Liu Yingzuo


ML Engineer | Ad Tech & Recommendation Systems
FastAPI
Redis
Bidding Strategy
Ranking
PyTorch
Deep Learning
MLOps
Mechanism Design
Machine Learning
Kubernetes
Multi-Stage Pipeline
Budget Pacing
Causal Inference
SQL
A/B Testing
FAISS
Python
Recommendation
CTR/CVR Prediction
Airflow

Overview of My Projects

Projects Focus — Bridging Machine Learning, Mechanism Design, and Intelligent Bidding in Advertising Systems

Project I – Unified CTR/CVR Prediction for Smart Bidding

Goal: Improve ad relevance and conversion prediction while ensuring real-time inference efficiency for oCPM bidding.

Project II – Multi-Agent Bidding Strategy Simulation Platform

Goal: Design a digital sandbox to test and analyze new bidding mechanisms and strategies before deployment.

“From prediction → decision → simulation”

Project I — Driving the Core of Intelligent Advertising Systems: From eCPM Optimization to the Smart Bidding Flywheel

Background and Positioning

In modern advertising systems, the balance among platform revenue, advertiser ROI, and user experience is determined by a single core formula:

$$eCPM = pCTR \times pCVR \times Bid$$

Where:

- **pCTR / pCVR**: Reflects the predictive power of the model
- **Bid**: Represents the intelligent bidding strategy
- **eCPM**: Serves as both the ranking metric and the unified measure of system value

My research and engineering work center on how to simultaneously optimize both "prediction" and "bidding strategy," making intelligent advertising systems both accurate and efficient.

2. Experimental Objectives and Data Construction

Project Goal
Under a constrained hardware environment (RTX 4050 16GB + 400GB SSD), this project integrates **DeepFM + LightGCN + PLE** on public datasets to:

- Improve **joint prediction accuracy of CTR and CVR**
- Enhance the **ranking metric (NDCG)**
- Support **real-time Smart Bid computation** in the oCPM system

3. Data Pipeline

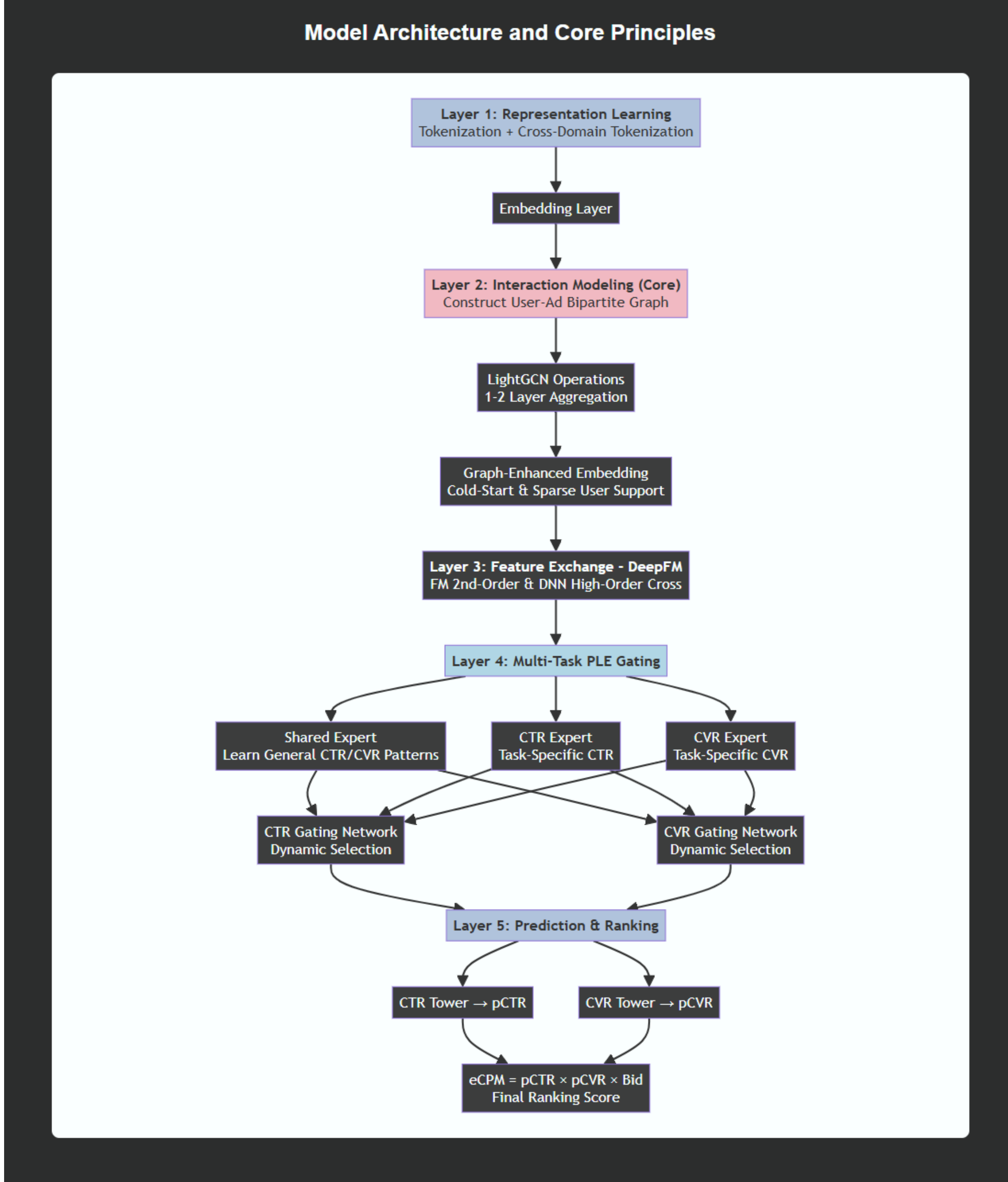
Using a **Criteo Kaggle subset (1M rows)** for experimental validation

Pipeline:

1️⃣ Load criteo → sample 1M rows
2️⃣ Split dataset: train/val/test = 8/1/1
3️⃣ Define CTR = click
4️⃣ Define CVR = 5% of the samples with click=1 are labeled conversion=1
5️⃣ Dense features → log1p + normalization
6️⃣ Sparse features → hash encoding (2^16)
7️⃣ Build user–ad bipartite graph → LightGCN input

4. Model Architecture

Model Structure: DeepFM + LightGCN + PLE



5. Training Configuration and Hyperparameter Strategy

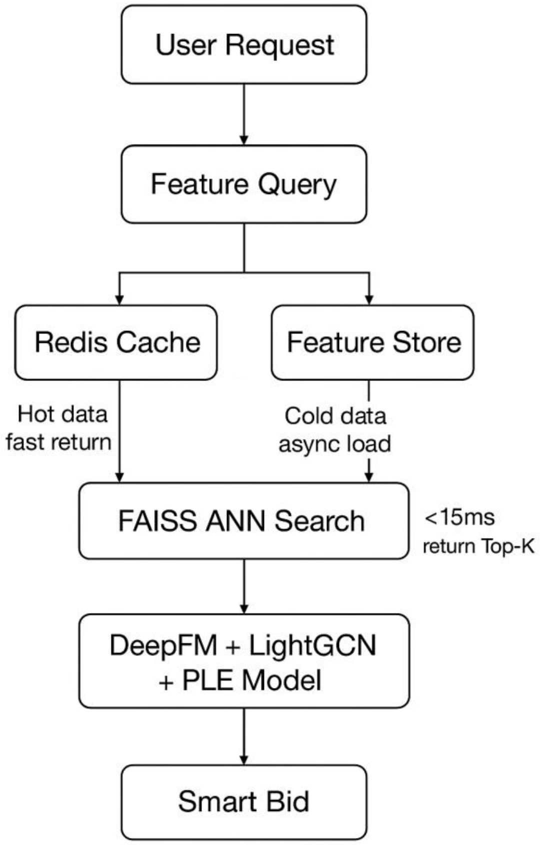
Parameter	Range / Selection	Description
embedding_dim	16	Friendly to memory
deep_layers	[128,64]	Balanced capacity and memory
experts_per_task	2	PLE lightweight design
lightgcn_layers	2	Control computation
batch_size	64	Combined gradient accumulation
fp16	True	Mixed precision
optimizer	Adam (lr=1e-3, 2e-3, 5e-4)	Supports warmup
epochs	10-20	Early stopping

Tuning Insights:
LightGCN significantly improves performance on **cold-start and long-tail samples**
PLE mitigates the **CTR/CVR "see-saw effect"**, stabilizing multi-task optimization
fp16 + embedding cache reduces GPU memory usage by ~30%

6. High-Performance Inference Architecture Validation

Why?

In an oCPM production system, each ad request must complete a **bidding decision within 50ms**. Thus, the model must not only be **accurate**, but also **fast and stable**.



Inference Optimization Components

Module	Technique	Outcome
Redis	In-memory storage with vector indexing	Consistent latency ~70%
FAISS	IVF + PQ nearest neighbor search	Latency within 15ms
ONNX Runtime	Graph optimization + batch processing	40% latency reduction
P99 Latency	<50ms	Fixed upper bound on latency requirement

“Make the Smart Bidding model not only accurate — but also fast and robust.”

7. Experimental Results

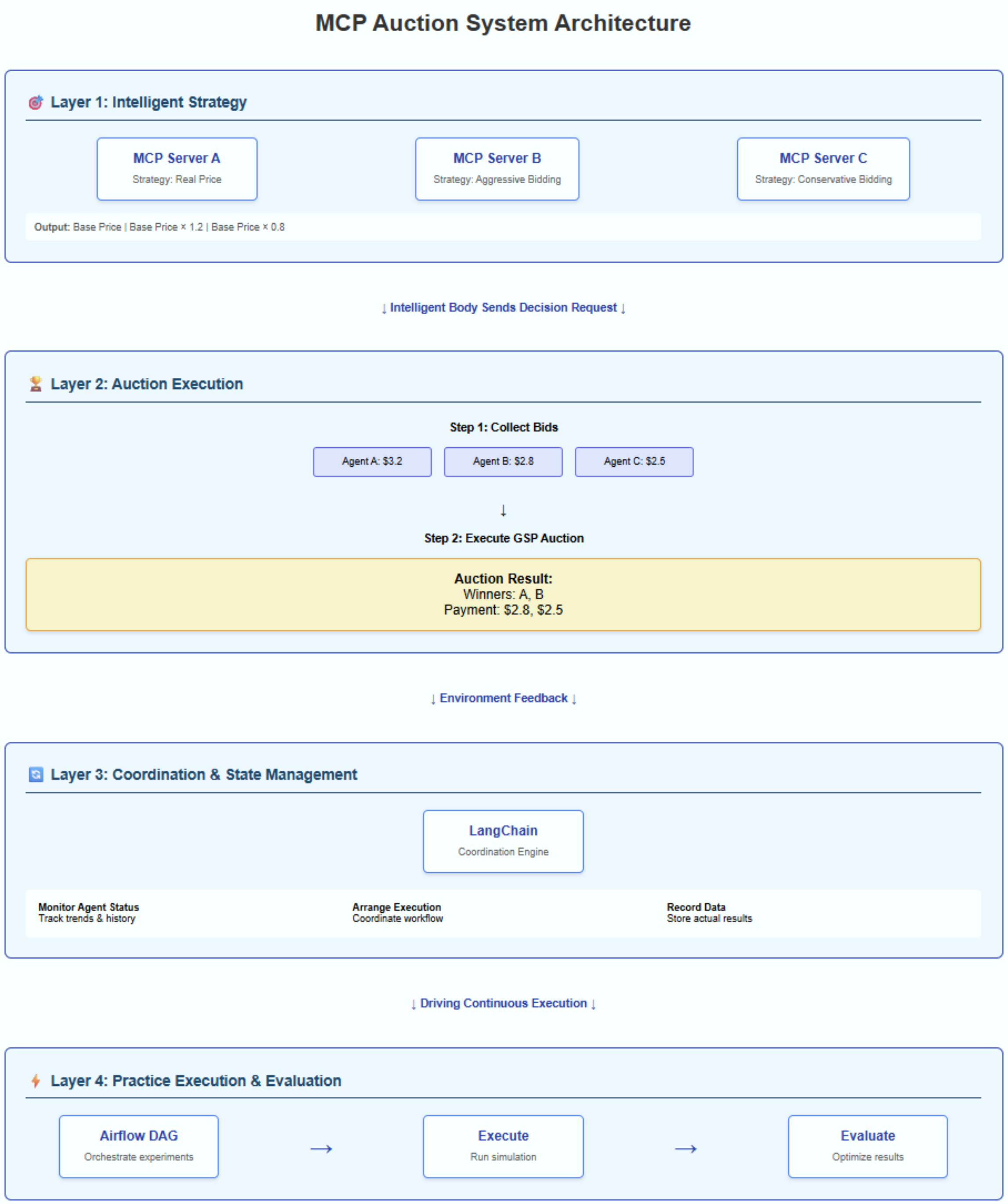
Model	AUC(CTR)	AUC(CVR)	NDCG@10
DeepFM	0.768	0.742	0.69
DeepFM+PLE	0.783	0.757	0.72
DeepFM+LightGCN+PLE	**0.793**	**0.764**	**0.74**

8. Reflections and Trade-offs

Dimension	Scheme	Advantages	Disadvantages	Decision
Model Complexity	LightGCN+PLE	High efficiency	High memory	✓ Keep offline
Search Mechanism	FAISS+Redis	Scalable	Memory bottleneck	✓ Hybrid mode
Multi-task Weight	α=0.3	Resolve conflicting gradients	Slow convergence	✓ Fixed optimal point

Project II — Simulating Bidding Strategies to Master the Complexity of Game Dynamics

- **Pain Point:**
  Directly deploying new oCPM bidding strategies or auction mechanisms via A/B testing is **high-risk and time-consuming**.
- **My Solution:**
  Build a **digital sandbox** for oCPM bidding strategies and mechanism design.



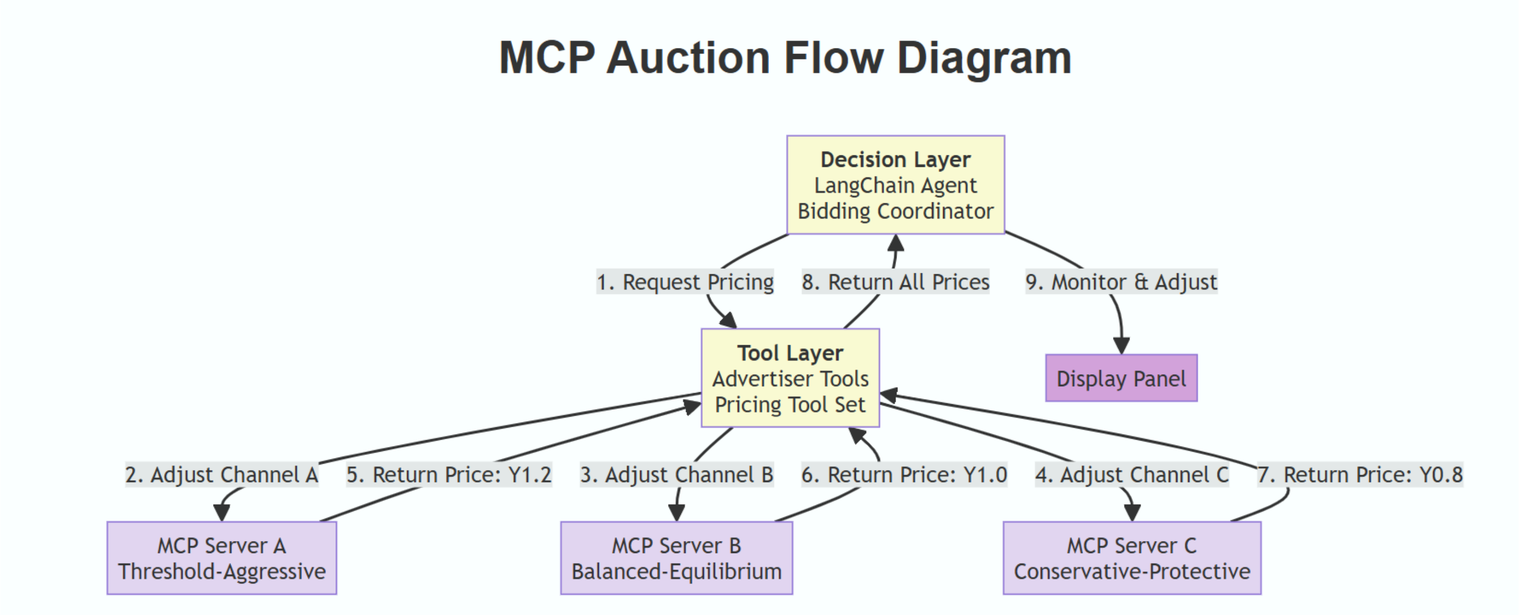
Implementing "Strategy-as-a-Plugin" with MCP

Core Idea of MCP:
Encapsulate complex capabilities as **independent, pluggable servers**.

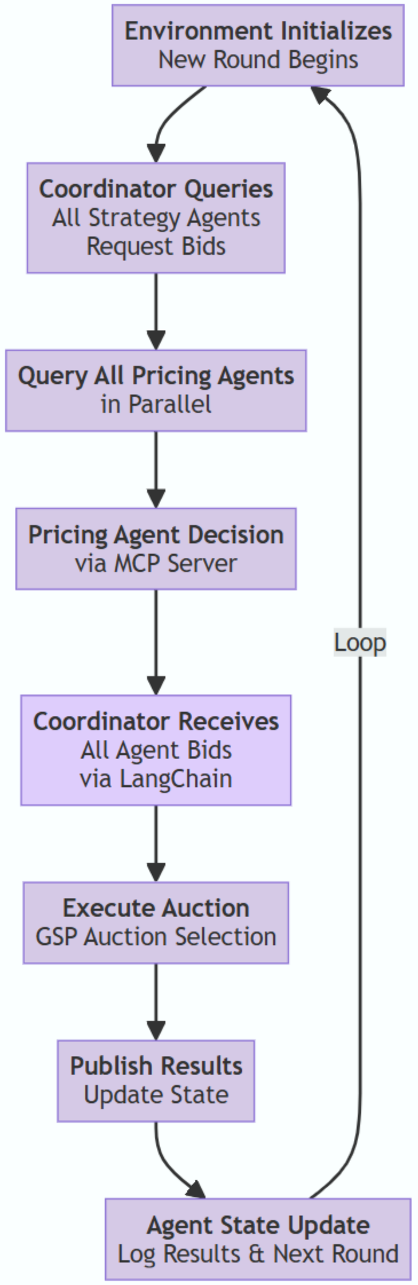
My Implementation:
Each advertiser's bidding strategy is an **independent MCP server**, achieving complete decoupling.
Advantages:
- **Hot-swappable:** Strategy updates without changing core code
- **Language-agnostic:** Strategies can be implemented in any language
- **Unified protocol:** All strategies share the same invocation interface

Building an Intelligent Agent Orchestration Hub with LangChain

Interaction Workflow Explanation
- **LangChain Agent:** Acts as the coordinator, deciding when and which tool to invoke
- **LangChain Tools:** Serve as the adaptation layer, providing a simple interface to call MCP servers
- **MCP Servers:** Act as domain experts, executing specific and complex bidding strategies



Core Workflow — Agent Interaction

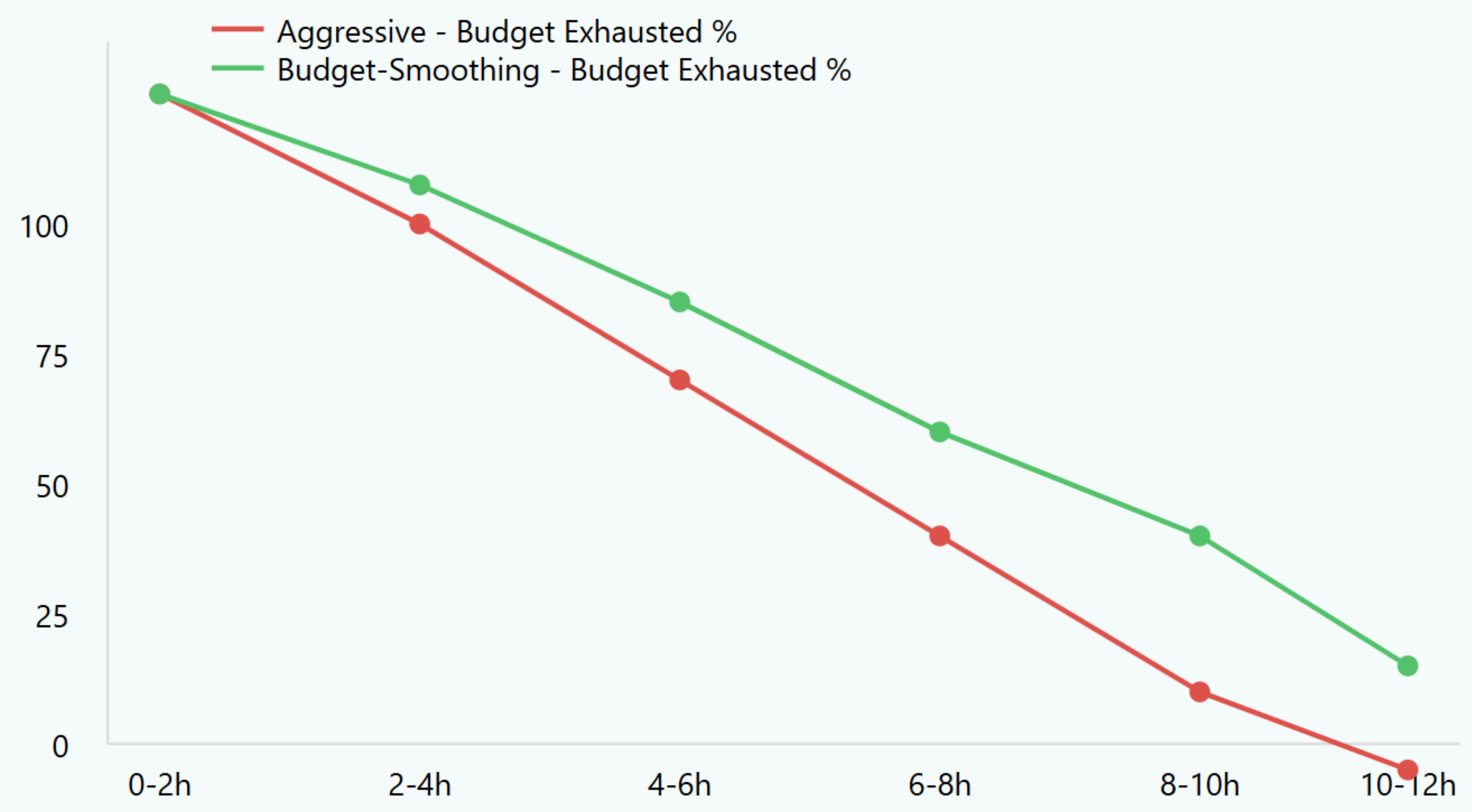


oCPM Strategy Analysis

Based on system design insights into composable strategy and risk early warning

Insight 1: Budget Management | Insight 2: Auction Fairness

Aggressive vs Budget-Smoothing Strategy Comparison

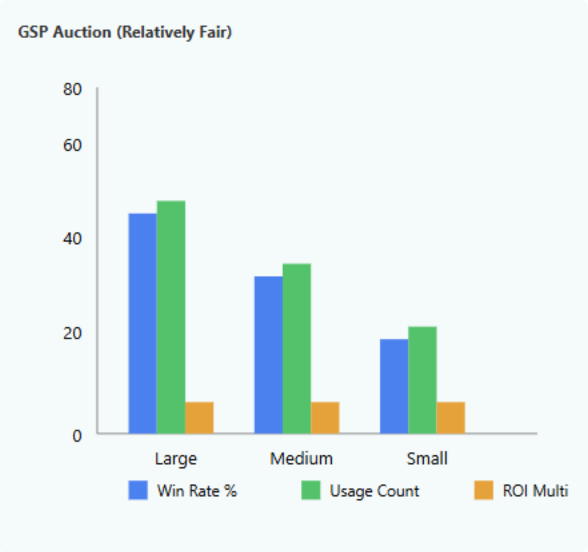


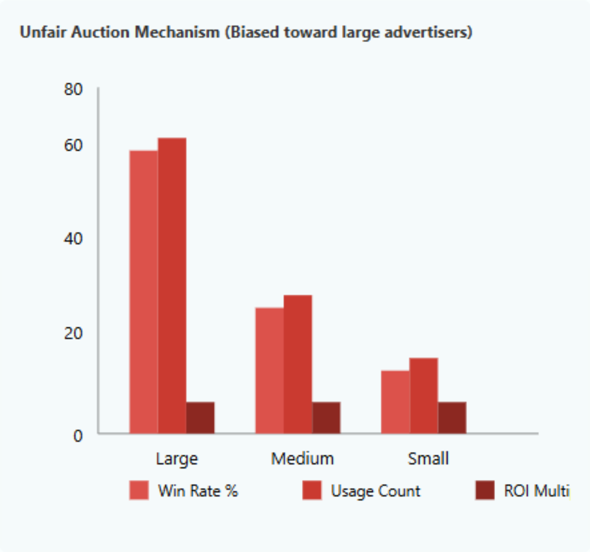
Key Finding: Aggressive bidders gain early exposure but deplete budgets too quickly, losing competitiveness in later hours. Budget-smoothing strategies maintain steadier exposure throughout the period and achieve higher long-term returns. This demonstrates that smart budget control is essential, not optional.

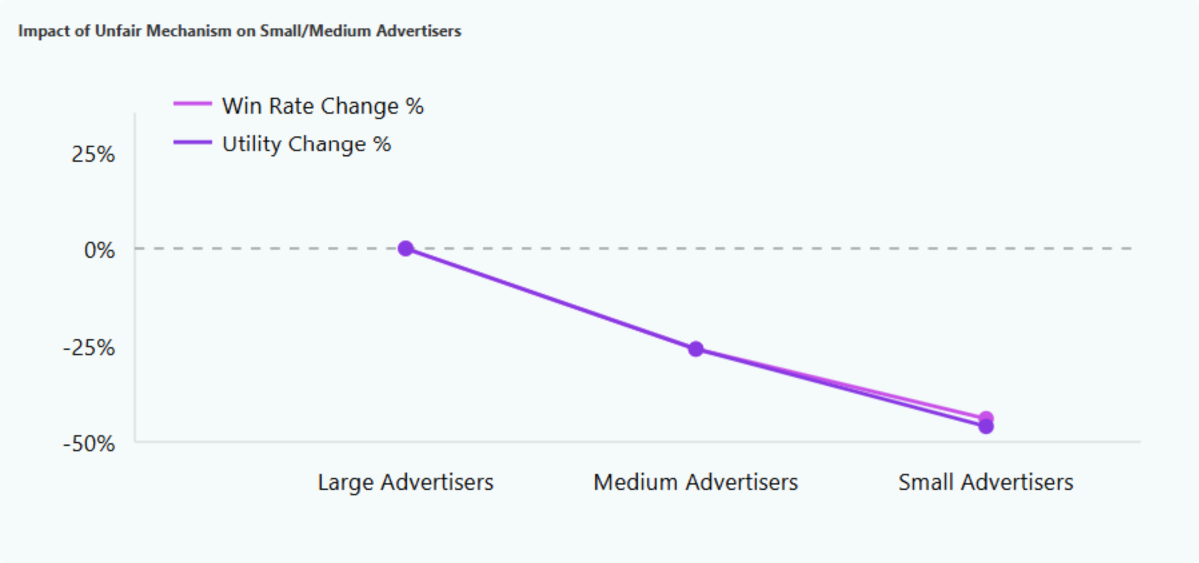
oCPM Strategy Analysis

Based on system design insights into composable strategy and risk early warning

Insight 1: Budget Management | Insight 2: Auction Fairness







Key Finding: When simulating a biased auction mechanism that favors large advertisers, small and medium advertisers' win rates and utilities dropped sharply. The system provides early warnings about market imbalance before real deployment — protecting platform health and ensuring ecosystem sustainability.

From Prediction to Decision — Towards an Intelligent Advertising System

- **Core Insight**

Bridged *machine learning–based prediction* with *game-theoretic bidding optimization*

Built a full-stack research pipeline: **Prediction → Decision → Simulation**

- **Practical Value**

Improved CTR/CVR estimation stability under constrained hardware

Enabled sandbox verification of new bidding strategies before deployment

- **Future Direction**

Integrate *AI Agents* for adaptive bidding and creative generation

Move toward an end-to-end intelligent advertising ecosystem that learns, adapts, and optimizes in real time

Thank you