ADKDD 2026 | KDD | JEJU, SOUTH KOREA

CADET

Context-Conditioned Ads CTR Prediction with a Decoder-Only Transformer

David Pardoe · Neil Daftary · Miro Furtado · Aditya Aiyer · Ruoyan Wang · et al.

LinkedIn Ads AI

Presented by **Neil Daftary**

What CTR prediction does in LinkedIn's homefeed

1

Member opens the feed

The request provides member, device, and feed context.

2

Candidate ads are retrieved

Many eligible sponsored updates may compete for a small number of slots.

3

The model estimates CTR

Each candidate receives an estimate of its probability of being clicked.

4

The auction ranks candidates

The CTR estimate contributes to ranking and ad placement.

More accurate CTR estimates improve ad relevance for members and help advertisers obtain more value from their budgets.

MODEL EVOLUTION

From a hybrid CTR ensemble to one sequence model

LiRank production baseline

DCNv2

feature interactions

+

TransAct

behavioral history

CADET

Decoder-only transformer

Learns feature interactions and temporal behavior in one user-ad sequence

Unified recommenders such as HSTU, OneRec, and MTGR follow favorable scaling laws, improving with more data and compute where DLRMs plateau. CADET adapts this approach to ads CTR prediction under production constraints.

PRODUCTION SETTING

Requirements for a decoder-only ads CTR model

1 **Post-scoring context**

Final position is not known when candidates are scored.

2 **Train-serve consistency**

Recent events may be visible offline but unavailable online.

3 **Temporal modeling**

Behavior depends on the time gaps between events, from seconds to months.

4 **Industrial efficiency**

Long histories and many candidates must meet latency constraints.

Each requirement motivates a corresponding architectural or systems component in CADET.

MODEL OVERVIEW

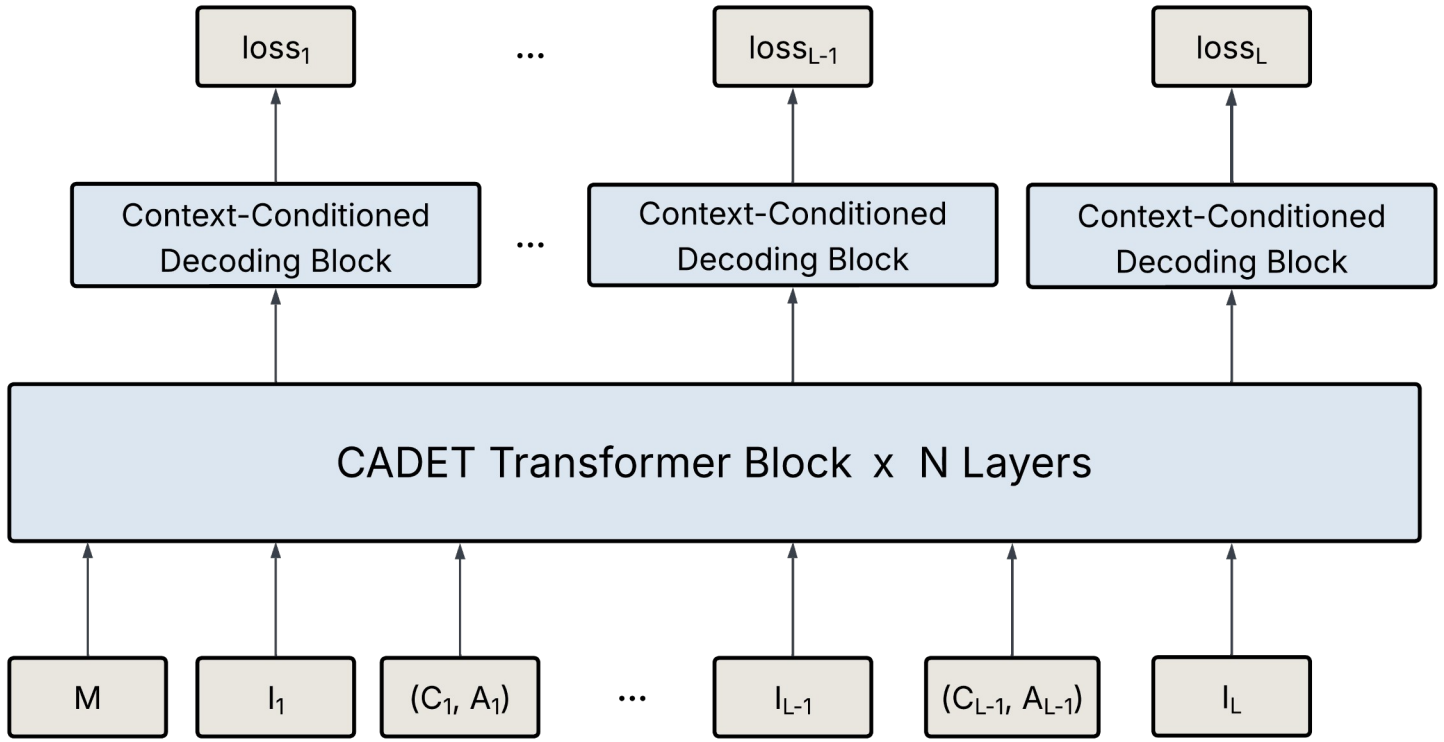
CADET combines a shared transformer with coordinated adaptations

One shared model replaces the LiRank CTR ensemble. Each adaptation addresses a distinct production requirement.

User history and candidate impressions form one causal sequence

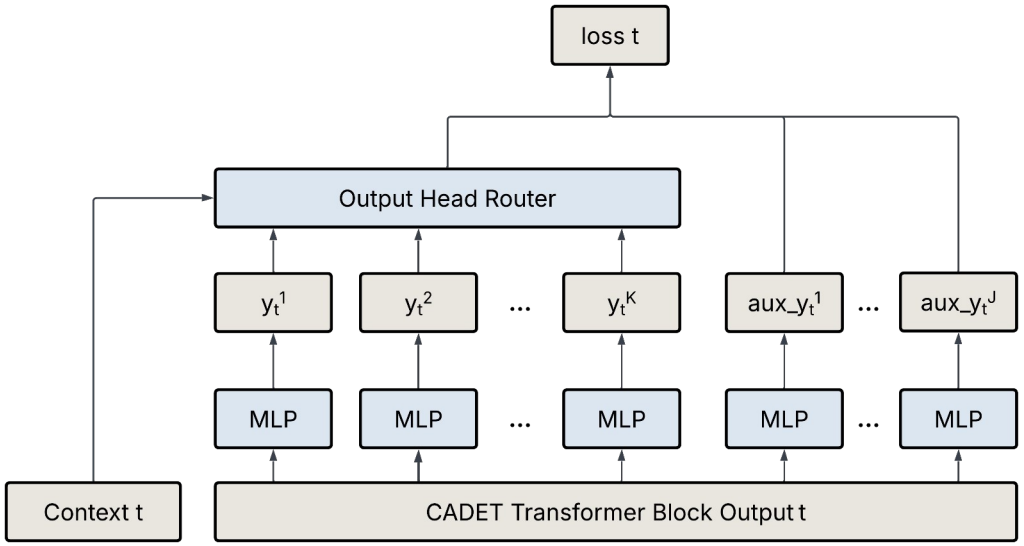
- M contains static member and profile features.
- Each historical impression I is followed by its post-scoring context C (e.g., ad position) and the action A that occurred.
- The current candidate impression has no observed context or action yet.
- During training, the model predicts every historical action in one causal pass.

M ; I1 , (C1 , A1) ; I2 , (C2 , A2) ; … ; IL



Temporal encoding: timestamp t replaces sequence position p in RoPE; the causal mask preserves order.

Context-conditioned prediction heads



Buckets follow historical CTR patterns. K=2, positions 1 to 4 and 5 or later, simplifies auction integration.

Training-only auxiliary tasks: click type, landing-page dwell time, and impression duration.

1 **Predict**

The model produces a CTR estimate for every position group.

2 **Train**

Example: position 3 selects the positions 1 to 4 estimate for the click loss.

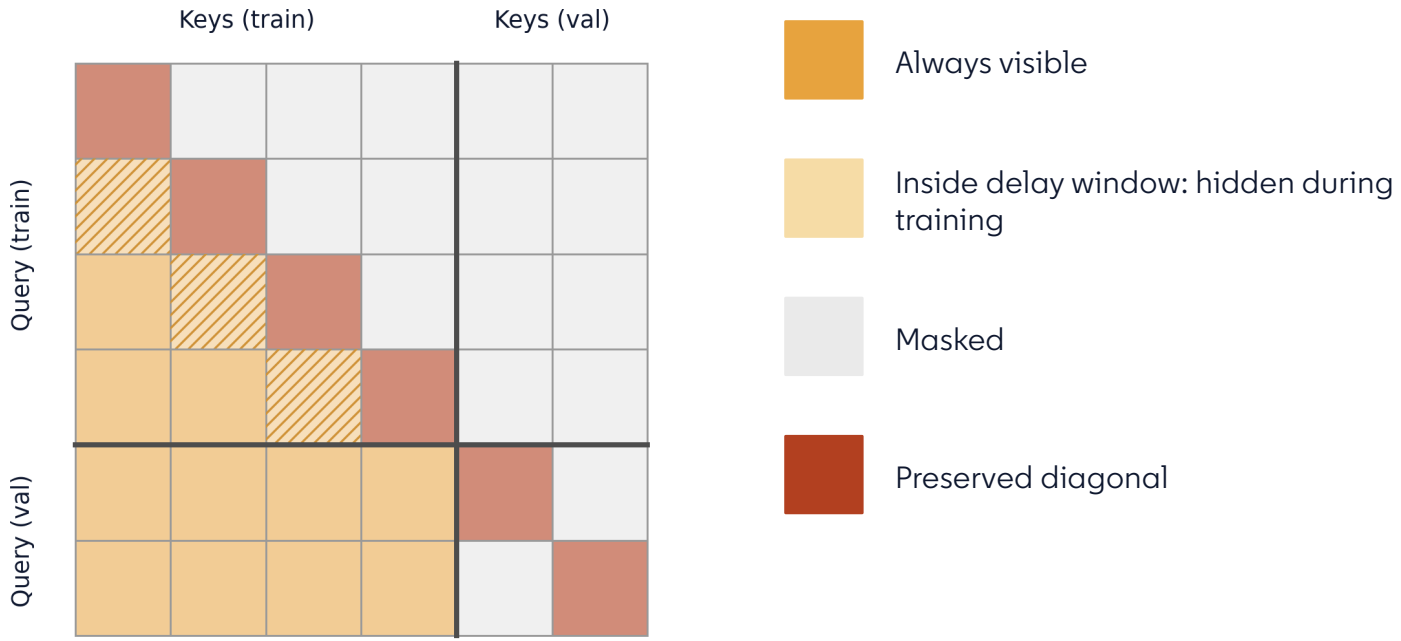
3 **Serve**

Return both estimates; the auction for the slot uses the value for its position group.

Observed position selects the training head. It is not used as an input for the current impression.

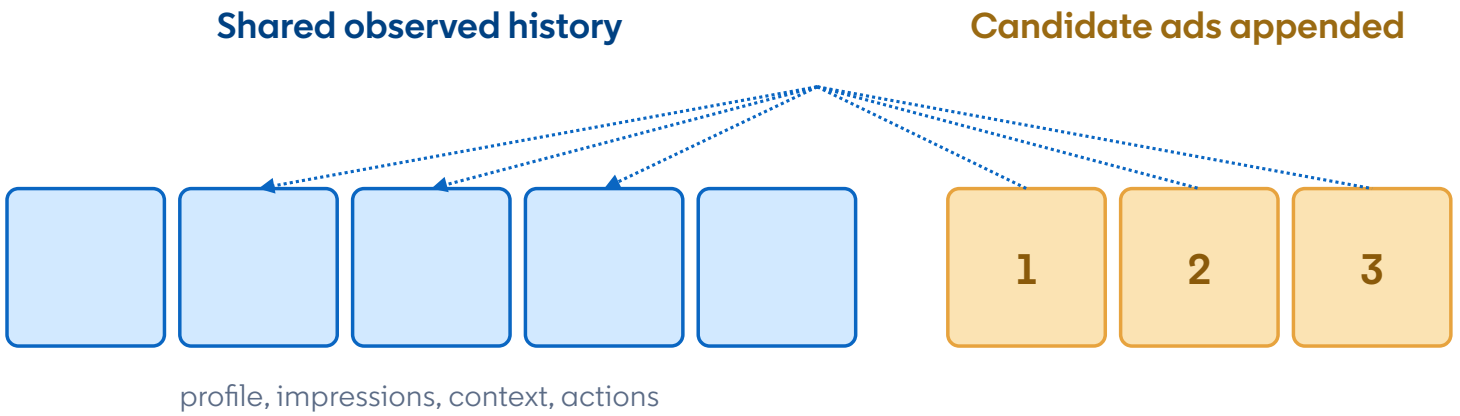
Matching training information to online observability

- Recent events may not be in the online sequence store yet, but are present in training sequences.
- Events inside the observability delay window are hidden during training.
- This prevents inflated offline metrics and reliance on unavailable history.
- The mask is time-based; it does not require a hard session identifier.



Striped cells are causally available but deliberately hidden during training to match online observability.

Scoring many ads against one shared user history



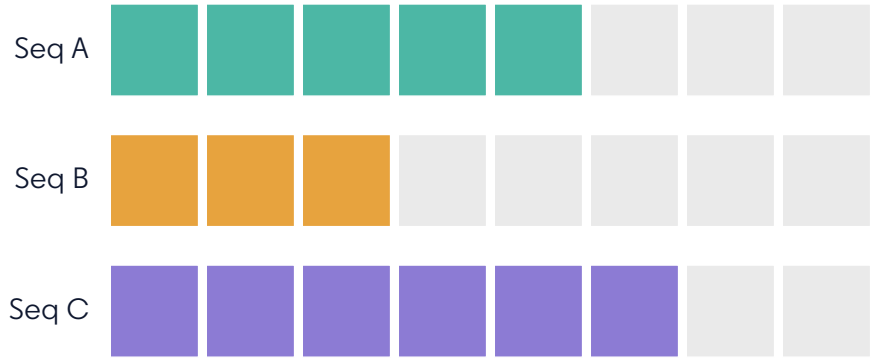
Each candidate attends to the shared history; all are scored in one forward pass.

Custom FlashAttention: 3× faster than FMHA. Work is $L^2/2 + LN$ instead of $(L + N)^2$, so candidate scaling is linear.

Candidates cannot attend to one another

From padded pointwise records to packed user sequences

Padded batch



Gray cells are padding

User-centric sequences retain 12 high-utility signals instead of more than 100 pointwise features.

Packed buffer

Sequences stay contiguous; prefix-sum offsets preserve boundaries.

98%
data footprint reduction

~4×
effective batch size

EXPERIMENTAL SETUP

Data, model, training, and serving configuration

Data

- Approximately one year of interactions
- Final day used for validation
- One-hour observability delay

Model

- 8 transformer layers
- 4 attention heads
- Model dimension 352
- Maximum sequence length 4096

Training

- NVIDIA H200 GPUs
- PyTorch HSDP
- FlexAttention
- Fixed packed-token budget

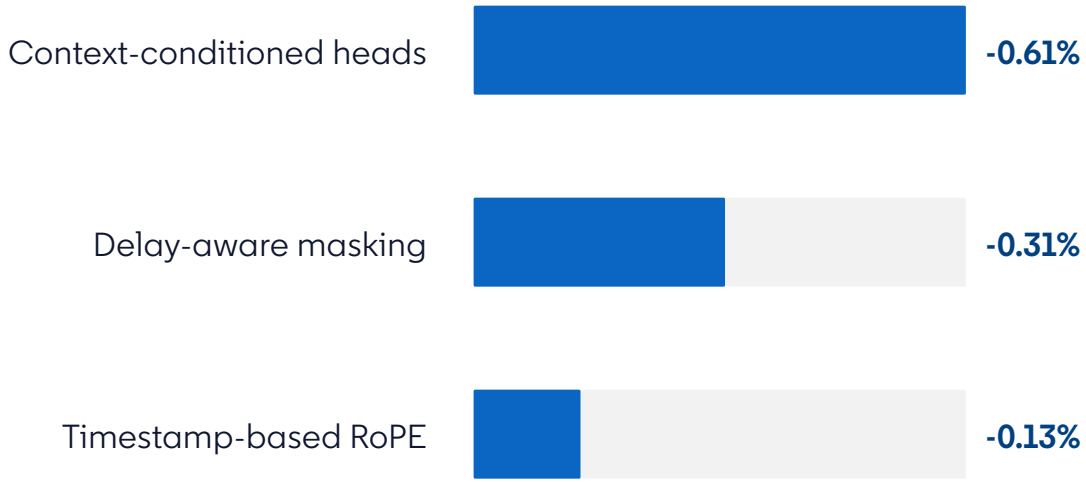
Serving

- Multiple candidates per request
- Custom FlashAttention kernel
- p99 latency below 50 ms
- Approximately 330 requests per second per GPU

OFFLINE EVALUATION

Component ablations and backbone comparison

Relative AUC change when each component is removed



Backbone comparison

Hold CADET components fixed and swap only the sequence backbone.

Model variant	AUC
HSTU-Base	0.8497
HSTU-Large	0.8505
CADET-HSTU*	0.8535
CADET-Transformer*	0.8534
Full CADET	**0.8554**

*Both backbone-swapped variants exclude delay-aware masking because the released HSTU Triton kernel does not readily support this mask.

ONLINE EVALUATION

A/B test against the production LiRank baseline

+11.04%

Relative CTR lift
Statistically significant

-10.9%

Cost-per-click
More clicks per budget

~Neutral

Revenue
Advertisers generally spend fixed budgets

Under auto-bidding, advertisers generally spend their budgets. Better relevance produces more clicks and lower CPC without materially increasing total spend.

CADET now serves main LinkedIn homefeed sponsored-updates traffic.

SUMMARY

CADET adapts a unified transformer for production ads CTR prediction

1

Post-scoring context

Context-conditioned heads produce predictions for position groups without requiring current position as an input.

2

Reliable temporal modeling

Timestamp-based RoPE models elapsed time, and delay-aware masking aligns training with serving.

3

Production deployment

Packing and sparse shared-context attention enable practical training and serving at LinkedIn scale.

Future work: cross-domain modeling, extending to other ad placements, and incorporating additional contextual signals.

Neil Daftary | LinkedIn Ads AI | AdKDD 2026