

29 DE OCTUBRE DE 2024

# KuaiSim

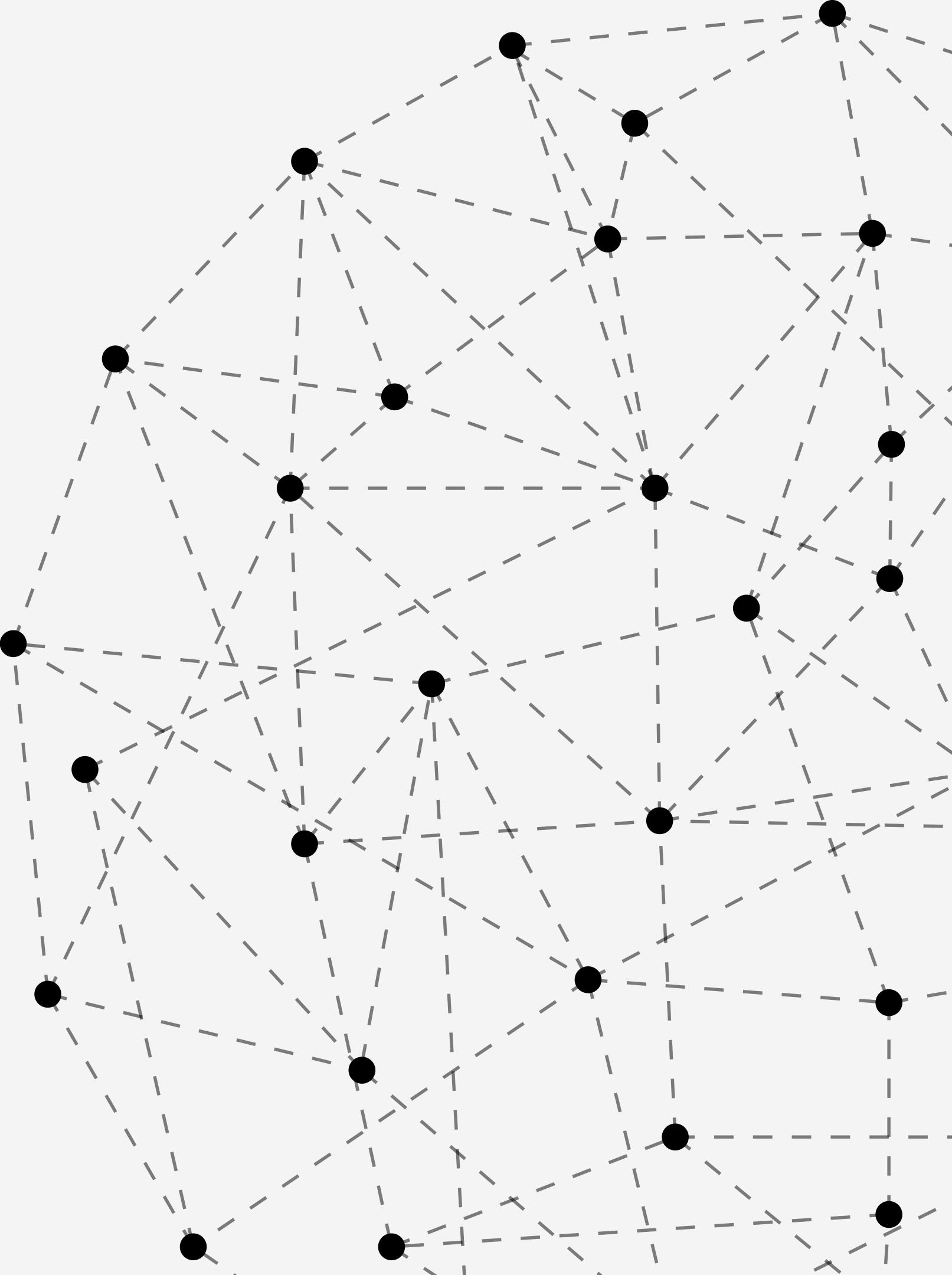
A Comprehensive Simulator for  
Recommender Systems

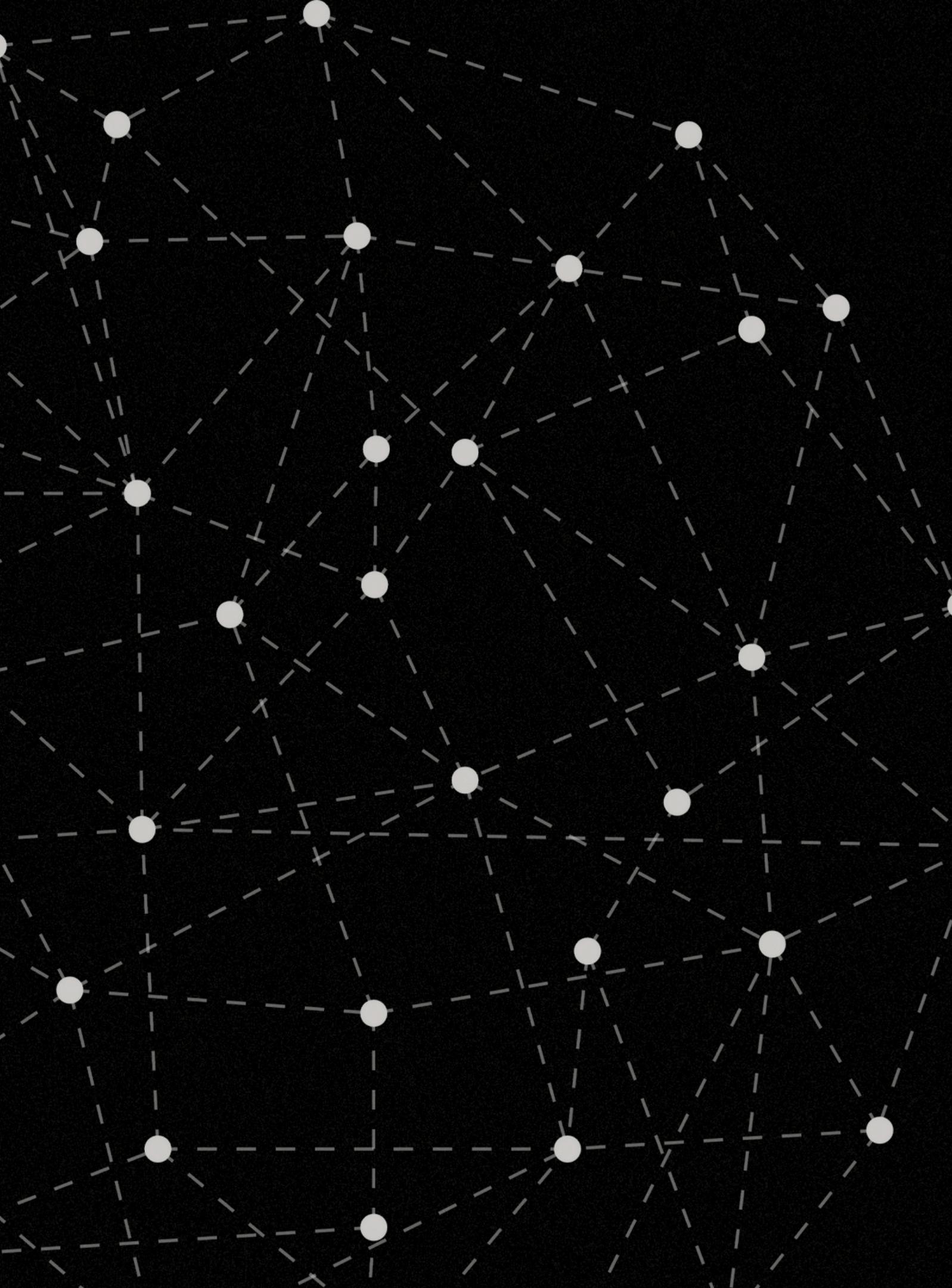
**Presentado por:**

Eduardo Contreras  
Gonzalo Fuentes  
Sebastián Salgado

**Profesor**

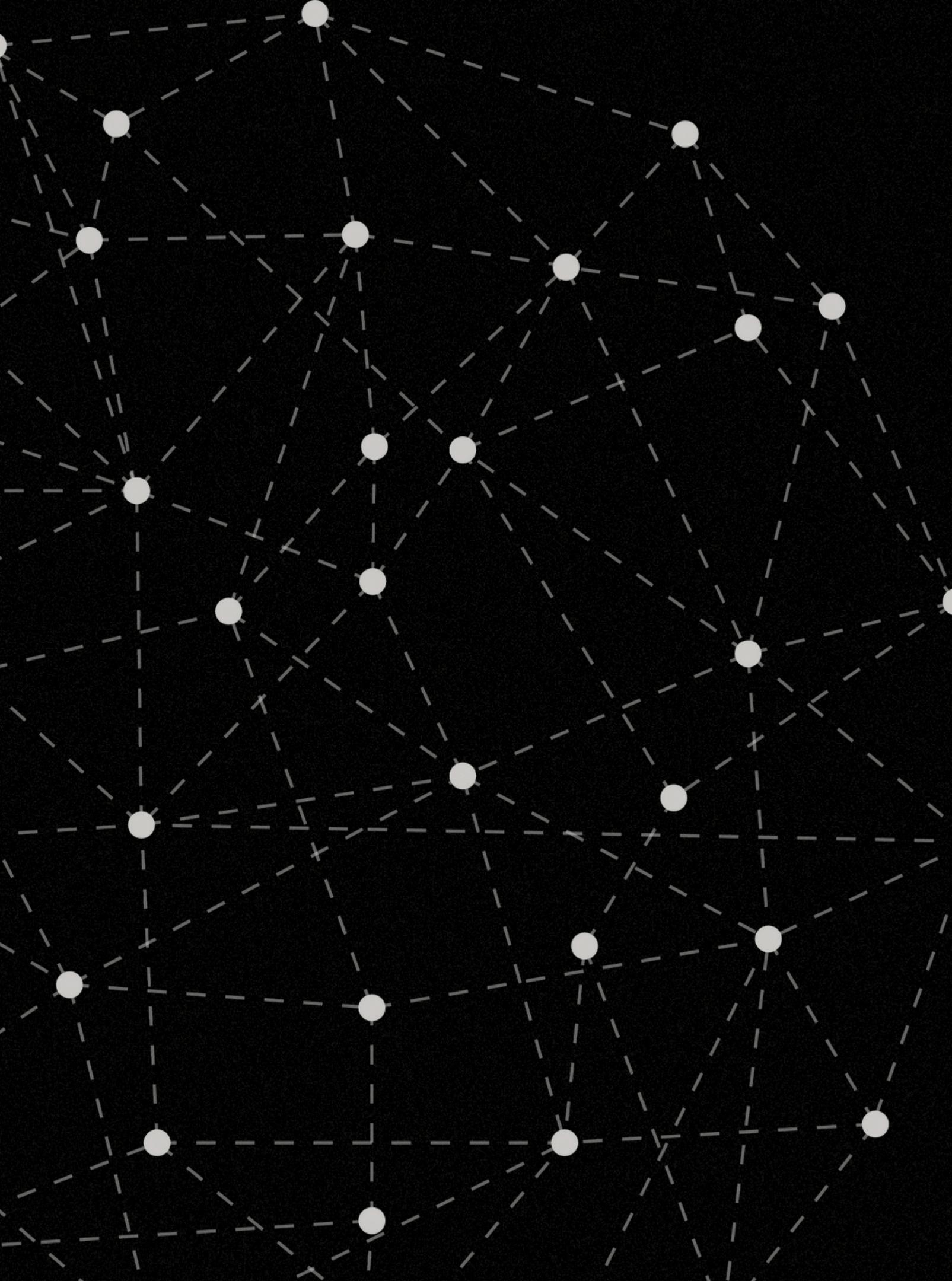
Denis Parra





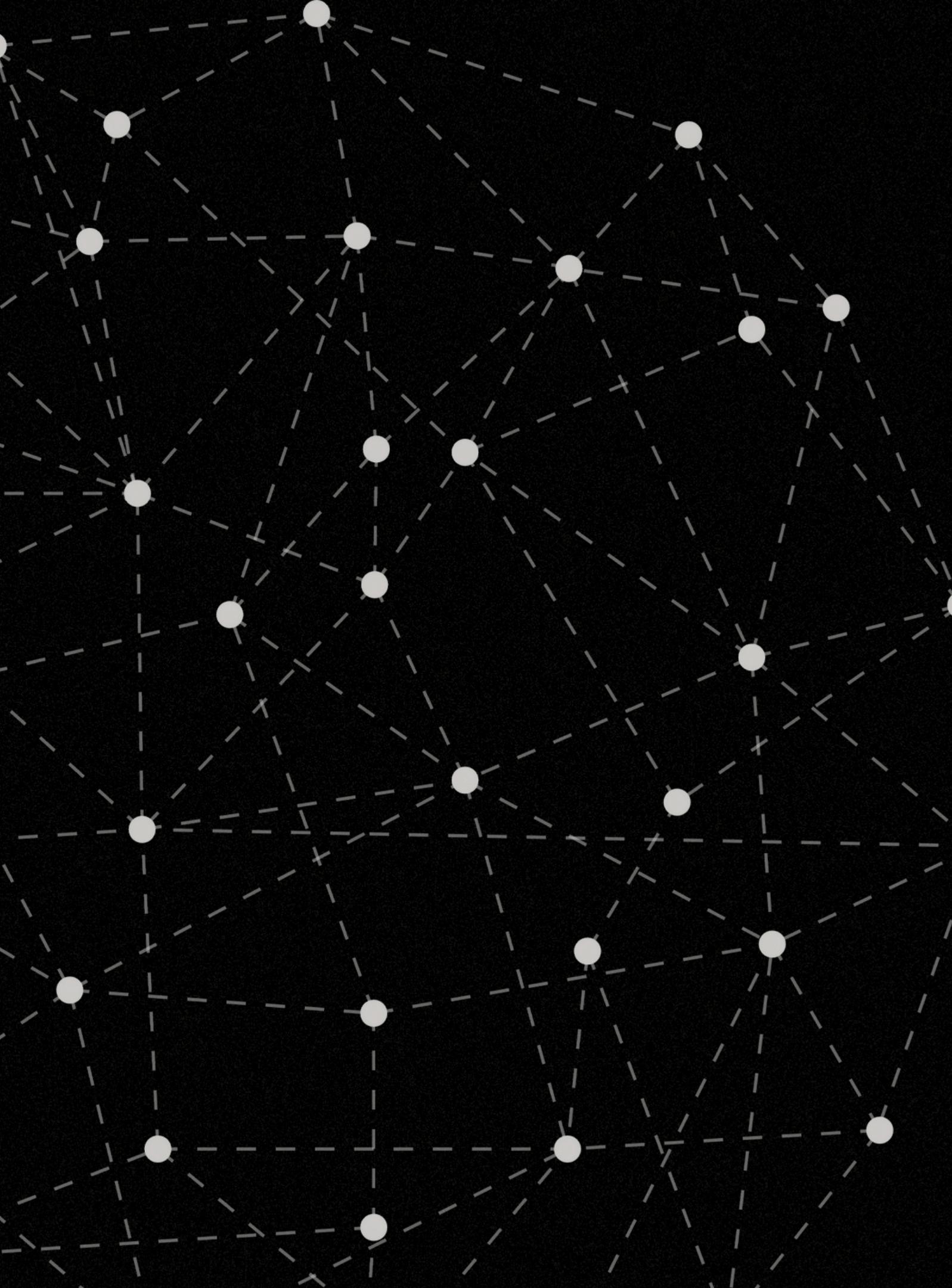
# Motivación

- Los sistemas recomendadores (RS) basados en Reinforcement Learning (RL) pueden aprender políticas que maximizan las recompensas acumuladas a largo plazo.
- En la teoría este approach es superior a learning to rank
- En la práctica **NO**



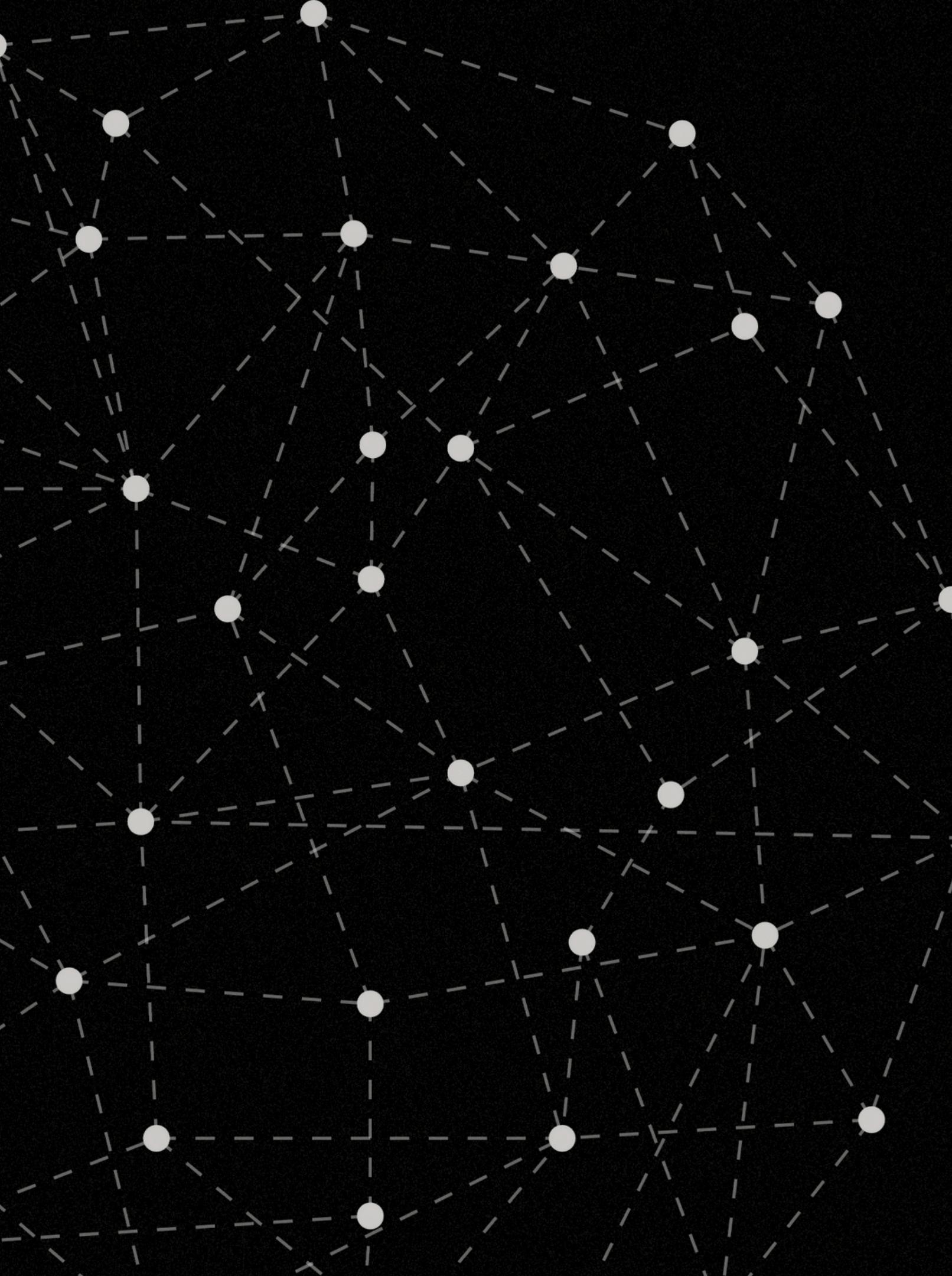
# ¿Por qué?

- Se entrena usando métodos offline (registros)
- Lo cual limita la capacidad de exploración
- También imposibilita la evaluación counter-factual que tienen los métodos online
- Pero los métodos online **tampoco son buena idea**



# ¿Por qué?

- Los A/B testing puede ser **costoso** (en tiempo y dinero)
- Dejar a un agente RL mal entrenado que interactúe con clientes puede llevar a una **mala experiencia**
- ¿Qué se hace entonces?



# Simuladores

- Permiten el entrenamiento y evaluación online
- No dañan la experiencia del usuario
- Permiten probar nuevos RS rápidamente.

# Trabajos Relacionados

## RECOGYM

Simulador para reacciones de usuario para políticas de recomendación arbitrarias. Se basa en un ciclo de recopilación de datos de rendimiento, desarrollando una nueva versión del modelo de recomendación, A/B testing y rolling out. Comparte similaridades con el setup de RL.

## VIRTUAL-TAOBAO

Simulador avanzado de comercio electrónico basado en datos reales de Taobao. Simula el comportamiento de usuarios en una plataforma de e-commerce, para mejorar esta simulación utiliza GAN-SD (GAN for Simulating Distributions) y MAIL (Multi-agent Adversarial Imitation Learning).

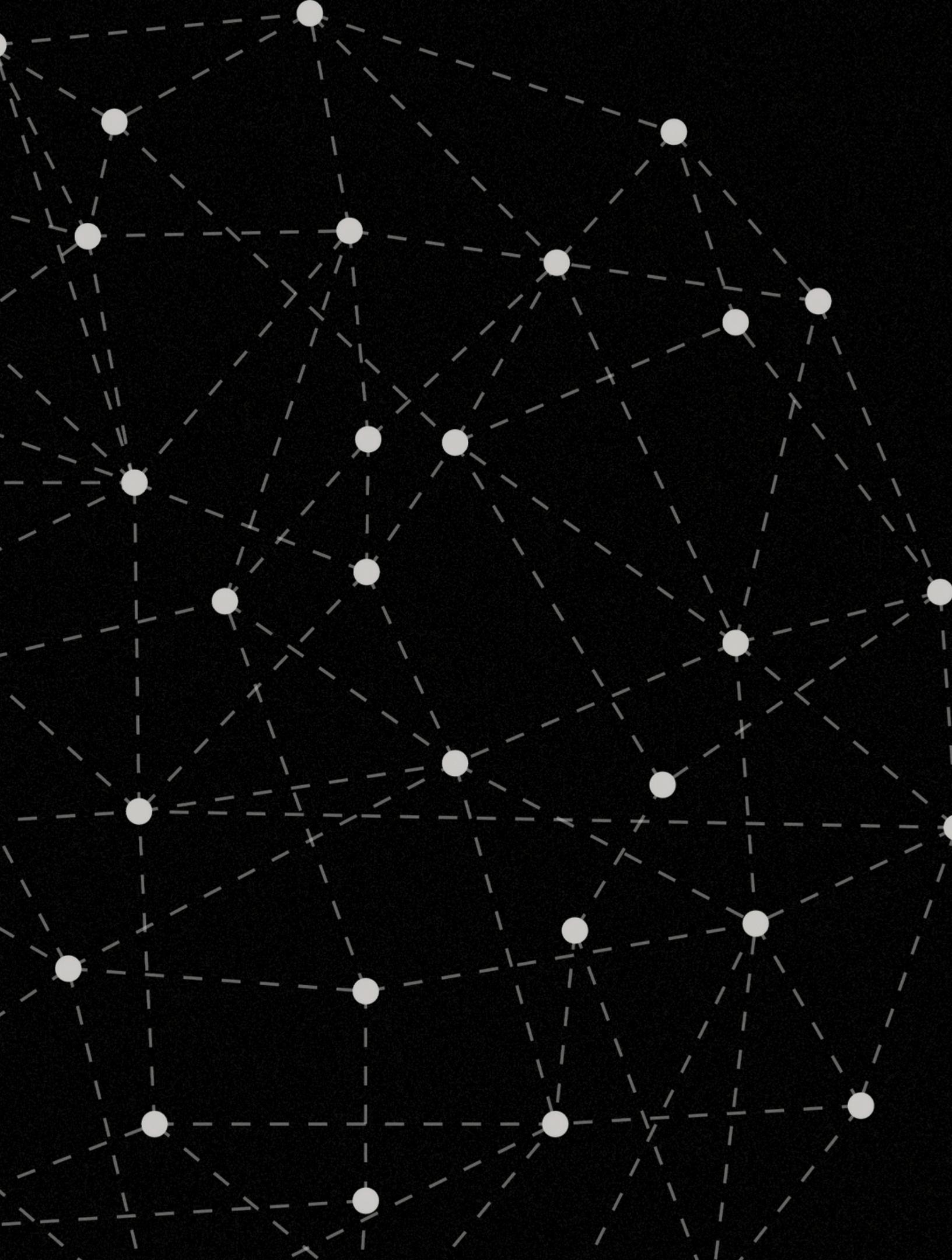
# Trabajos Relacionados

## RL4RS

Un entorno de prueba especializado en aprendizaje por refuerzo para sistemas de recomendación. Facilita el desarrollo y evaluación de algoritmos de refuerzo en aplicaciones de recomendación, optimizando la experiencia del usuario y la precisión de las recomendaciones.

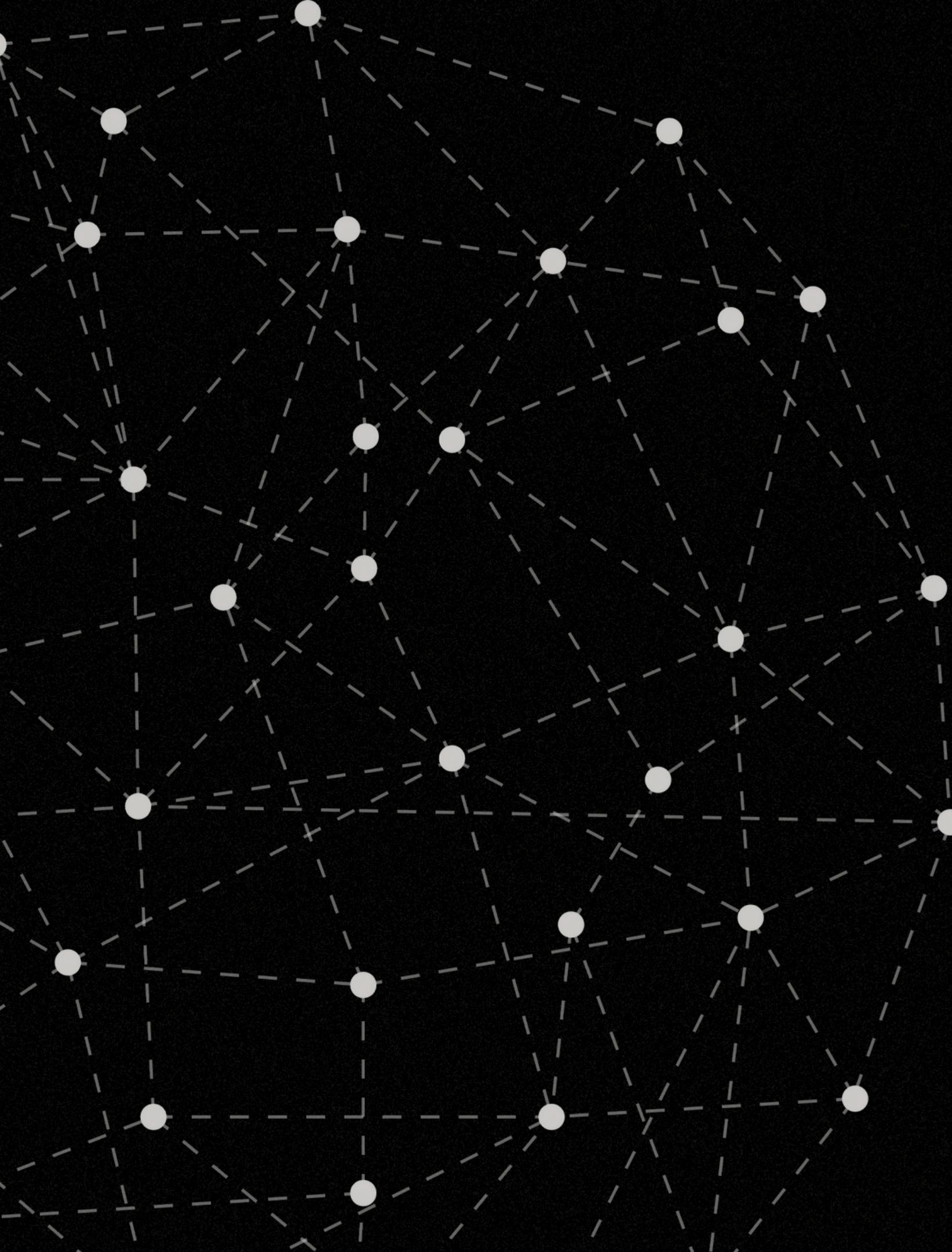
## RECSIM

Desarrollado por Google, plataforma de simulación configurable. Soporta interacción secuencial con usuarios y permite una configuración sencilla variando suposiciones del usuario como preferencias, elecciones, familiaridad de los items y estado latente y dinámico del usuario.



# Problemas de estos simuladores

- Suelen simular un tipo de **interacción inmediata**.
- Pero los servicios actuales tienen **múltiples tipos de interacción**.
- Además los usuarios tienen acciones con efectos no inmediatos.
- Estos comportamientos no son simulados.
- Muchos no son fácilmente transferibles.
- No hay mucha investigación sobre **como evaluar** simuladores.



# Kuai Sim

# **Simulación que permite 3 niveles de aprendizaje**

**REQUEST**



**WHOLE-SESSION**



**CROSS-SESSION**



# Request

## Significado

Se refiere a la **interacción inmediata** del usuario con el RS. El usuario solicita una recomendación Top-N al RS.

## Aprendizaje

Permite que el RS aprenda a optimizar una lista Top-N con implicit feedback

## Simulación

Para lograr esto se **simulan distintos tipos de interacciones** (like, view, forward, hate, leave, etc)



# Whole-session

## Significado

Se considera el **tiempo continuo dentro de la aplicación como una sesión**. Por ende, en este contexto el usuario durante la sesión esta constantemente enviando requests.

## Simulación

KuiaSim genera una “**leave-signal**” que indica que el usuario ha salido de la aplicación.

## Aprendizaje

Permite que el RS aprenda a **optimizar las recompensas a largo plazo**.



# Cross-Session

## Significado

Hace referencia a la **interacción** con el usuario durante **varias sesiones**.

## Simulación

Si se presenta una “leave-signal” se genera un “**return time**” el cual se elige haciendo un sampling de una distribucion geométrica.

## Aprendizaje

Se impone la meta de **optimizar la retención**, es decir, que se aprenda una política que minimice el return time.  
return time: tiempo que el usuario vuelve a la aplicación



# RL-Setup y notación

$O_t$ : La request de un usuario. Esta request genera por parte del RS una recomendación  $A_t \in C^K$ . Esto significa que la recomendación corresponde a una lista de K candidatos (Top-K)

$U$ : Corresponde a las features de un usuario

$H_{:t-1}$ : Corresponde al historial de interacciones del usuario

# RL-Setup y notación

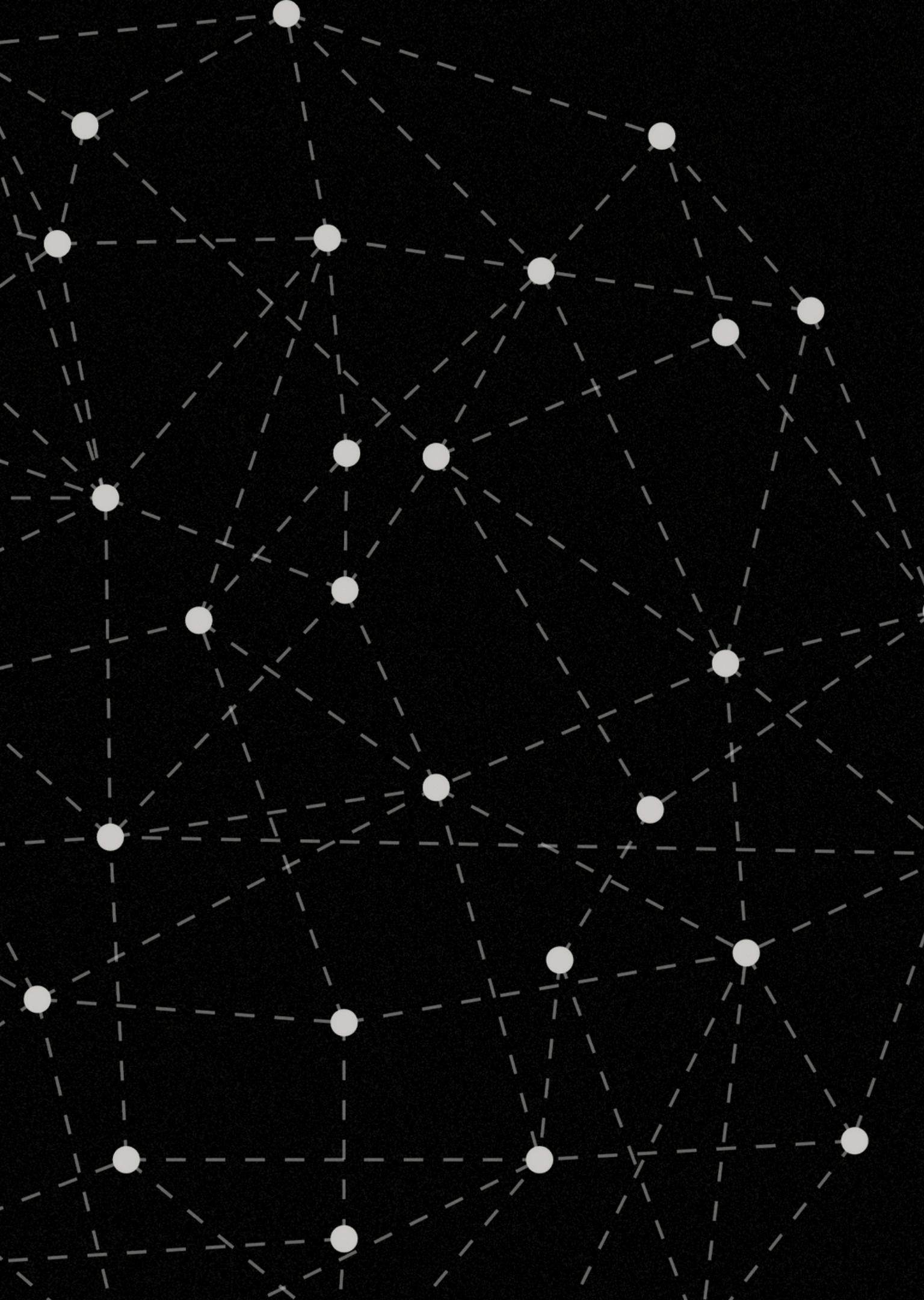
$$\mathcal{H}_{:t} \leftarrow \mathcal{H}_{:t-1} \oplus (\mathcal{A}_t, \mathcal{Y}_t^{(I)}, \mathcal{Y}_t^{(L)}, \mathcal{Y}_t^{(R)})$$

El historial de un usuario en el tiempo “t” corresponde a la concatenación del historial anterior con el vector correspondiente a la concatenación de la recomendación del tiempo t y el feedback del usuario en el tiempo t. Estos feedbacks pueden ser:

$\mathcal{Y}_t^{(I)} \in \mathbb{R}^{b \times K}$  : La interacción inmediata

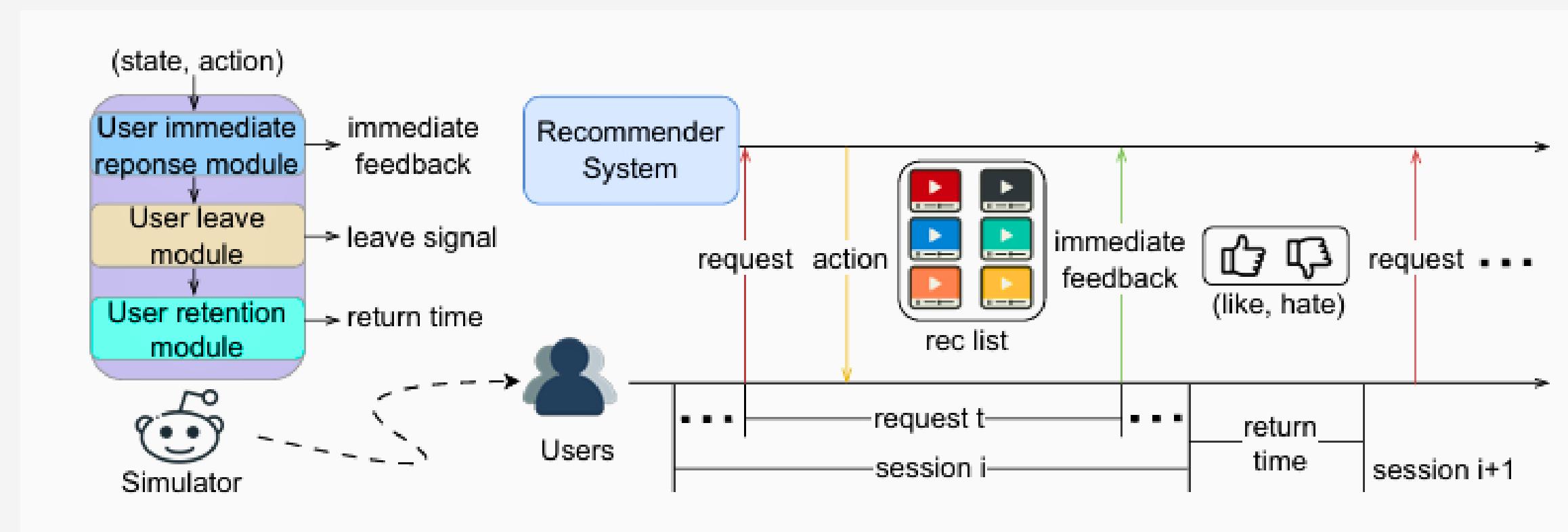
$\mathcal{Y}_t^{(L)} \in \{0, 1\}$  : La leave signal

$\mathcal{Y}_t^{(R)}$  : El return time



# **Simulator Building Blocks**

# Work Flow



# User immediate response module

## The user immediate response module:

- 1: User history encoding  $\mathbf{h}_t \leftarrow \text{Transformer}(\mathcal{U}, \mathcal{H}_{:t-1})$
- 2: Ground truth user state  $\mathbf{s}_t \leftarrow \mathbf{h}_t \oplus \mathcal{U}$
- 3: Behavior attention  $w_t \leftarrow \text{DNN}(\mathbf{s}_t)$
- 4: Behavior likelihood  $p(y|\mathbf{s}_t) \leftarrow w_t \odot \mathcal{A}_t - \rho \times \text{item\_correlation}(\mathcal{A}_t)$
- 5: Sample final immediate feedback  $\mathcal{Y}_t^{(I)} \sim p(y|\mathbf{s}_t)$

Este módulo es responsable de generar el **feedback inmediato** del usuario.

La función **item\_correlation** se introduce para **suprimir el comportamiento positivo** para items con alta correlación en una misma lista de recomendación.

# User Leave module

## The user leave module:

- 6: Immediate reward  $r_t \leftarrow \text{reward\_func}(\mathcal{Y}_t^{(I)})$
- 7: User temper  $\leftarrow$  user temper - immediate reward
- 8: Leave signal  $\mathcal{Y}_t^{(L)} \leftarrow 1$  if user temper  $\leq \mathbb{T}$ ; 0 otherwise

Mantiene un **factor de paciencia**/temperamento de usuario que determina una señal de cuándo el usuario va a salir, se va perdiendo con interacciones.

Se determina un **tiempo de sesión máximo**

Valores iniciales de temperamento, su decrecimiento y su valor mínimo son hiperparámetros ajustables

# User retention module

## The user retention module:

- 9: Personal retention bias  $b_u \leftarrow \text{DNN}(\mathbf{s}_t)$
- 10: Response retention bias  $b_r \leftarrow \lambda_1 r_t$
- 11: Next day return probability  $p_{\text{ret}} \leftarrow b_u + b_r + \lambda_2 b$ , where  $b$  is the global retention bias
- 12: Return time  $\mathcal{Y}_t^{(R)} \sim \text{Geometric}(p_{\text{ret}})$  if  $\mathcal{Y}_t^{(L)} = 1$ , otherwise  $\mathcal{Y}_t^{(R)} = 0$

Diseñado para tareas cross-session, predice cuánto tiempo se demora el **usuario en volver a conectarse**. Con un máximo de 10 días.

Se predice la probabilidad de que vuelva al siguiente día ( $p_{\text{ret}}$ ).  
Esta **probabilidad** se distribuye de **forma geométrica**.

Una buena recomendación provoca que el usuario vuelva antes

# Post processing module

## Post processing module:

- 13: Update user history  $\mathcal{H}_{:t} \leftarrow \mathcal{H}_{:t-1} \oplus (\mathcal{A}_t, \mathcal{Y}_t^{(I)}, \mathcal{Y}_t^{(L)}, \mathcal{Y}_t^{(R)})$
- 14: If  $\mathcal{Y}_t^{(L)} == 1$ , the user leave the current session
- 15: Else if not reaching the max session number, then continue.
- 16: Otherwise, sample a new user  $\mathcal{U}, \mathcal{H}$  from data and replace the current user.

Módulo de post procesamiento, **ejecuta el término, continuación o cambio.**

Se **actualiza el historial** del usuario **concatenando** los **resultados** dados en los módulos previos.

Notemos que se puede volver a retomar el mismo usuario mas adelante.  
Tambien, notemos que al actualizar el historial el vector tendria un largo constantemente creciente, este problema no se menciona en el paper.

# Work Flow

---

**Algorithm 1** Step — the detail workflow of KuaiSim.

---

**Input Format:** observation  $\mathcal{U}$ ,  $\mathcal{H}_{:t-1}$ , and recommendation action  $\mathcal{A}_t$

**Output:** immediate feedback  $\mathcal{Y}_t^{(I)}$ , leave signal  $\mathcal{Y}_t^{(L)}$ , and retention signal  $\mathcal{Y}_t^{(R)}$  (cross-session)

**The user immediate response module:**

- 1: User history encoding  $\mathbf{h}_t \leftarrow \text{Transformer}(\mathcal{U}, \mathcal{H}_{:t-1})$
- 2: Ground truth user state  $\mathbf{s}_t \leftarrow \mathbf{h}_t \oplus \mathcal{U}$
- 3: Behavior attention  $w_t \leftarrow \text{DNN}(\mathbf{s}_t)$
- 4: Behavior likelihood  $p(y|\mathbf{s}_t) \leftarrow w_t \odot \mathcal{A}_t - \rho \times \text{item\_correlation}(\mathcal{A}_t)$
- 5: Sample final immediate feedback  $\mathcal{Y}_t^{(I)} \sim p(y|\mathbf{s}_t)$

**The user leave module:**

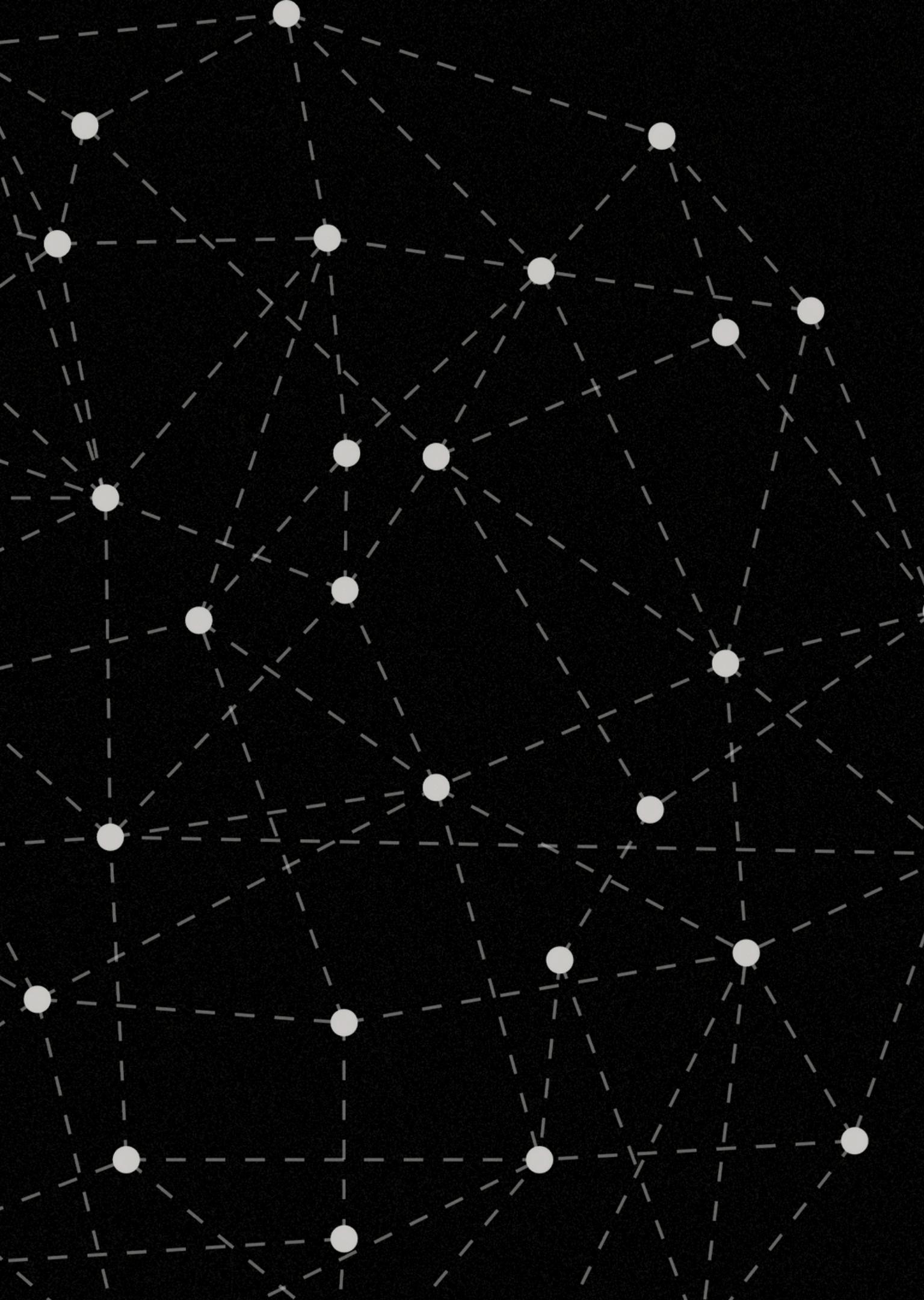
- 6: Immediate reward  $r_t \leftarrow \text{reward\_func}(\mathcal{Y}_t^{(I)})$
- 7: User temper  $\leftarrow$  user temper - immediate reward
- 8: Leave signal  $\mathcal{Y}_t^{(L)} \leftarrow 1$  if user temper  $\leq T$ ; 0 otherwise

**The user retention module:**

- 9: Personal retention bias  $b_u \leftarrow \text{DNN}(\mathbf{s}_t)$
- 10: Response retention bias  $b_r \leftarrow \lambda_1 r_t$
- 11: Next day return probability  $p_{\text{ret}} \leftarrow b_u + b_r + \lambda_2 b$ , where  $b$  is the global retention bias
- 12: Return time  $\mathcal{Y}_t^{(R)} \sim \text{Geometric}(p_{\text{ret}})$  if  $\mathcal{Y}_t^{(L)} = 1$ , otherwise  $\mathcal{Y}_t^{(R)} = 0$

**Post processing module:**

- 13: Update user history  $\mathcal{H}_{:t} \leftarrow \mathcal{H}_{:t-1} \oplus (\mathcal{A}_t, \mathcal{Y}_t^{(I)}, \mathcal{Y}_t^{(L)}, \mathcal{Y}_t^{(R)})$
  - 14: If  $\mathcal{Y}_t^{(L)} == 1$ , the user leave the current session
  - 15: Else if not reaching the max session number, then continue.
  - 16: Otherwise, sample a new user  $\mathcal{U}, \mathcal{H}$  from data and replace the current user.
-



# **Resultados Experimentales**

# Data sets

## KuaiRand

Recolectado de la red social Kuaishou. Se construyó para que fuera **insensgado** a las preferencias de los items. Presenta 6 atributos positivos ('click', 'view time', 'like', 'comment', 'follow' y 'forward) y 2 negativos ('hate' y 'leave').



## MovieLens-1m

Conjunto de datos con un millón de calificaciones de películas de 6,000 usuarios sobre 4,000 películas.



# Baselines

List-wise  
recommendation con  
**Request level simulator**

- Collaborative Filtering (CF)
- ListCVAE
- Pre-ranking Approach (PRM)

Sequential  
recommendation con  
**Whole-session level  
simulator**

- A2C
- SA2C
- **DDPG**
- TD3
- HAC

Retention optimization  
con  
**Cross-session level  
simulator**

- CEM
- TD3
- RLUR

# Métricas

List-wise recommendation con  
**Request level simulator**

- **List-wise reward (L-reward)**: Promedio de la recompensa de la lista
- **Coverage**: Promedio del número de distintos items recomendados por mini-batch.
- **Intra-list diversity (ILD)**: disimilitud entre todos los pares de ítems.

Sequential recommendation con  
**Whole-session level simulator**

- **Total reward**: Suma del promedio de las recompensas inmediatas por sesión.
- **Average reward**: Promedio de total rewards por request
- **Depth**: Número de interacciones promedio antes de desconexión.

Retention optimization con  
**Cross-session level simulator**

- **Return day**: Intervalo de días promedio de retorno.
- **User retention**: Razón de usuarios que vuelven a visitar el sistema.

# Resultados

Table 3: Benchmarks for the request level task of KuaiSim. Best values are in bold.

Algorithm	Average L-reward	Max L-reward	Coverage	ILD
CF	<b><math>2.253 \pm 0.024</math></b>	$4.039 \pm 0.001$	$100.969 \pm 7.193$	$0.543 \pm 0.007$
ListCVAE	$2.075 \pm 0.039$	<b><math>4.042 \pm 0.001</math></b>	<b><math>446.100 \pm 15.648</math></b>	<b><math>0.565 \pm 0.004</math></b>
PRM	$2.174 \pm 0.017$	$3.811 \pm 0.003$	$27.520 \pm 3.210$	$0.538 \pm 0.004$

# Resultados

Table 4: Benchmarks for the whole-session task of KuaiSim. Best values are in bold.

Algorithm	Depth	Average reward	Total reward	Coverage	ILD
TD3	$14.63 \pm 0.03$	$0.6476 \pm 0.0028$	$9.4326 \pm 0.0756$	$24.20 \pm 2.55$	$0.9864 \pm 0.0004$
A2C	$14.02 \pm 0.02$	$0.5950 \pm 0.0019$	$8.3905 \pm 0.1026$	$27.41 \pm 1.08$	$0.9870 \pm 0.0002$
SA2C	$14.34 \pm 0.02$	$0.6251 \pm 0.0014$	$8.9547 \pm 0.0241$	$27.14 \pm 2.01$	$0.9872 \pm 0.0002$
DDPG	$14.89 \pm 0.04$	$0.6841 \pm 0.0013$	$10.0873 \pm 0.0571$	$20.95 \pm 3.27$	$0.9850 \pm 0.0006$
HAC	<b><math>14.98 \pm 0.03</math></b>	<b><math>0.6895 \pm 0.0017</math></b>	<b><math>10.1742 \pm 0.0634</math></b>	<b><math>35.70 \pm 1.22</math></b>	<b><math>0.9874 \pm 0.0004</math></b>

Table 5: Benchmarks for the cross-session task of KuaiSim. Best values are in bold.

Algorithm	Return day ↓	User retention ↑
CEM	$3.573 \pm 0.012$	$0.572 \pm 0.002$
TD3	$3.556 \pm 0.010$	$0.581 \pm 0.001$
RLUR	<b><math>3.481 \pm 0.010</math></b>	<b><math>0.607 \pm 0.002</math></b>

↑: the higher the better; ↓: the lower the better.

# Resultados

Table 6: A comparison between KuaiSim and other simulators. Best values are in bold.

Simulators	Depth	Average reward	Total reward	AUC
RL4RS	$14.39 \pm 0.02$	$0.640 \pm 0.015$	$9.235 \pm 0.122$	$0.6929 \pm 0.0019$
Recogym	$13.55 \pm 0.01$	$0.535 \pm 0.013$	$7.194 \pm 0.109$	$0.6729 \pm 0.0026$
RecSim	$14.05 \pm 0.02$	$0.588 \pm 0.006$	$9.347 \pm 0.143$	$0.6842 \pm 0.0031$
VirtualTaobao	$14.45 \pm 0.02$	$0.646 \pm 0.009$	$9.570 \pm 0.077$	$0.6866 \pm 0.0014$
KuaiSim	<b><math>14.86^* \pm 0.01</math></b>	<b><math>0.679^* \pm 0.011</math></b>	<b><math>10.081^* \pm 0.116</math></b>	<b><math>0.7234^* \pm 0.0021</math></b>

“\*” indicates the statistically significant improvements (i.e., two-sided t-test with  $p < 0.05$ ) over the best baseline.

Table 7: Benchmarks for the whole-session task on ML-1m datasets. Best values are in bold.

Algorithm	Depth	Average Reward	Total reward	Coverage	ILD
TD3	$13.50 \pm 0.01$	$0.5388 \pm 0.007$	$7.4035 \pm 0.0152$	$44.18 \pm 2.19$	$0.9866 \pm 0.0001$
A2C	$13.55 \pm 0.01$	$0.5468 \pm 0.002$	$7.4487 \pm 0.0171$	$27.31 \pm 1.44$	$0.9870 \pm 0.0001$
DDPG	$13.64 \pm 0.02$	$0.5476 \pm 0.005$	$7.5333 \pm 0.0273$	<b><math>61.03 \pm 2.08</math></b>	$0.9871 \pm 0.0001$
HAC	<b><math>13.70 \pm 0.01</math></b>	<b><math>0.5482 \pm 0.008</math></b>	<b><math>7.5791 \pm 0.0206</math></b>	$29.85 \pm 1.92$	<b><math>0.9872 \pm 0.0001</math></b>

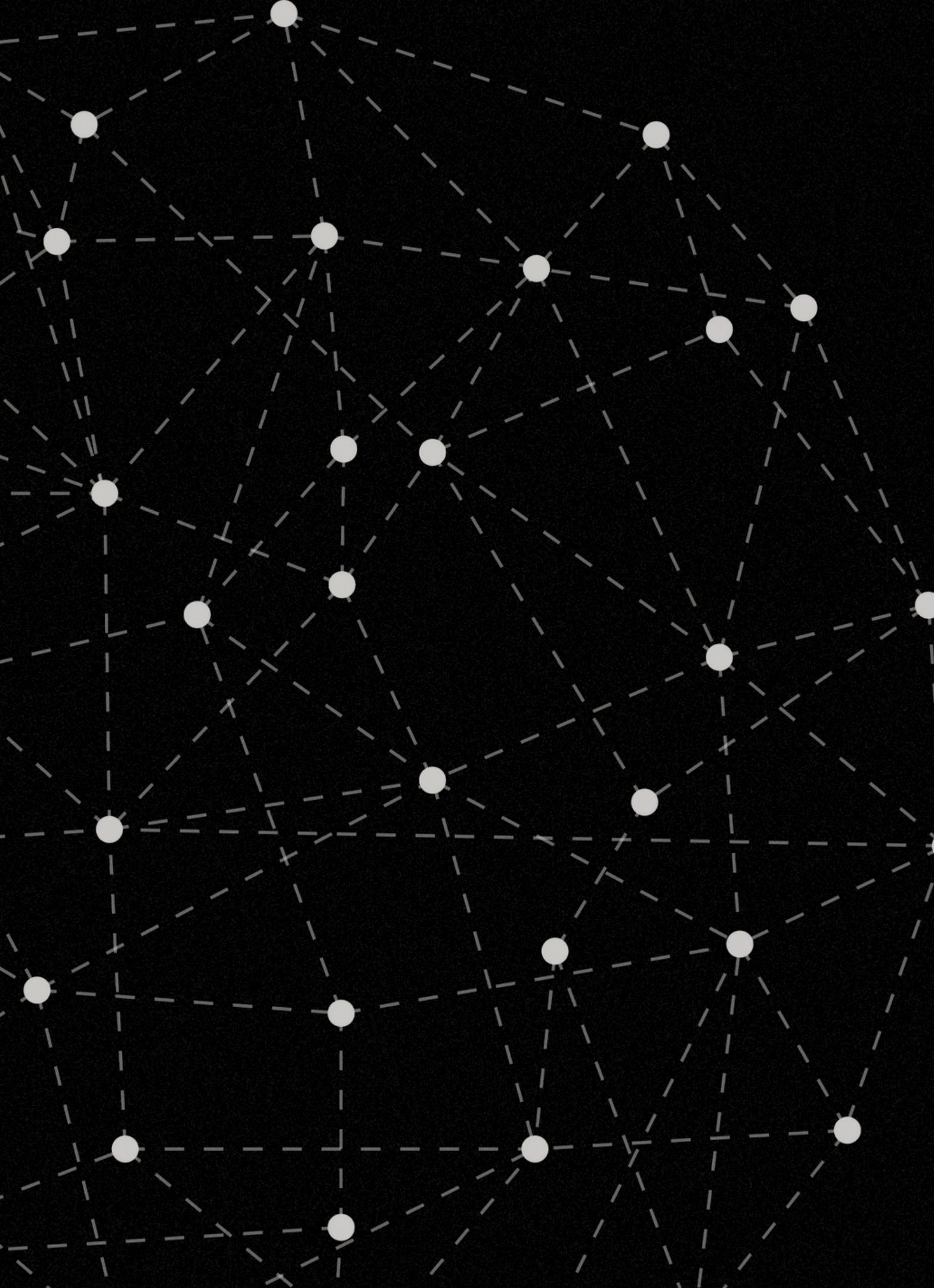
\*Algunos simuladores tienen solo una señal de retroalimentación de click por lo que el AUC va a esa señal

# Conclusiones

- KuaiSim funciona eficientemente como ambiente de simulación a los niveles de **Request**, **Whole-session** y **Cross-session**.
- Funciona a través de evaluar la **respuesta inmediata**, la **salida** y la **retención** del usuario.
- Los resultados experimentales muestran que KuaiSim tiene la **capacidad de migrar** de dataset (ML-1m).
- La comparación con otros simuladores muestra que KuaiSim posee **ventaja significativa**.

# Comentarios

- Al igual que otros frameworks de simuladores de recomendadores, no considera desafíos importantes: **diversity** y **fairness**. Incluir métricas de interpretabilidad podría ser una estrategia prometedora.
- Si bien mencionan que KuaiSim posee capacidad de migrar, solo lo comparan con otro dataset de **videos**. Podría ser interesante verlo en otro campo (música, e-commerce, etc).
- No se discute el tema de la **adicción** que puede presentar problemas éticos.
- Habian partes del paper que nunca explicaban. Especificamente item\_correlation



29 DE OCTUBRE DE 2024

# ¡Gracias por la atención!

## Presentado por:

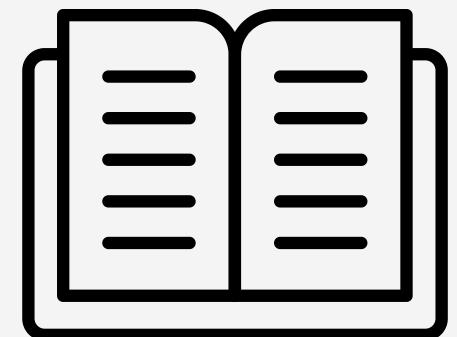
Eduardo Contreras  
Gonzalo Fuentes  
Sebastián Salgado

## Profesor

Denis Parra



# Bibliografía



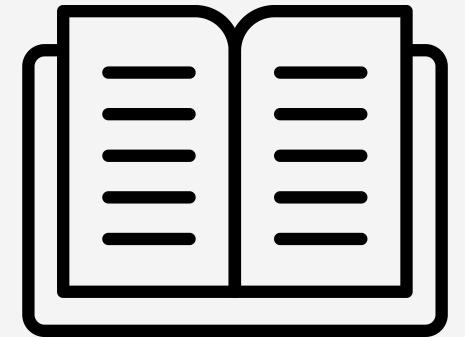
Fidelity Investments. (n.d.). Jurity: A library for recommendation system evaluation. Retrieved October 27, 2024, from [https://fidelity.github.io/jurity/about\\_reco.html](https://fidelity.github.io/jurity/about_reco.html)

le, E., Hsu, C., Mladenov, M., Jain, V., Narvekar, S., Wang, J., Wu, R., & Boutilier, C. (2019, September 11). RecSim: a configurable simulation platform for recommender systems. arXiv.org. <https://arxiv.org/abs/1909.04847>

Rohde, D., Bonner, S., Dunlop, T., Vasile, F., & Karatzoglou, A. (2018, August 2). RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. arXiv.org. <https://arxiv.org/abs/1808.00720>

Wang, K., Zou, Z., Zhao, M., Deng, Q., Shang, Y., Liang, Y., Wu, R., Shen, X., Lyu, T., & Fan, C. (2021, October 18). RL4RS: A Real-World Dataset for Reinforcement Learning based Recommender System. arXiv.org. <https://arxiv.org/abs/2110.11073>

# Bibliografía



Shi, J., Yu, Y., Da, Q., Chen, S., & Zeng, A. (2018, May 25). Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. arXiv.org. <https://arxiv.org/abs/1805.10000>

# Anexo: Métricas

## Intra-List Diversity

Intra-List Diversity@k measures the intra-list diversity of the recommendations when only k recommendations are made to the user. Given a list of items recommended to one user and the item features, the averaged pairwise cosine distances of items is calculated. Then the results from all users are averaged as the metric Intra-List Diversity@k. This metric has a range in  $[0, 1]$ . The higher this metric is, the more diversified items are recommended to each user. Let  $U$  denote the set of  $N$  unique users,  $u_i$  denote the i-th user in the user set,  $i \in \{1, 2, \dots, N\}$ .  $v_p^{u_i}, v_q^{u_i}$  are the item features of the p-th and q-th item in the list of items recommended to  $u_i$ ,  $p, q \in \{0, 1, \dots, k - 1\}$ .  $I^{u_i}$  is the set of all unique pairs of item indices for  $u_i$ ,  $\forall p < q, \{p, q\} \in I^{u_i}$ .

$$\text{Intra-list diversity} = \frac{1}{N} \sum_{i=1}^N \frac{\sum_{p,q,\{p,q\} \in I^{u_i}} (\text{cosine\_distance}(v_p^{u_i}, v_q^{u_i}))}{|I^{u_i}|}$$

out the spam such as advertisements. There are 7,583 items in total. For the target users, randomly select a batch of users and remove robots, which includes over 200,000 real users. Each time the recommender system recommends a video list to a user, decide whether to insert a random item with a fixed probability. If the answer is yes, then intervene in the recommendation list by randomly selecting one video from this list and replacing it with a random item uniformly sampled from the 7,583 items. KuaiRand removes the users that have been exposed to less than 10 randomly exposed videos for faithful evaluation. There are 27,285 users retained. All 7,583 items have been inserted at least once, and the total number of random interventions is 1,186,059. The KuaiRand dataset

Table 2: Statistics of datasets.

Datasets	Users	Items	Interactions	Sessions	Density
KuaiRand-Pure	27077	7551	1,436,609	246738	0.70%
ML-1m	6,400	3,706	1,000,208	16629	4.22%

## Probabilidad de retorno de usuario, se distribuye geométricamente

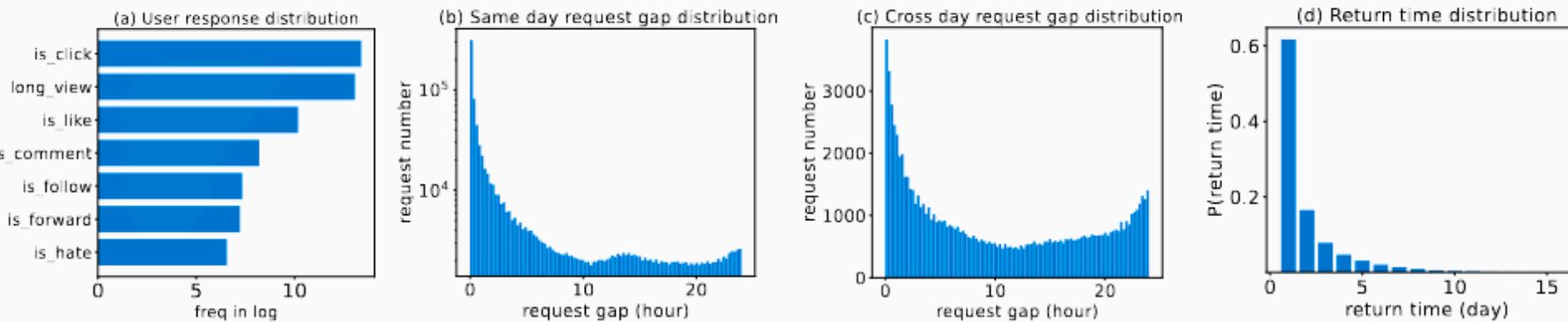


Figure 2: Data analysis on KuaiRand dataset. (a) User response distribution. (b) Same day request gap distribution. (c) Cross day request time gap distribution. (d) Return time distribution