RP,
R correlator,
...

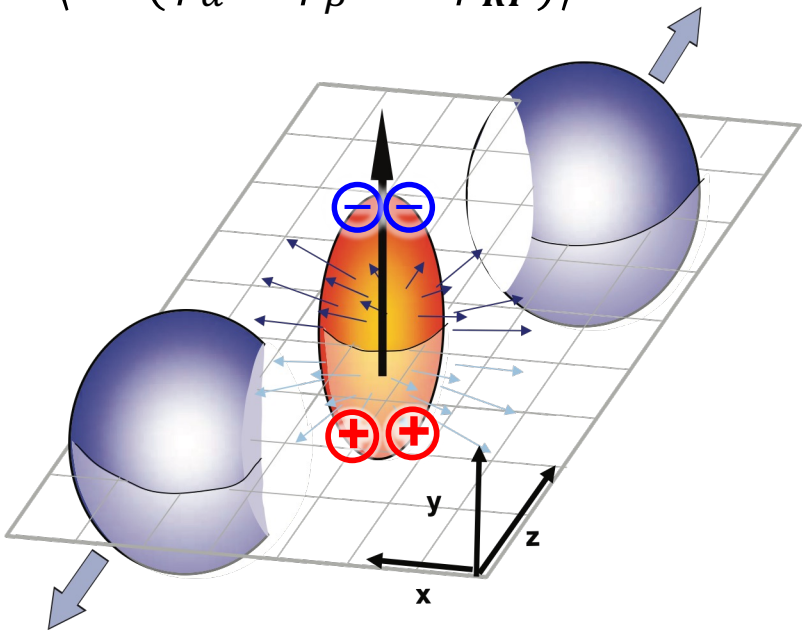
STAR PRL103 251601 (2009)
STAR PRC 81 054908 (2010)

"A signal consistent with several expectations from the theory is detected."

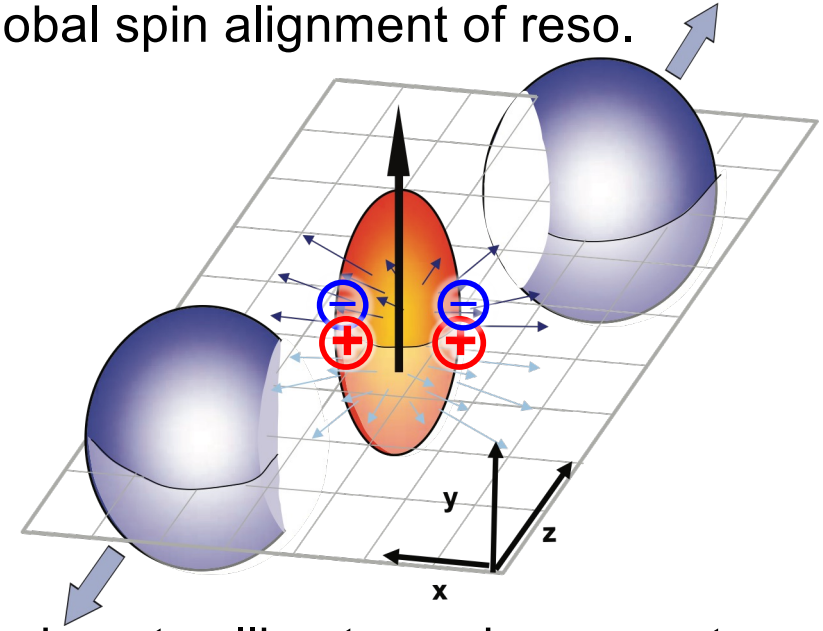
# Chiral Magnetic Effect : Backgrounds

Signal

$$\gamma = \langle \cos(\phi_\alpha + \phi_\beta - 2\psi_{RP}) \rangle$$




Backgrounds:
Total Momentum Conservation (TMC)
Local Charge Conservation (LCC)
Flowing cluster in plane
Global spin alignment of reso.
…




Flow boost collimates pairs more strongly in-plane than out of plane. Most background is driven by flow ($v_2$)

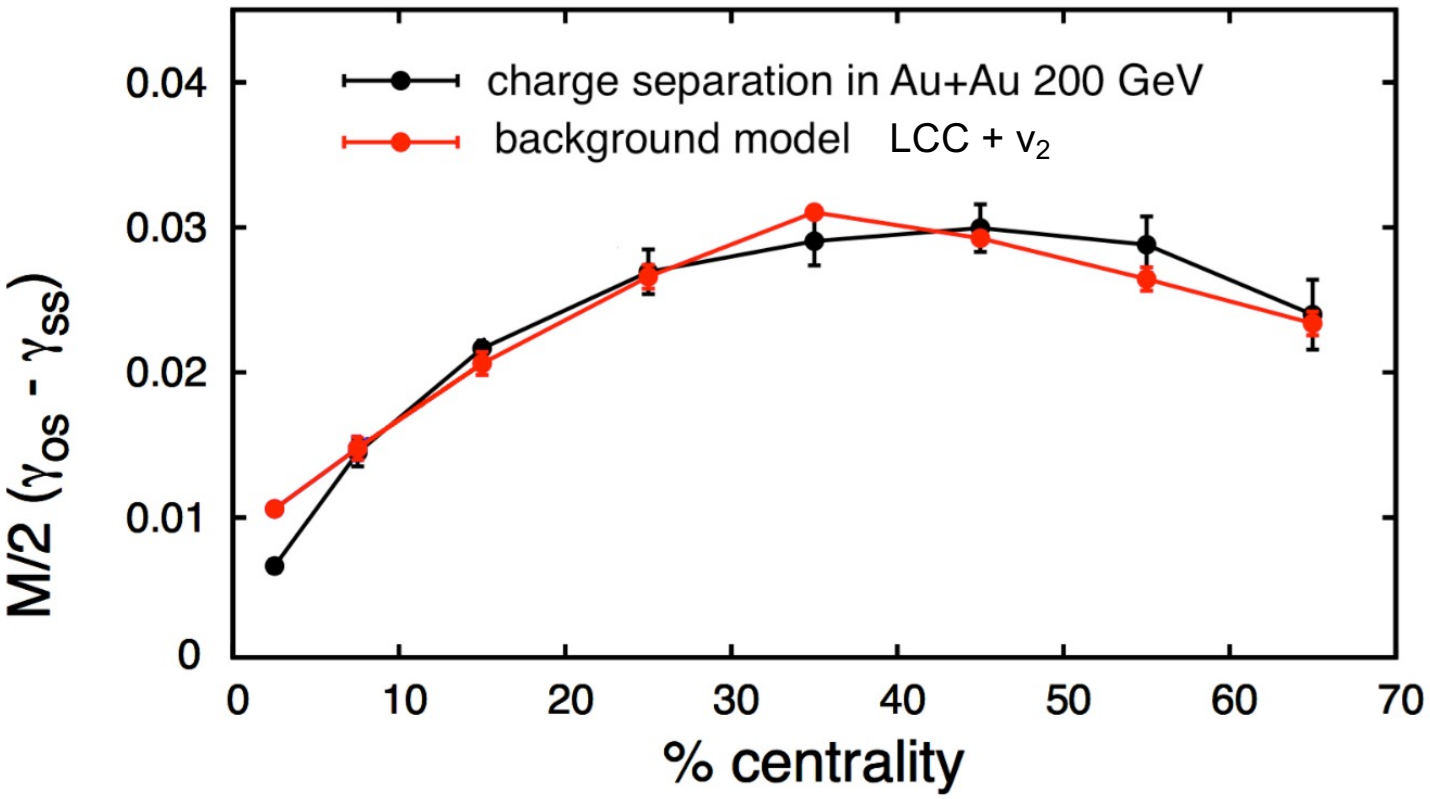
Signal OS : $\phi_\alpha + \phi_\beta = \frac{\pi}{2} + \frac{3\pi}{2} = 2\pi$

Bkg OS : $\phi_\alpha + \phi_\beta = 0$ or $2\pi$


A. Bzdak, V. Koch, and J. Liao , Lect. Notes Phys. 871, 503 (2013)
S. Pratt, S. Shlichting and S. Gavin, PRC 84 024909 (2011)
S. Schlichting and S. Pratt, PRC 83 014913 (2011)
F. Wang PRC 81 064902 (2010)
A. Tang, Chin. Phys. C 44 No.5 054101 (2020)
…

# Chiral Magnetic Effect : Backgrounds

S. Schlichting and S. Pratt, Phys. Rev. C **83**, 014913 (2011)




Small signal in a complex, large background.
Could AI help ?

# Transformer

**Transformer is the engine behind ChatGPT**

- **Self-attention = adaptive correlations.** Each layer forms an all-to-all, data-driven "correlation matrix" between inputs.

- **It's the core of ChatGPT.** ChatGPT is a large Transformer pretrained to predict the next word; the same machinery can be reused for other tasks by adding a tiny task-specific final step that turns its internal summary into the target number we want.

- **Why it helps us.** The model naturally learns many-body patterns and works with multiple variables.

# Why Not Other AI Models?

- Convolutional Neural Network (**CNN**) and Recurrent Neural Network (**RNN**) both assume a fixed shape and lack self-attention (no easy interaction among inputs). Difficult to see correlations.

- Gradient-boosted Decision Trees **(GBDT**). Great for tabular, hand-crafted features, but not naturally suited for learning complex cross-particle correlations directly from raw particle data.

Transformer's self-attention makes it a natural choice for correlation study, and stands out of other models for our task.

# Toy Model Input

- Same model as used in STAR method paper "Investigation of Experimental Observables in Search of the CME …"
  - For 30-40% AuAu at 200 GeV
  - Realistic spectra and $v_2$
  - ρ-meson resonance with flow
  - CME signal via $a_1$

$$\frac{dN_{\pm}}{d\Delta\phi} \propto 1+2v_1\cos\Delta\phi+2v_2\cos2\Delta\phi+2v_3\cos3\Delta\phi+\cdots+2a_{\pm}\sin\Delta\phi+\cdots$$

$$a_1 \equiv |a_+| \equiv |a_-|$$

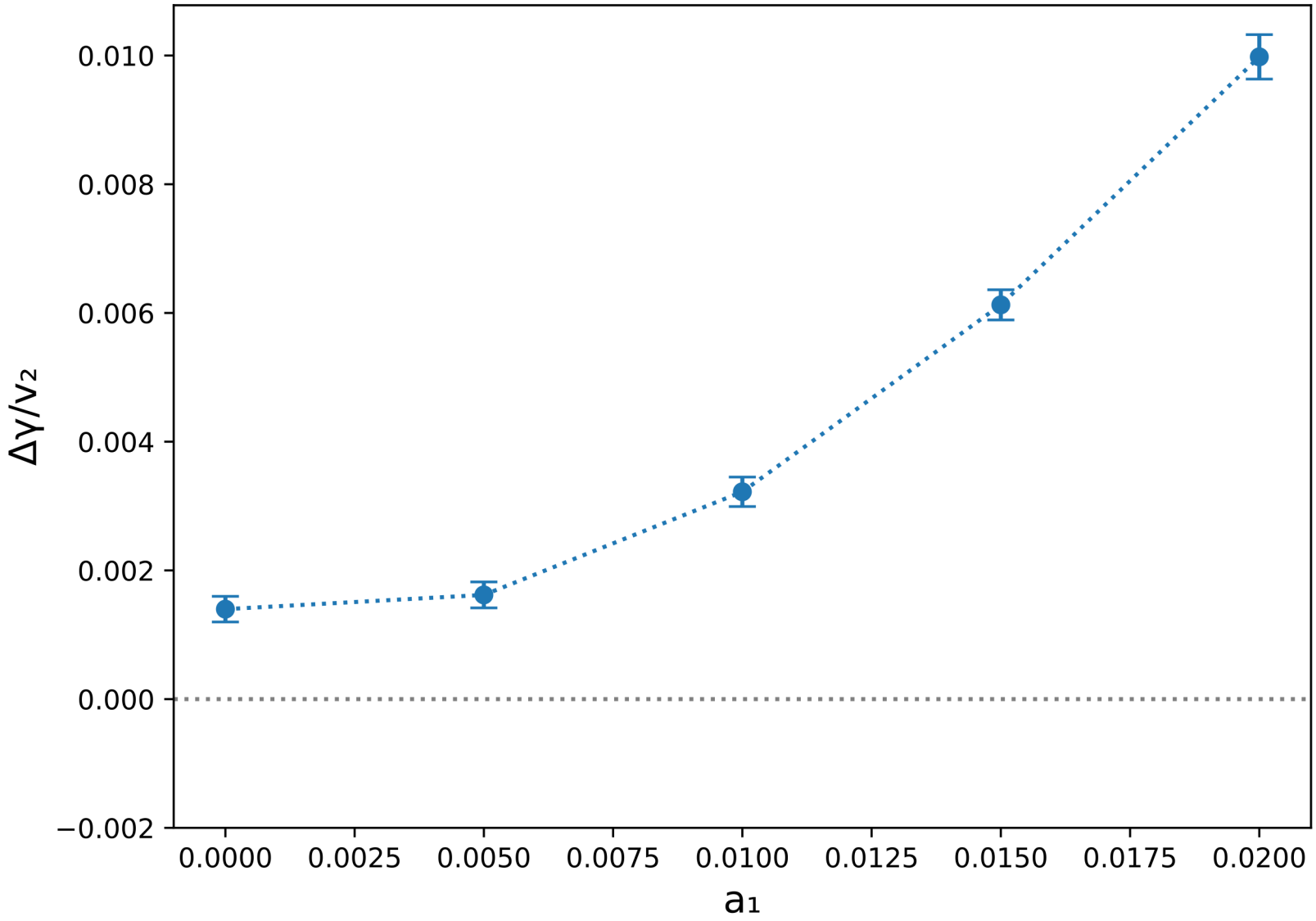
Chinese Phys. C 46 14101 (2022)

+

- Ks resonance with flow
- Pt efficiency resembling TPC efficiency loss
- Positive and negative charge $v_2$ , $v_3$, and spectra difference

New additions

# Toy Model


0.010
0.008
0.006
0.004
0.002
0.000
−0.002
Δγ/v₂
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
a₁

# Inputs to the Transformer Model

**Particle-level inputs**

q,
$m^2$,
$E^2$,
$p^2$,
pt,
η,
$p_z$,
ϕ, (w.r.t EP)
c1= cosϕ,
s1= sinϕ,
c2,
s2,
c3,
s3,
ptqc1 = pt*q*c1,
ptqs1 = pt*q*s1,
ptqc2,
ptqs2,
ptqc3,
ptqs3

**Event-level inputs**

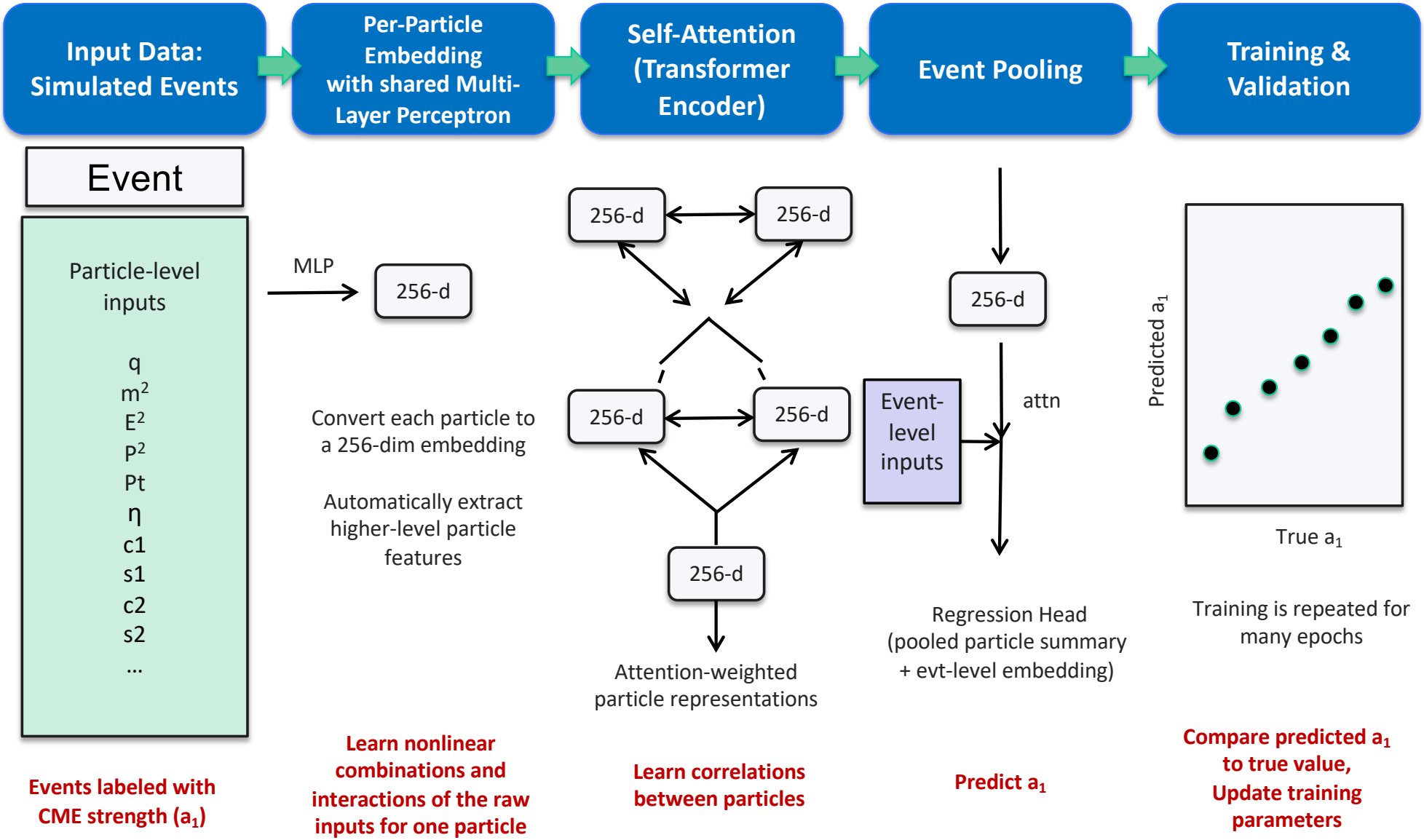
hist_bin_0 … hist_bin_15
(net charge in 16 azimuthal bin w.r.t EP)

$Dx = \sum q \cdot \cos(\phi) / \sum |q|$ (q-weighted dipole)
$Dy = \sum q \cdot \sin(\phi) / \sum |q|$
$Dx^2$
$Dy^2$
$Dy^2 - Dx^2$

$ptDx = \sum pt \cdot q \cdot \cos(\phi) / \sum | pt \cdot q|$ (pt*q-weighted dipole)
ptDy
$ptDx^2$
$ptDy^2$
$ptDy^2 - ptDx^2$

**Particle- and event-level inputs**

# Transformer for CME Regression



# Three-stage Training Workflow

**Head-only warmup**

- 2-epoch warmup with encoder frozen
- Stabilizes the initial output scale before full optimization

**Main end-to-end training with low-$a_1$ shaping**

- Full-network optimization
- Low-$a_1$-focused shaping

**Validation and checkpoint selection**

- Final model chosen on held-out validation
- Performance evaluation

# Implementation Provenance

- **Core engine:** PyTorch (vanilla modules -- nn.TransformerEncoder, autograd, AdamW).
- **Custom code:** Particle-encoder MLP, attention/mean pooling, regression head, slope/bias/variance losses, calibration, policy selection.
- **Data pipeline:** Toy-model CME generator + iterable data loader with padding/masks.

# Model Configuration

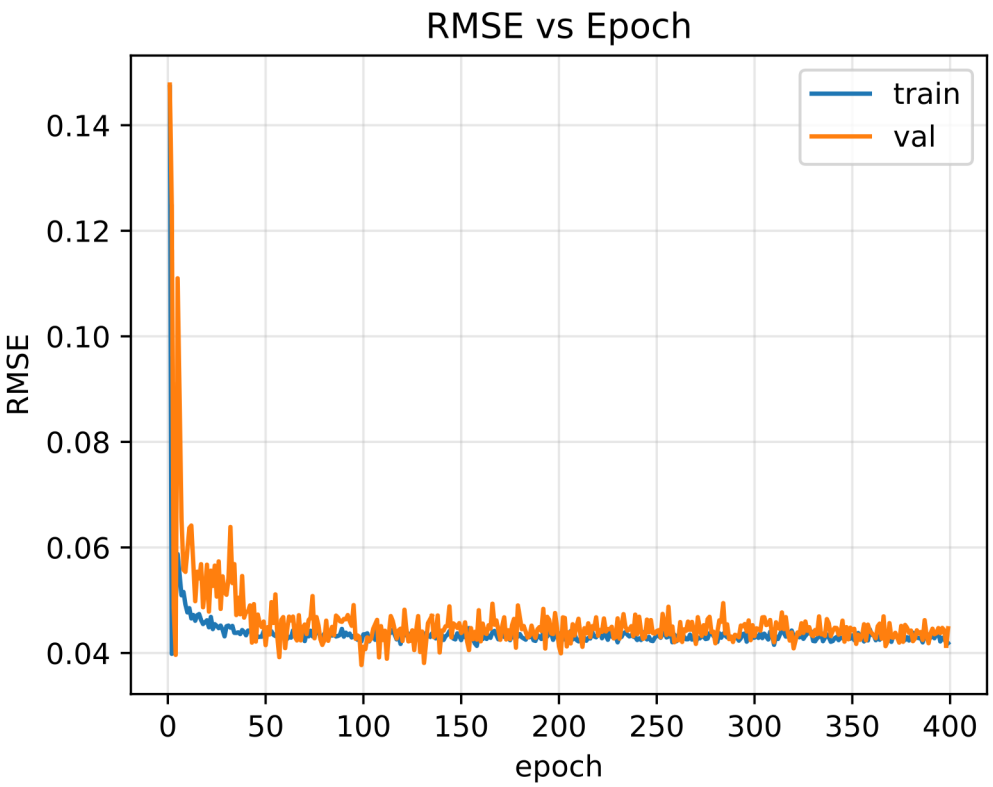
- **Inputs and encoders:** Per-particle two-layer MLP (P → 256 → 256, Layernorm + GELU).
- **Backbone and pooling:** 6x Transformer encoder layers [ d_model = 256, 8 heads, FFN (feed forward network) = 1024, GELU (soft gate), ] with attention pooling over learned particle weights.
- **Output and size:** Two-layer regression head + lightweight output calibrator; total trainable parameters ~ 5M.

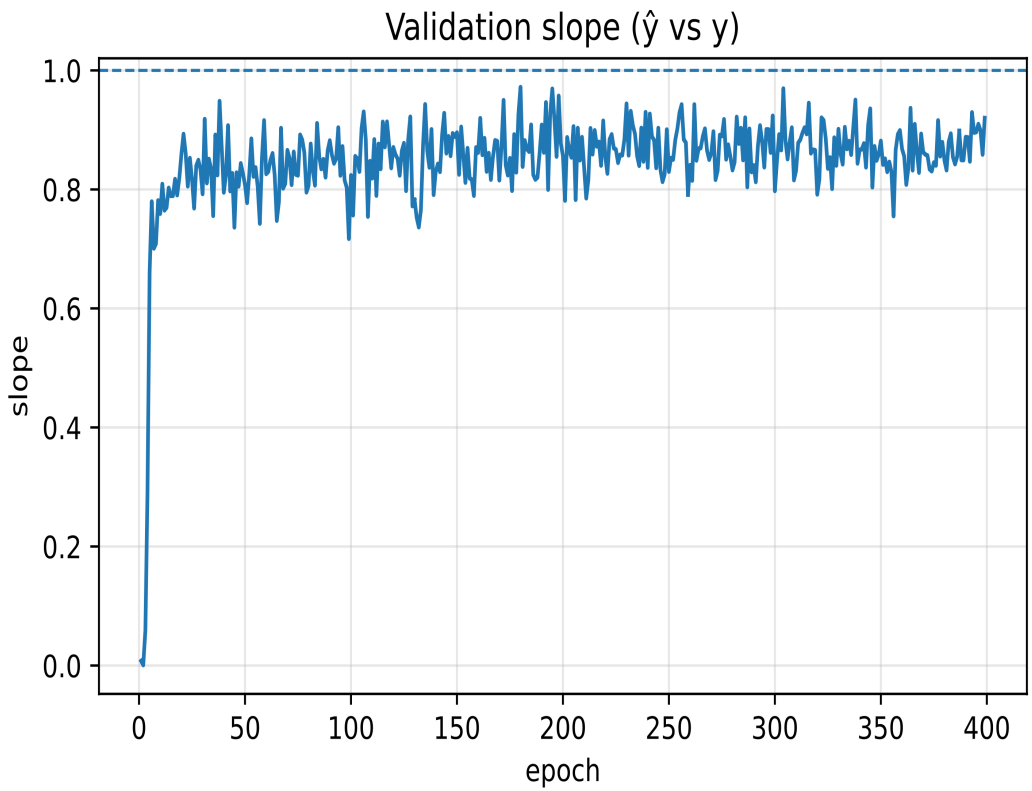
For context : ChatGPT-3 used 96 decoder layers, d_model = 12,288 96 heads, FFN=4xd_model, for a total of 175 billion parameters.

# Training Setup

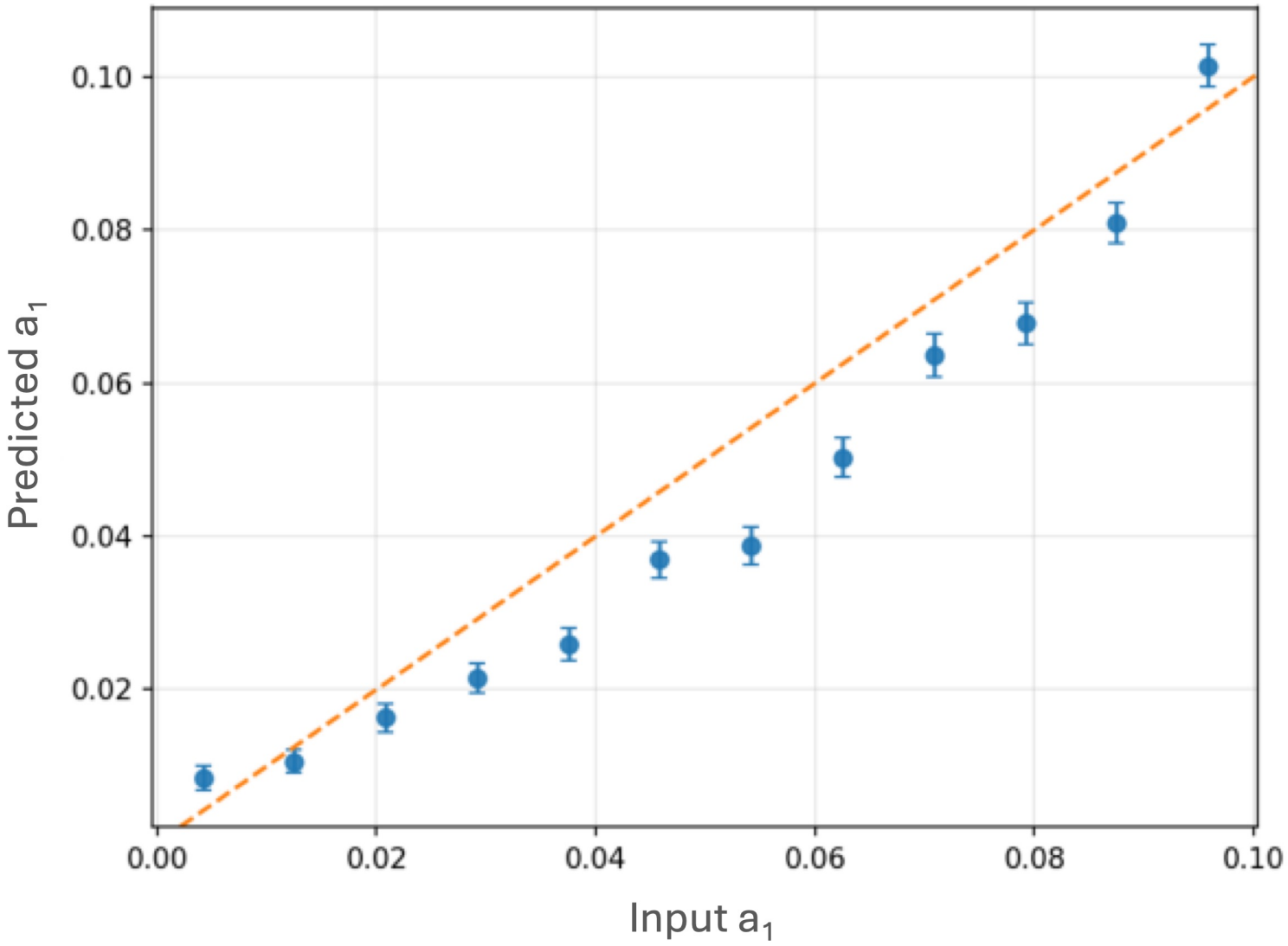
- **Data per epoch:** Train 10k events (mixed source, 90% base + 10% zero-$a_1$), validation 5k events.
- **Batching and update:** 256 events per effective batch, 40 batches per epoch (40 updates per epoch).
- **Run length and cadence:** Typical runs are O(100) epochs, 12-18 hours total. Validation every epoch.
- **Hardware:** Trained with MacBook's GPU via Apple's MPS backend. 2023 Macbook Pro (M3), 12 CPU cores, 18 GPU cores and 18 GB memory.

# Training Progress



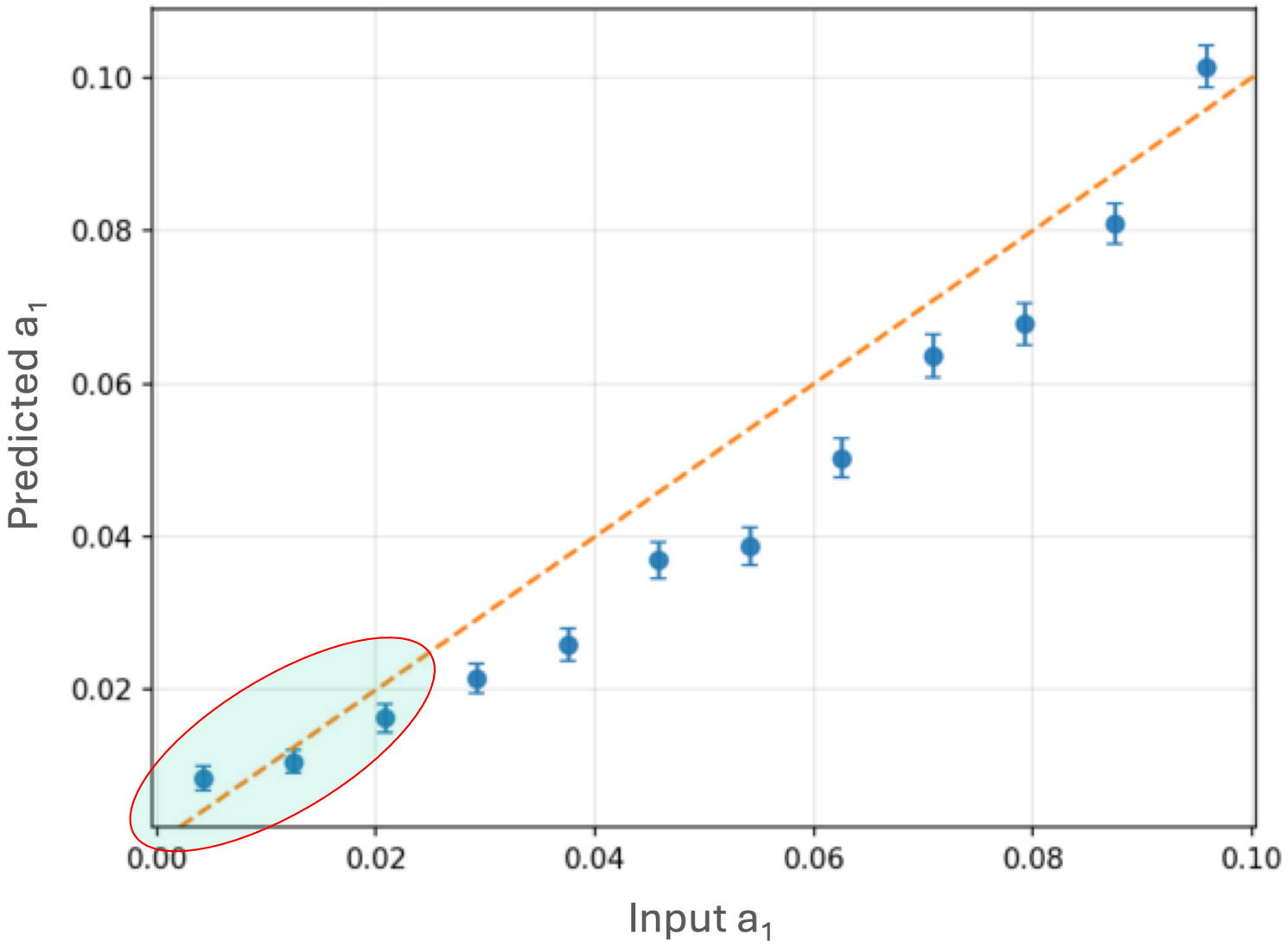



# Training Result




The result provides a proof of principle: the model exhibits an approximately linear response to increasing CME strength in the presence of typical resonance-flow backgrounds.

# Training Result




The prediction is reasonable but not yet optimal.
Further improvement may come from additional hyperparameter tuning and larger-scale training on dedicated GPU resources.

# Conclusion

We have developed and tested a Transformer-based AI framework for CME studies

The present results establish a proof of principle for learning CME signal directly from data via Transformer.

Good overall performance has been achieved, but precise extraction in the weak-signal region remains the key open issue.

Future directions :
Improve low-$a_1$ performance and overall calibration.
Train on larger AVFD datasets.
Scale to deeper models and longer training schedules.
Improve portability and access to stronger computing resources.