DCN^2: Interplay of Implicit Collision Weights and Explicit Cross Layers for Large-Scale Recommendation

Blaž Škrlj, Yonatan Karni, Grega Gašperšič, Blaž Mramor, Yulia Stolin, Martin Jakomin, Jasna Urbančič, Yuval Dishi, Natalia Silberstein, Ophir Friedler, Assaf Klein

Teads

August 2025

DCN^2

1. Motivation and overview
2. Architecture improvements
3. Offline evaluation
4. Online evaluation
5. Scaling and productization

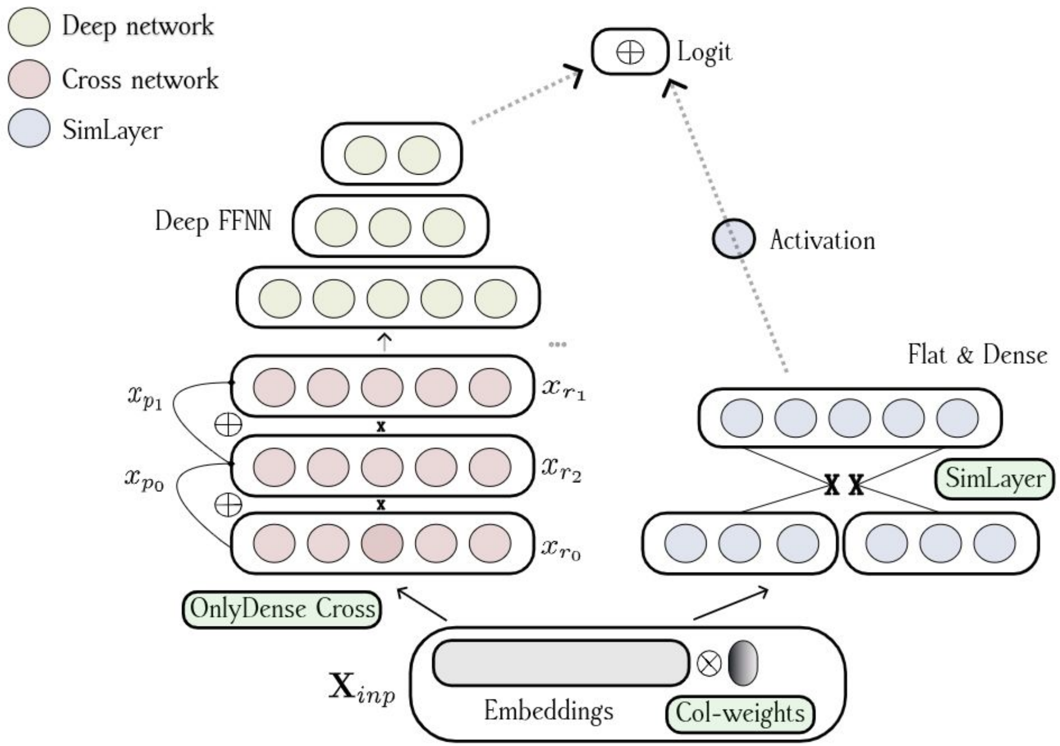
Motivation

1. Building reliable CTR/CVR models at scale is a **challenging task**
   a. Models operate on **streams of data**
   b. Item collisions can lead to **performance decay**
   c. Item interactions of **different order** contribute to final prediction

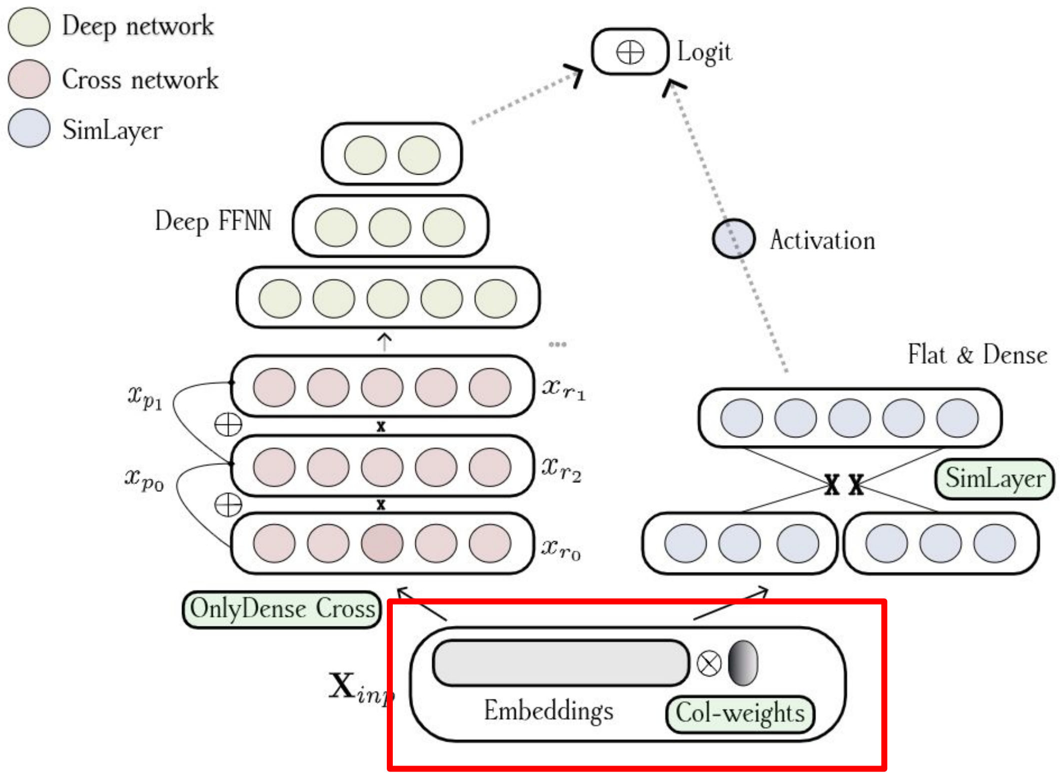
2. Existing **DCNv2** addresses many, yet not all of these challenges

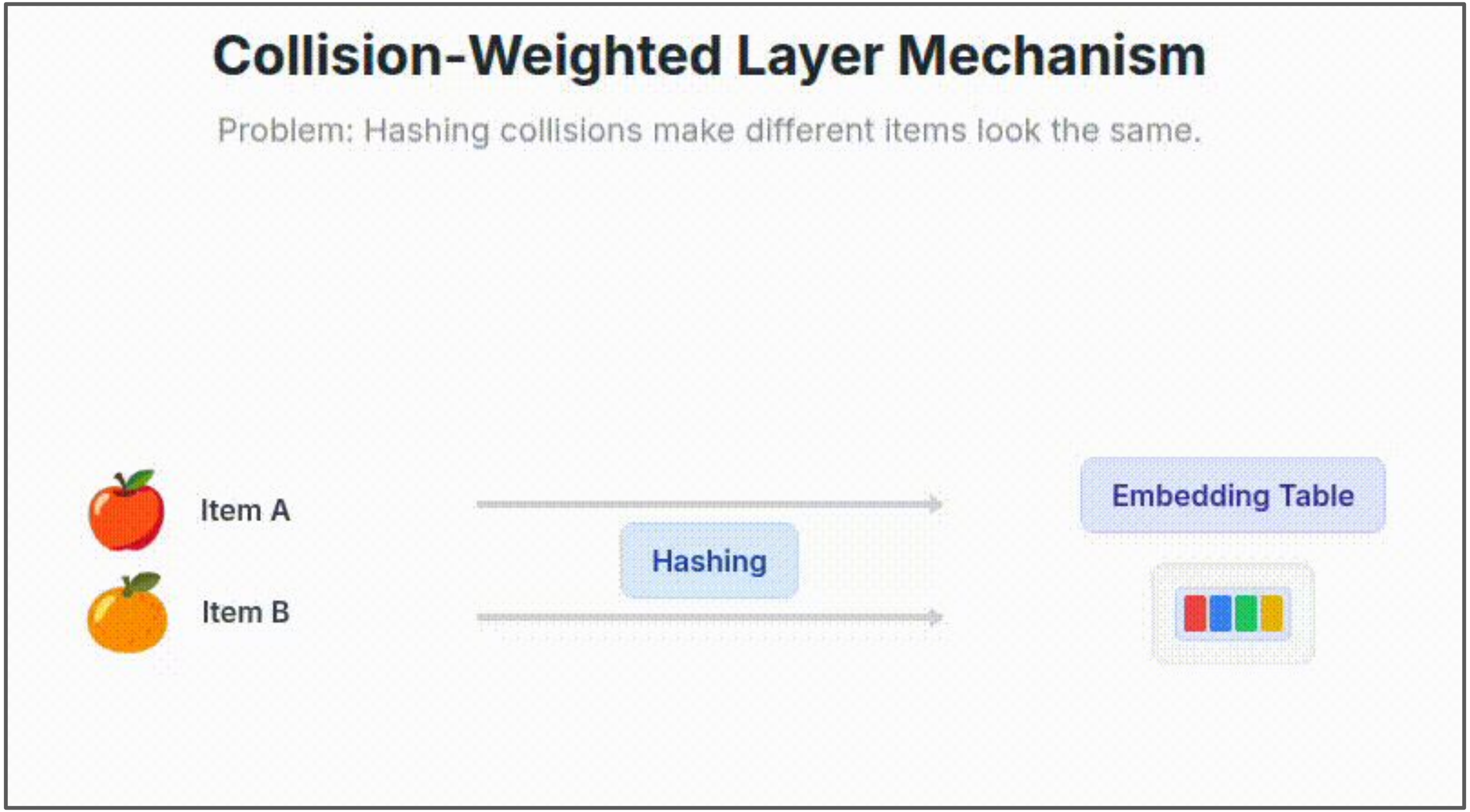
3. We systematically investigated possible **improvements** at different levels of the architecture, and deployed the result at scale

Architecture



Architecture




Collision-Weighted Layer Mechanism
Problem: Hashing collisions make different items look the same.
Item A
Item B
Hashing
Embedding Table

Architecture - *collision weights*

Step0:

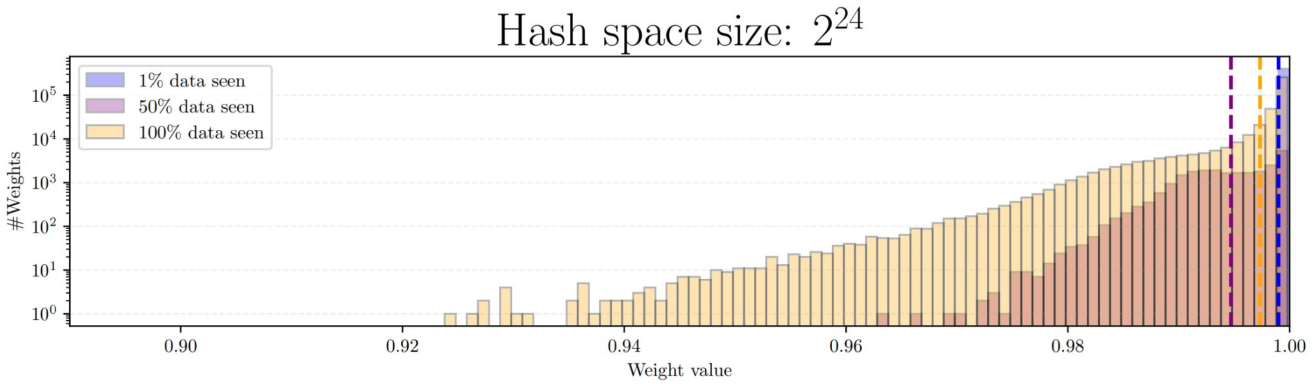
$$\mathrm{X}_{ec} = \begin{cases} \mathrm{X}[:, 1:d] = \mathrm{X}[:, 1:d]; -\omega \leq \mathcal{N}(\mu, \sigma^2) \leq \omega \\ \mathrm{X}[:, d+1] = \mathbb{1} \end{cases}$$

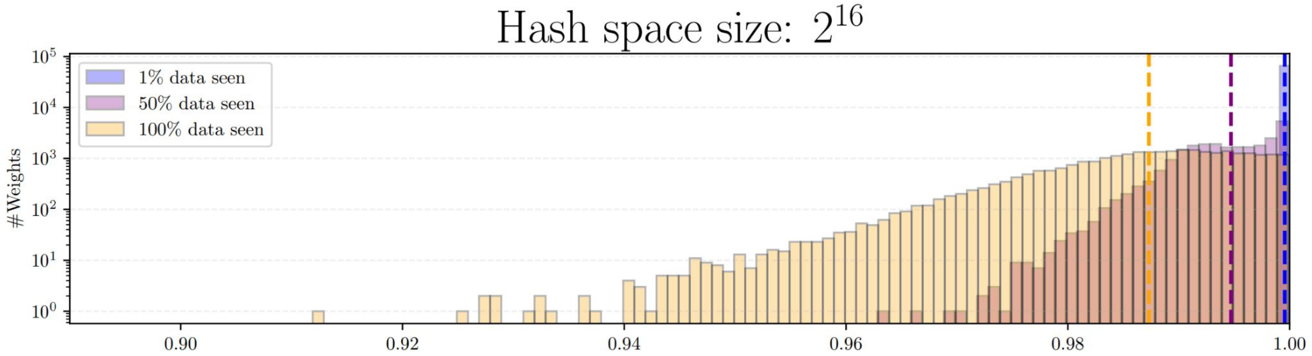
Step1:

$$\mathrm{X}_{\text{inp}} = \mathrm{X}_{ec}[:, \ldots d] \odot \mathrm{X}_{ec}[:, d+1]$$

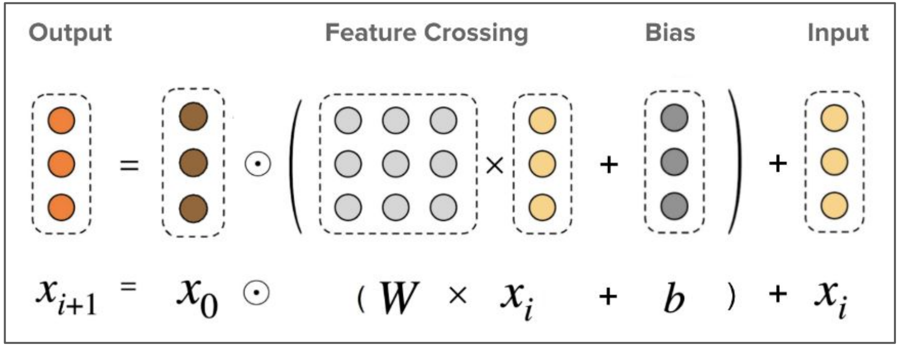
Addressed aspect: **Resilience to collisions**

Collision weight values, visualized



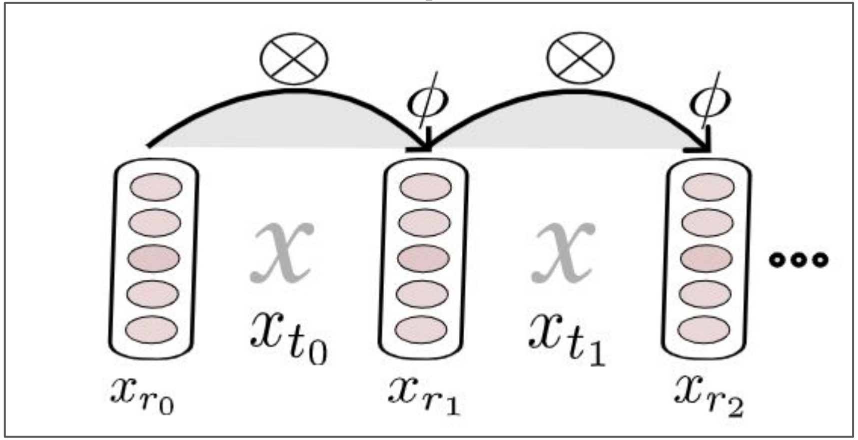


Architecture - *onlydense* layer



DCNv2 (Wang et al.)

$$x_t = \alpha(\mathbf{W} \cdot \mathbf{x} + b_0)$$
$$x_r = x_t \odot \mathbf{x} \cdot \phi.$$

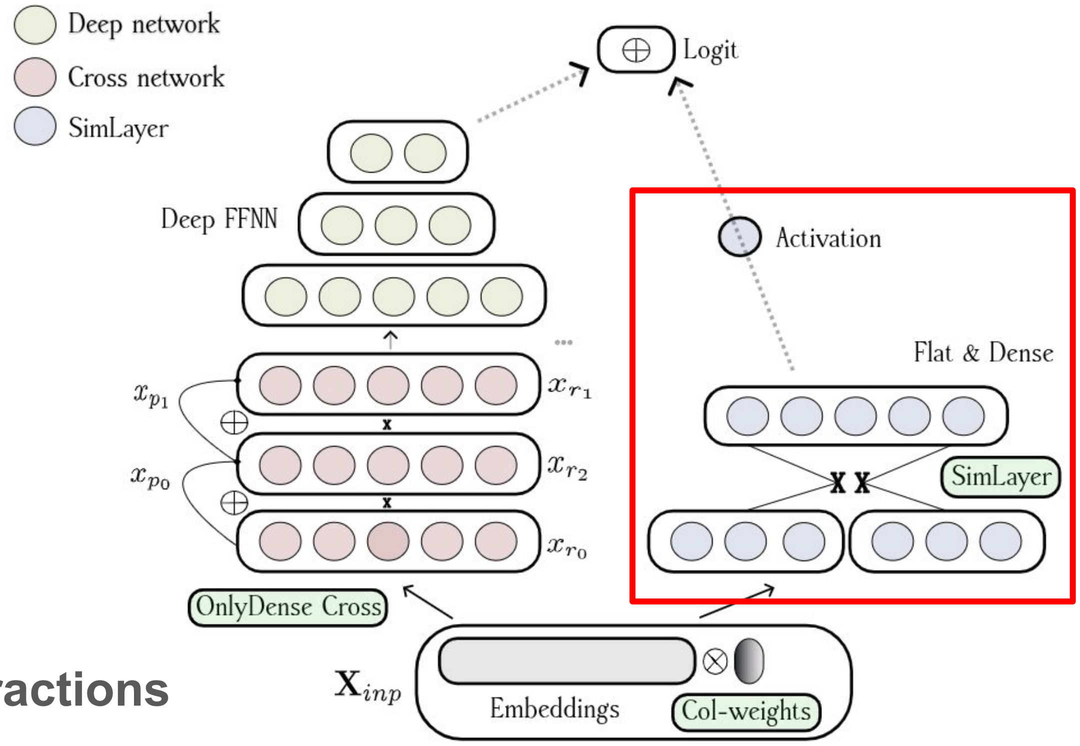


This work

Addressed aspect: **Info loss in Cross**

Architecture - similarity "kernel"

$$\hat{y}_{\text{sk}} = \alpha\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{w}_{k'(i,j)}\left(\sum_{k=1}^{m}\mathbf{e}_{ik}\cdot\mathbf{e}_{jk}\right)+b\right)$$



Addressed aspect: **Pairwise interactions**

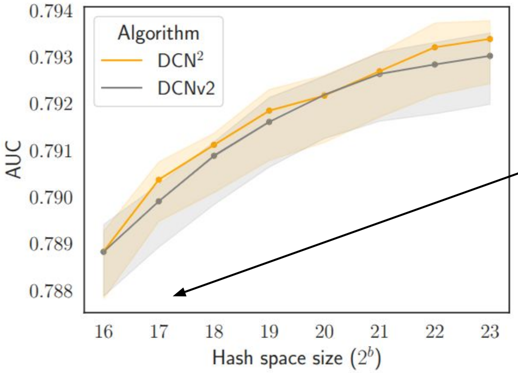
Benchmarks - offline

Avazu					
Algorithm	avg	median	max	min	std
FM	0.7748	0.7746	0.8304	0.7237	0.0180
deepFM	0.7812	0.7814	0.8350	0.7230	0.0183
DCNv2	0.7826	0.7832	0.8351	0.7244	0.0183
DCN^2	0.7846	0.7846	0.8387	0.7284	0.0183
DCN^2-simk	0.7824	0.7826	0.8354	0.7242	0.0183

Criteo					
Algorithm	avg	median	max	min	std
FM	0.7834	0.7831	0.8166	0.7617	0.0064
deepFM	0.7906	0.7904	0.8214	0.7716	0.0063
DCNv2	0.7922	0.7918	0.8229	0.7730	0.0063
DCN^2	0.7933	0.7930	0.8231	0.7751	0.0063
DCN^2-simk	0.7922	0.7919	0.8233	0.7738	0.0063

KDD2012					
Algorithm	avg	median	max	min	std
FM	0.7547	0.7545	0.8336	0.6769	0.0201
deepFM	0.7719	0.7677	0.8709	0.7058	0.0260
DCNv2	0.7730	0.7684	0.8731	0.7133	0.0265
DCN^2	0.7747	0.7699	0.8735	0.7051	0.0272
DCN^2-simk	0.7733	0.7693	0.8761	0.7105	0.0266

iPinYou					
Algorithm	avg	median	max	min	std
FM	0.7521	0.7572	0.9955	0.3638	0.1049
deepFM	0.7669	0.7683	0.9961	0.4275	0.0997
DCNv2	0.7659	0.7667	0.9975	0.4333	0.1001
DCN^2	0.7561	0.7615	0.9984	0.3574	0.1023
DCN^2-simk	0.7467	0.7518	0.9980	0.4181	0.1043



More collisions -> superior performance

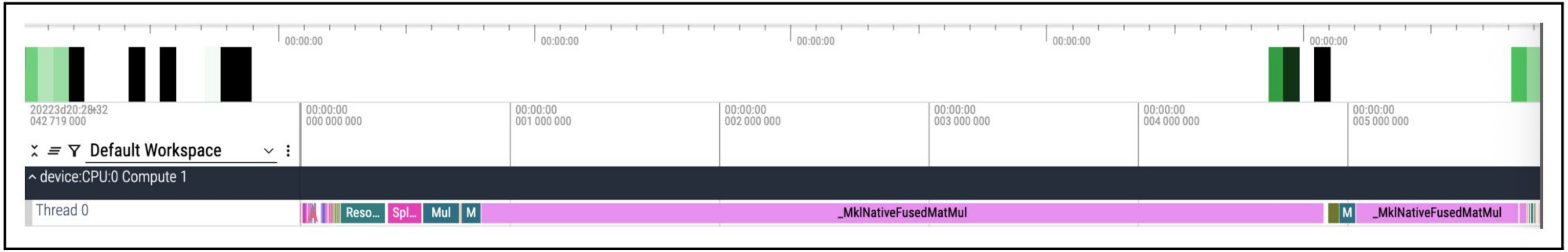
Taking it Online

Use case	Lift Offline (AutoML)	Lift Online (A/B)
CTR	0.0035 (RIG)	3.2% RPM
CVR	0.0010 (RIG)	4.2% swCR, 0.37% GR

Scaling DCN^2

- **Modernized Inference Stack:** Migrated the model to standard **TensorFlow/ONNX**, achieving peak performance with stock binaries after targeted kernel and graph optimizations
- **Optimized Execution:** Implemented a novel **"local fan-out" batching** strategy and optimized thread management, which cut p99 latency by **18%**
- **Final Performance:** Increased throughput **1.6x** via memory optimization (e.g., Jemalloc), delivering over **0.5 billion predictions per second** within strict latency limits

Profiling DCN^2 during inference



Conclusions

We introduced **DCN^2**, an improvement over DCNv2 that addresses issues with:

1. Item collisions
2. Information loss in Cross layers
3. Pairwise interactions being considered

Further work:

1. Can we use multiple embedding tables with different weight vectors?
2. Policy for explicit modulation of collision weights outside the model
3. Impact of hard resets at weight level to keep models fresh