LEARNING FINE-GRAINED USER INTERESTS FOR MICRO-VIDEO RECOMMENDATION

Francisco Wulf, José Tomás Valdivia
y Vicente Thomas

IIC3633 2024-2

CONTEXTO

Plataformas de micro-videos son cada día más populares

Una forma de diferenciarse entre sí es tener **mejores algoritmos de recomendación**

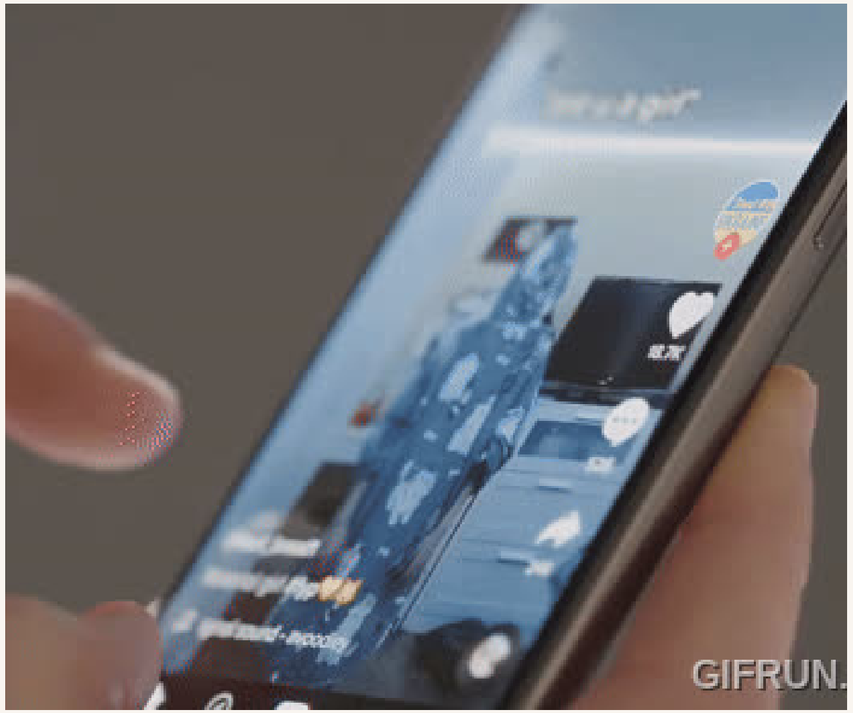
Las plataformas buscan **aumentar la retención** de sus usuarios

MOTIVACIÓN DEL ESTUDIO

Los usuarios tienen muchas formas de interactuar, más allá de Like vs Dislike

Es muy importante el momento del video donde se hace **SKIP**: puede indicar preferencias

Se busca estudiar las preferencias a nivel **granular**, esto es, fragmentos de cada video y no videos completos



DEFINICIÓN DEL PROBLEMA

Input: dado un conjunto de videos y los datos de interacción de los usuarios con ellos.

Output: calcular la probabilidad de que un usuario **no salte** ese video.

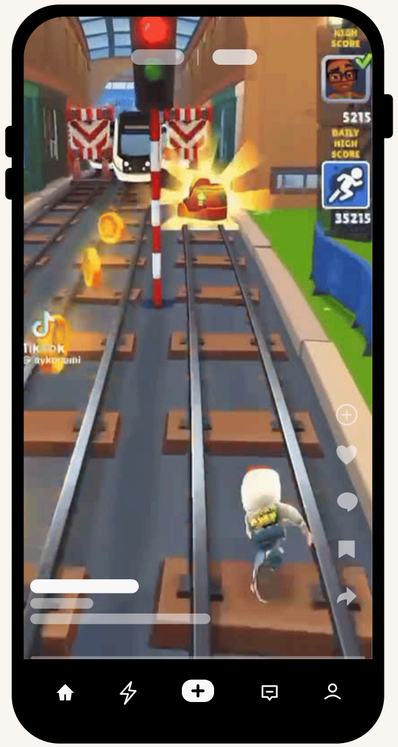
CONTRIBUCIÓN

Este modelo se enfoca en el **usuario** y cómo interactúa con cada **video** de manera diferente.

A diferencia de otros trabajos, el modelo divide cada video en **clips** de igual duración

Utiliza **grafos** para identificar cómo se relacionan los clips de cada video con cada usuario.

Toma en cuenta el clip exacto donde ocurre **SKIP como negativo**



TRABAJOS RELACIONADOS

> **1)** *Generalmente hablando, los métodos existentes se pueden clasificar según su método en a) collaborative filtering b) content-based y c) híbrido.*

> **2)** *Trabajos anteriores solo introducen* ***preferencias a nivel de video****, por ejemplo, extrayendo features del video completo [1] o incluso en algunos casos, utilizando solo un embedding del thumbnail del video [2, 3].*

TRABAJOS RELACIONADOS

> **3)** *Además* ***solo incorporan los skip o non-skip como interacciones*** *del usuario con el video, pero no incorporan qué clip del video fue el que gatilló el skip [1, 2, 3, 4, 5, 6].*

> **4)** *Los autores plantean que los trabajos anteriores presentan modelos basados en* ***coarse-grained user preferences.***

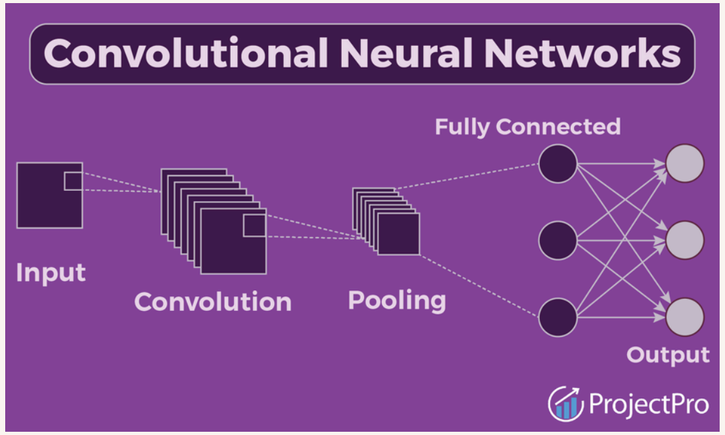
TRABAJOS RELACIONADOS

> **5)** *El trabajo de predicción es parecido a problema de Click-through rate (CTR) prediction, lo que hace que modelos que ya han sido adoptados en CTR como SVM y FM, puedan ser utilizados en el problema de micro-video recommendation.*

MARCO TEÓRICO

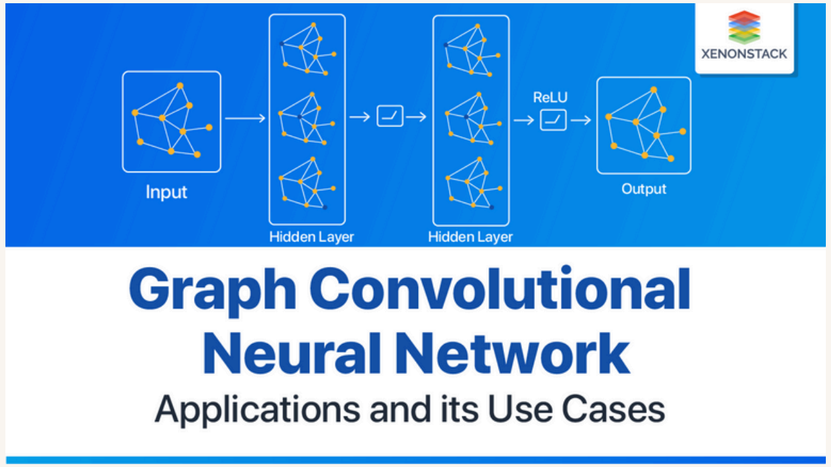
Convolutional Neural Network (CNN)

Compuestas por capas convolucionales, de agrupación y finalmente una capa totalmente conectada; las redes neuronales convolucionales **distinguen por su rendimiento superior en entradas de imaagen, voz y de señales de audio.** [7]



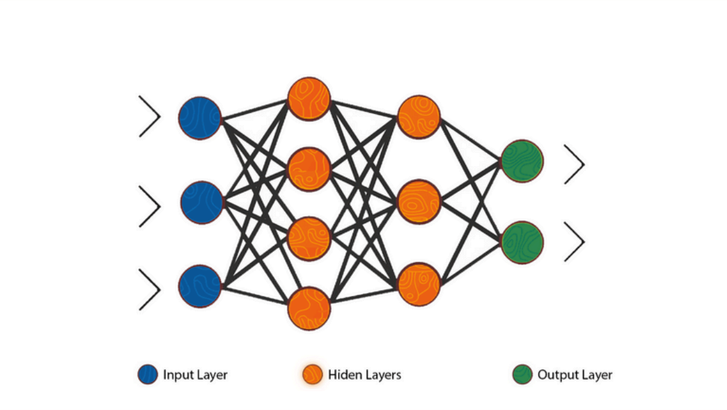
Graph Convolutional Network (GCN)

Similar a CNN pero **toma como input un grafo**, lo que la hace ideal para trabajar sobre estructruas de datos como grafos y árboles. [8]



Multilayer Perceptron

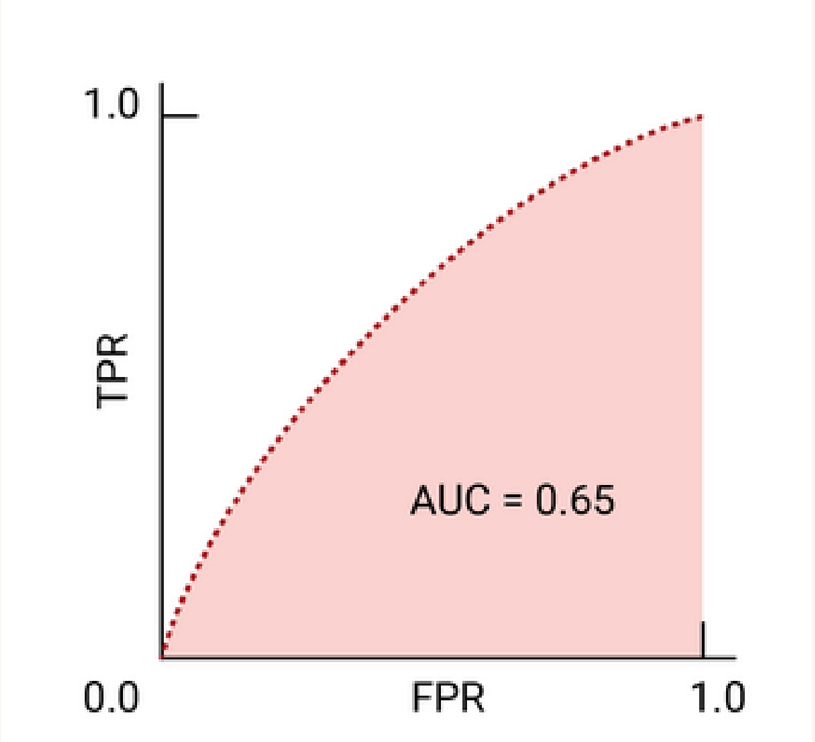
Algoritmo de aprendizaje supervisado. **Toma como input un vector de features** a través de la input layer, luego realiza operaciones de suma lineal en las hidden layers considerando los bias/weights, y finalmente produce un vector como output. [9]



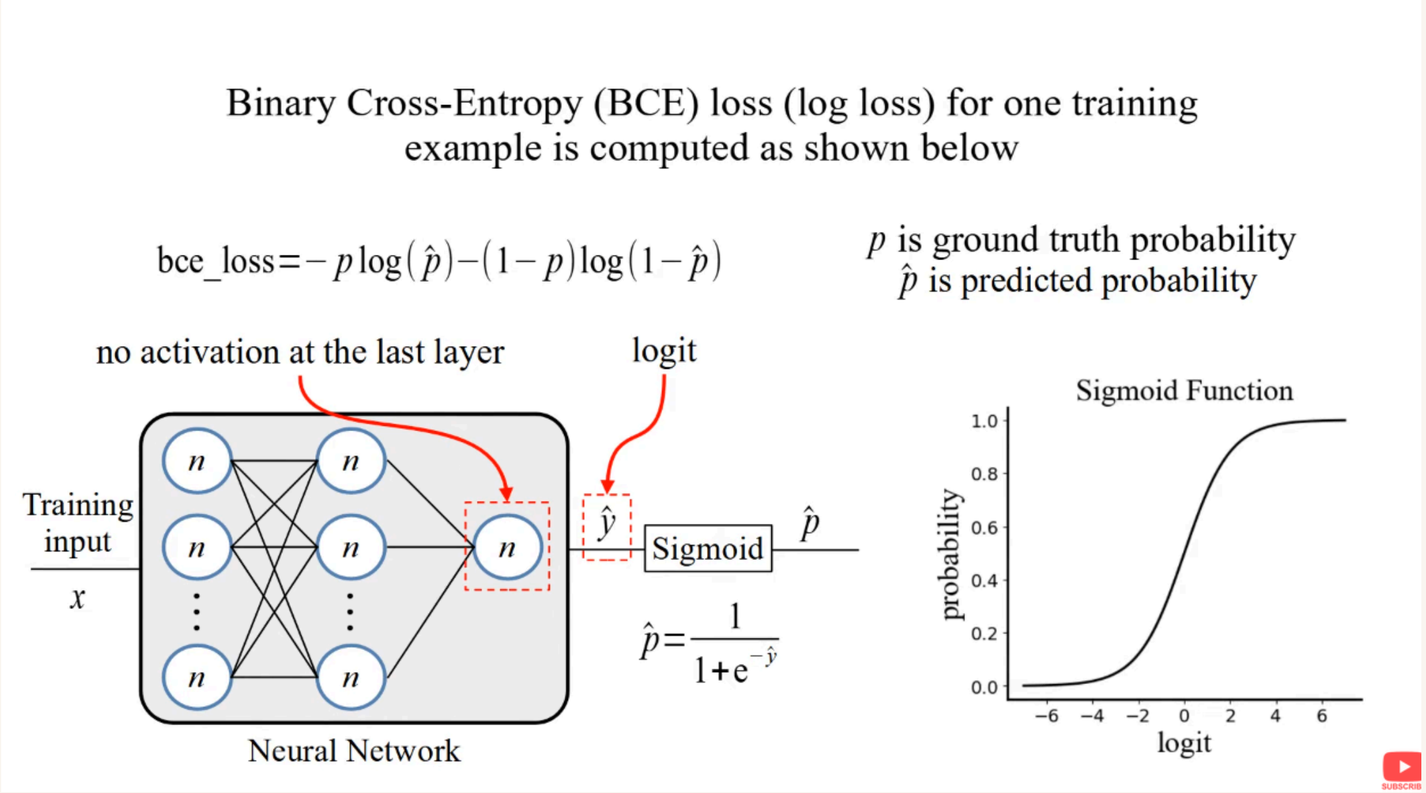
MARCO TEÓRICO

Area Under Curve ROC (AUC)

El área bajo la curva de precisión-recuperación representa la probabilidad (0 a 1) de que el modelo, si se da un ejemplo positivo y negativo elegidos al azar, clasificará el positivo mayor que el negativo [10].



MARCO TEÓRICO



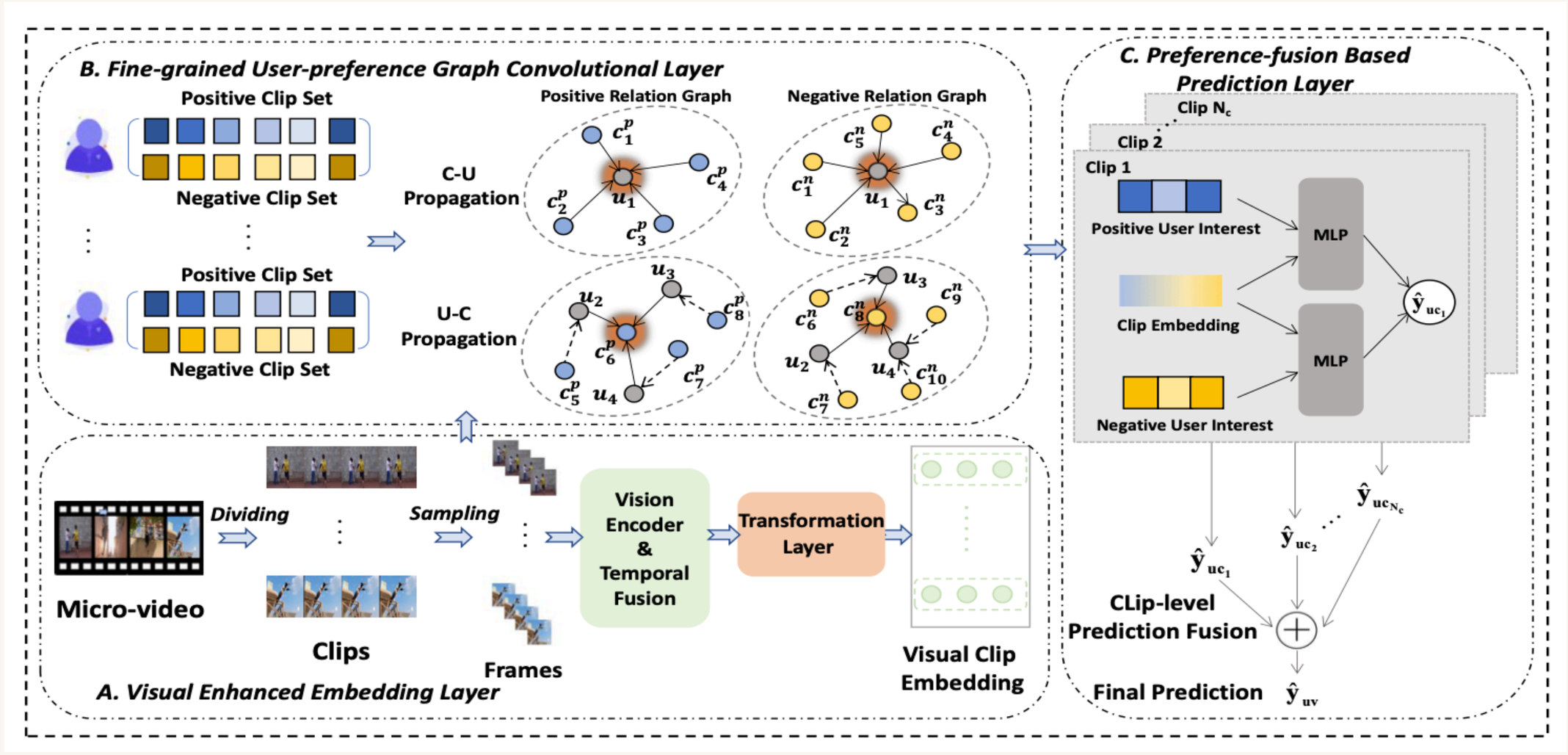
Binary cross-entropy loss (logloss)

Utilizado como función de pérdida en aprendizaje supervisado, binary cross-entropy loss **permite evaluar el desempeño de la tarea de predicción de probabilidad de ocurrencia de una variable binaria**. [11]

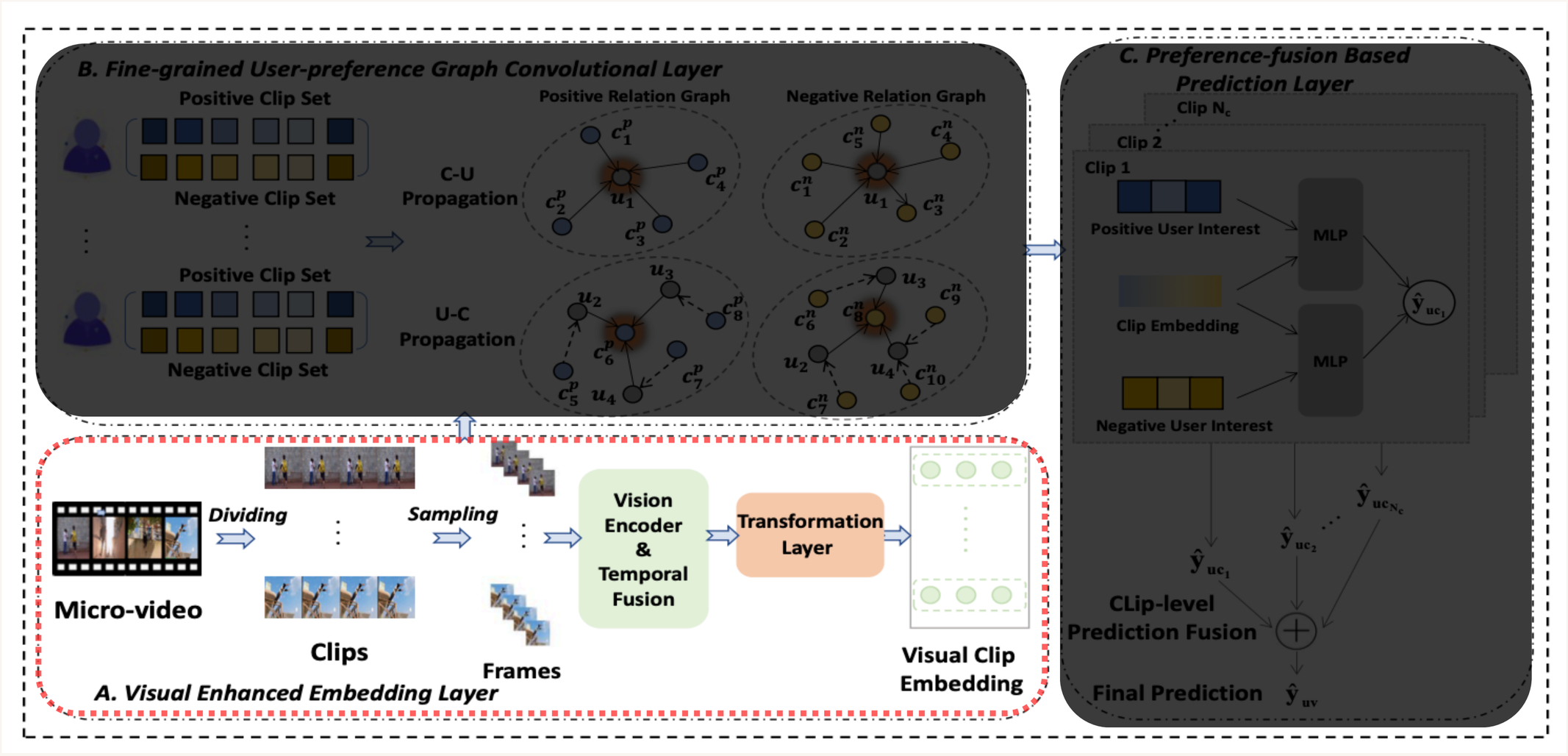
FRAME

Fine-g**RA**ined preference-modeling for **M**icro-video r**E**commendation

FRAME

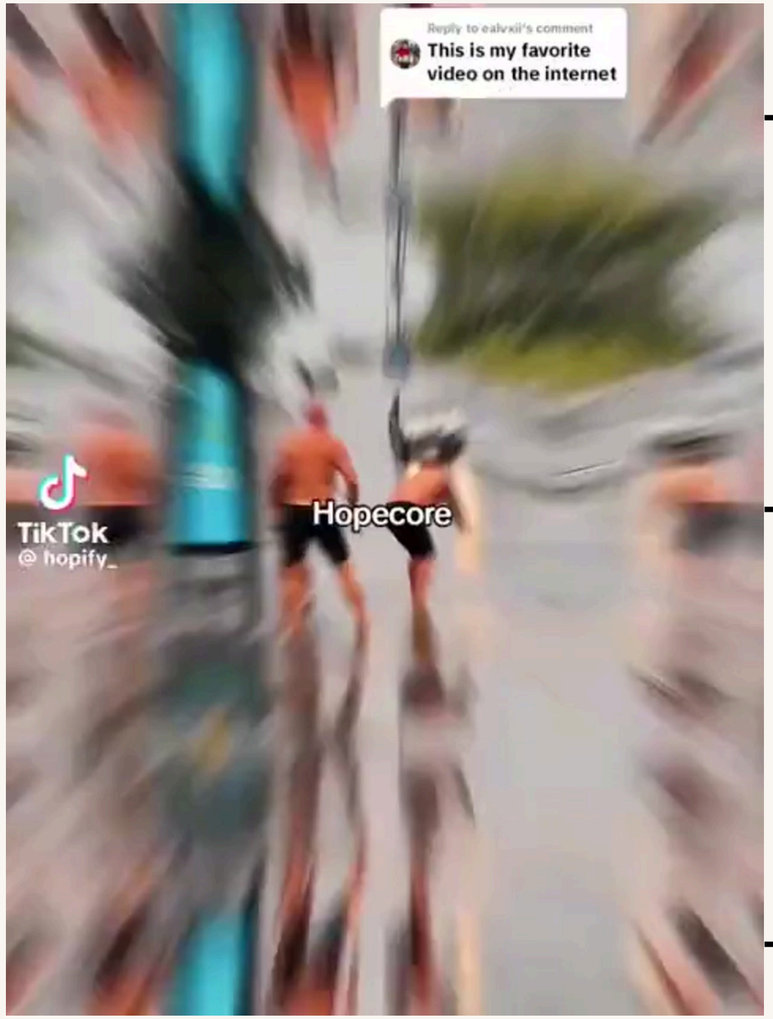

B. Fine-grained User-preference Graph Convolutional Layer
Positive Clip Set
Negative Clip Set
Positive Clip Set
Negative Clip Set
C-U Propagation
U-C Propagation
Positive Relation Graph
Negative Relation Graph
C. Preference-fusion Based Prediction Layer
Clip N_c
Clip 2
Clip 1
Positive User Interest
Clip Embedding
Negative User Interest
MLP
MLP
CLip-level Prediction Fusion
Final Prediction
Micro-video
Dividing
Sampling
Clips
Frames
Vision Encoder & Temporal Fusion
Transformation Layer
Visual Clip Embedding
A. Visual Enhanced Embedding Layer

FRAME


B. Fine-grained User-preference Graph Convolutional Layer
Positive Clip Set
Negative Clip Set
Positive Clip Set
Negative Clip Set
C-U Propagation
U-C Propagation
Positive Relation Graph
Negative Relation Graph
C. Preference-fusion Based Prediction Layer
Clip N_c
Clip 2
Clip 1
Positive User Interest
Clip Embedding
Negative User Interest
MLP
MLP
CLip-level Prediction Fusion
Final Prediction
Micro-video
Dividing
Sampling
Clips
Frames
Vision Encoder & Temporal Fusion
Transformation Layer
Visual Clip Embedding
A. Visual Enhanced Embedding Layer

FRAME

1. Visual Enhanced Embedding Layer



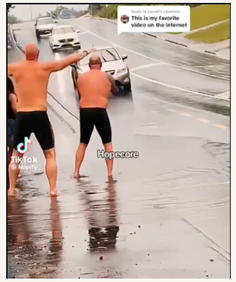
→ clip1
0s - 10s

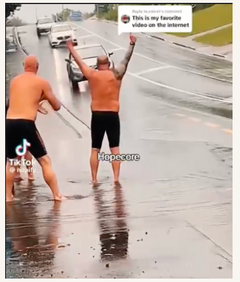
→ clip2
10s - 20s

→ clip3
20s - 30s

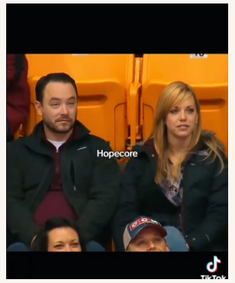
frames

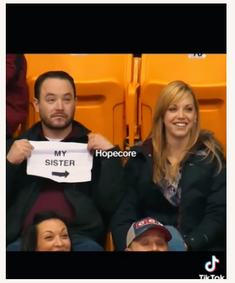
{



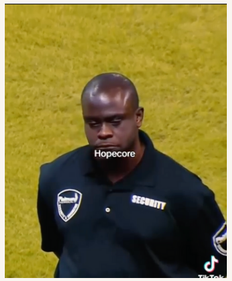


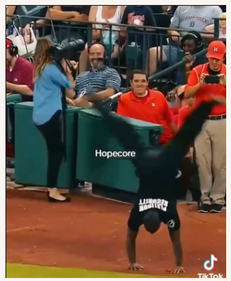
{





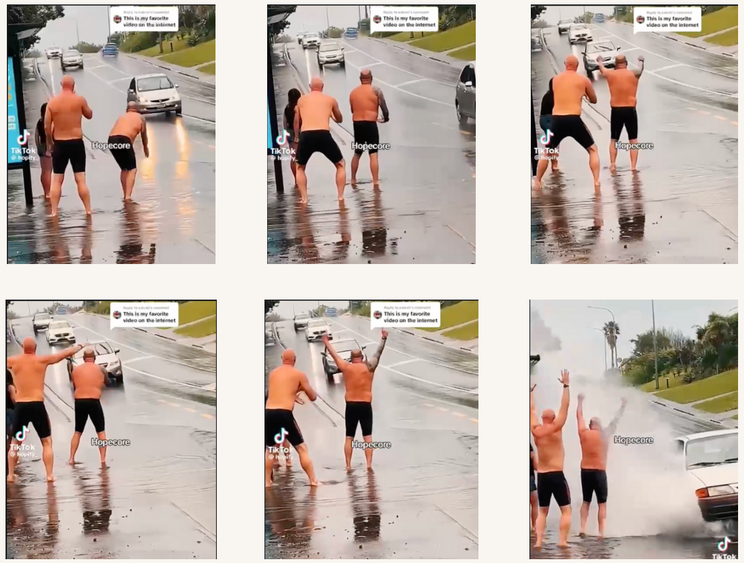
{





FRAME

1. Visual Enhanced Embedding Layer



2D-CNN
ResNet-50

Extrae características detalladas de imágenes

Visual Features de cada frame

$$\begin{bmatrix} 0.5 \\ -1.2 \\ 0.8 \\ 1.0 \end{bmatrix} \quad \begin{bmatrix} 0.3 \\ -1.0 \\ 0.9 \\ 0.8 \end{bmatrix} \quad \begin{bmatrix} 0.6 \\ -1.3 \\ 0.7 \\ 1.1 \end{bmatrix}$$

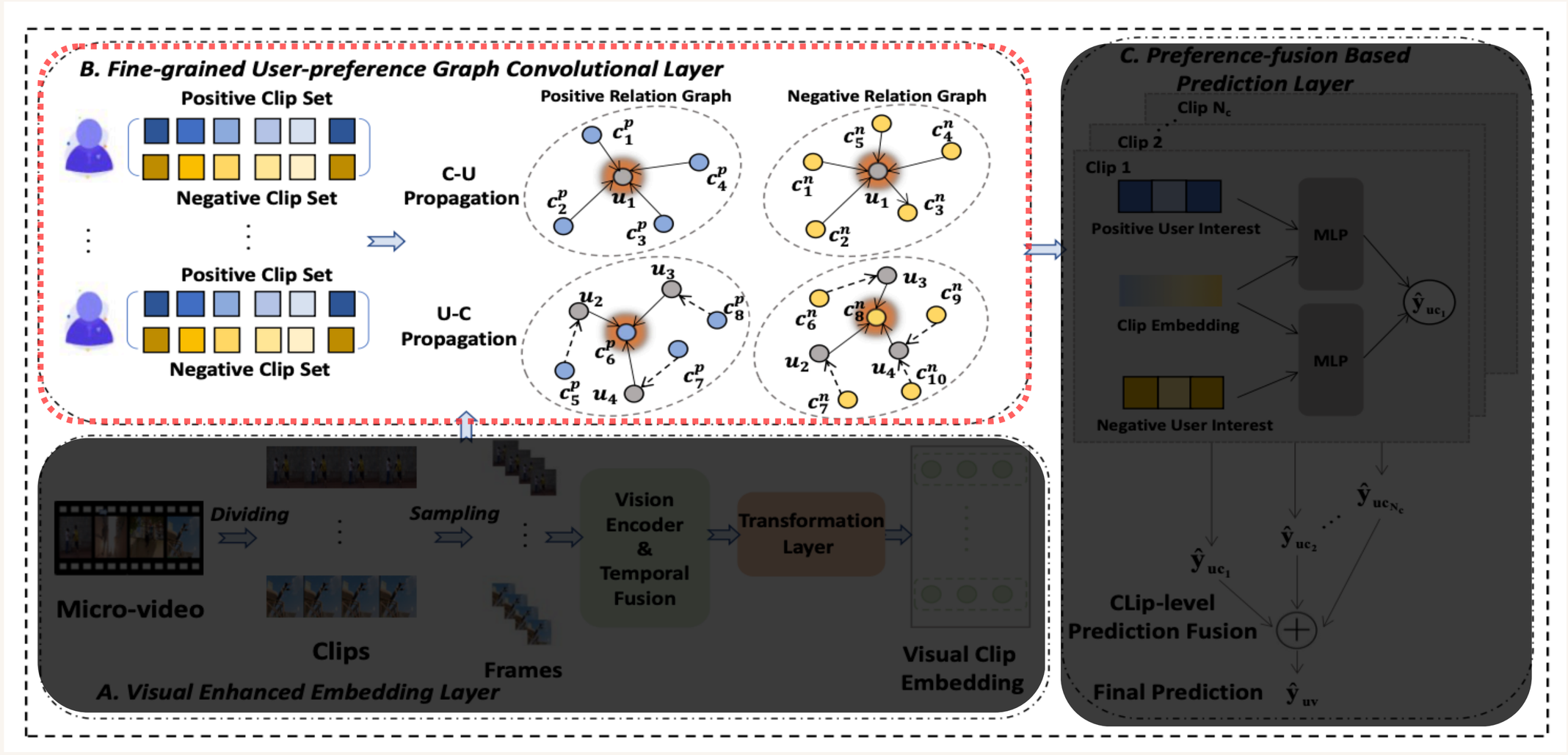
$$\begin{bmatrix} 0.4 \\ -1.1 \\ 0.85 \\ 0.95 \end{bmatrix} \quad \begin{bmatrix} 0.45 \\ -1.15 \\ 0.75 \\ 1.05 \end{bmatrix} \quad \begin{bmatrix} 0.55 \\ -1.25 \\ 0.65 \\ 1.2 \end{bmatrix}$$

mean-pooling

$$f_c$$

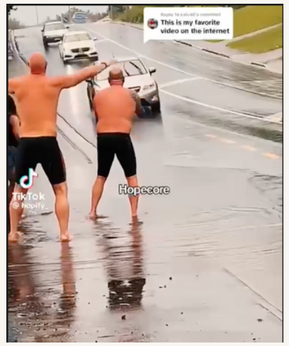
Visual Embedding del clip c

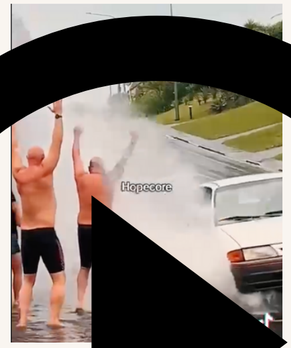
FRAME

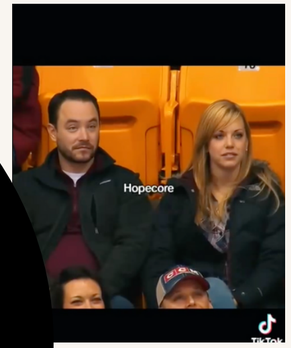

B. Fine-grained User-preference Graph Convolutional Layer
Positive Clip Set
Negative Clip Set
Positive Clip Set
Negative Clip Set
C-U Propagation
U-C Propagation
Positive Relation Graph
Negative Relation Graph
C. Preference-fusion Based Prediction Layer
Clip N_c
Clip 2
Clip 1
Positive User Interest
Clip Embedding
Negative User Interest
MLP
MLP
CLip-level Prediction Fusion
Final Prediction
Micro-video
Dividing
Sampling
Clips
Frames
Vision Encoder & Temporal Fusion
Transformation Layer
Visual Clip Embedding
A. Visual Enhanced Embedding Layer

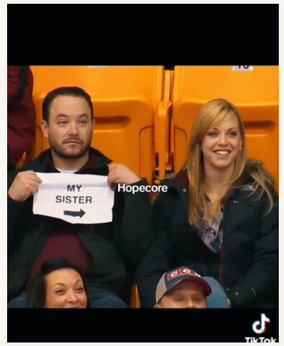
FRAME

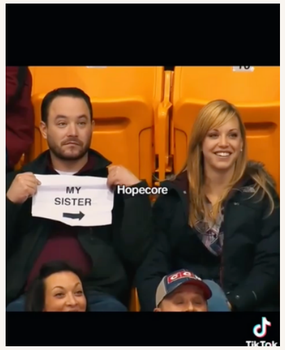
2. Fine Grained User Preference Graph

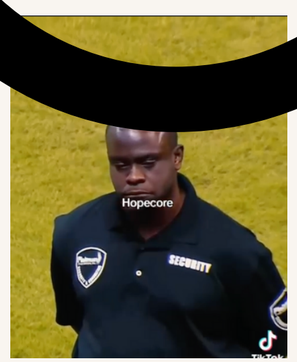


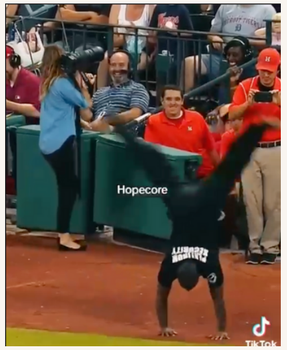






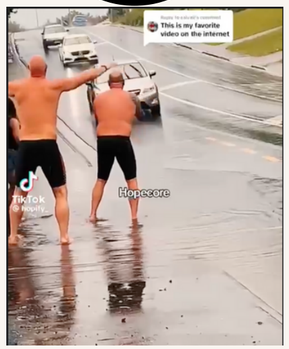


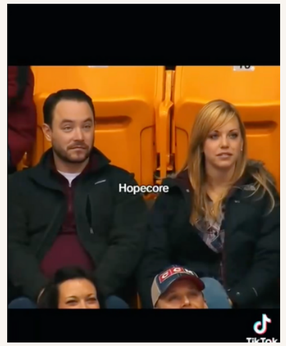


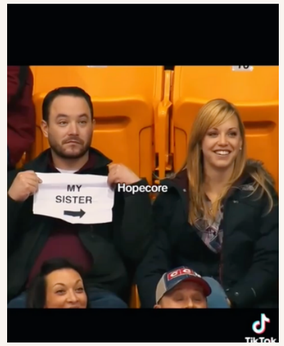


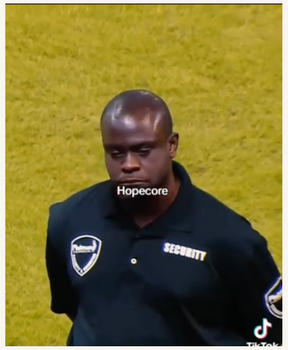
FRAME

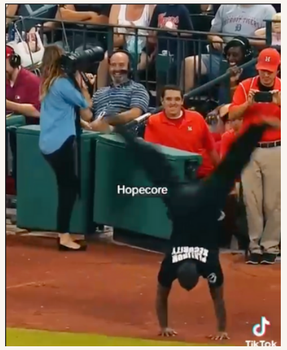
2. Fine Grained User Preference Graph





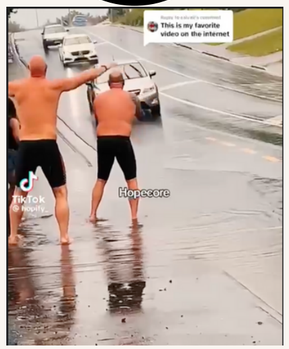


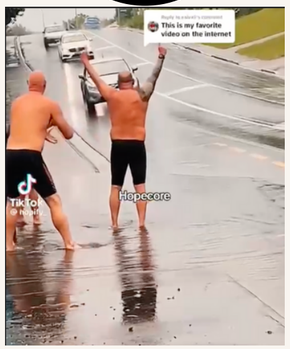


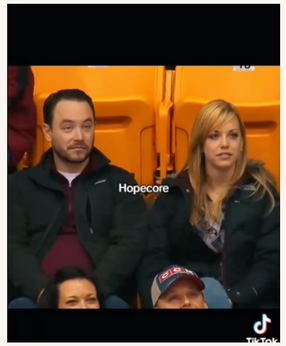


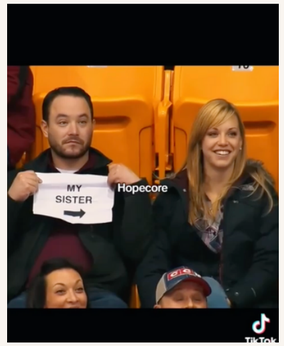
FRAME

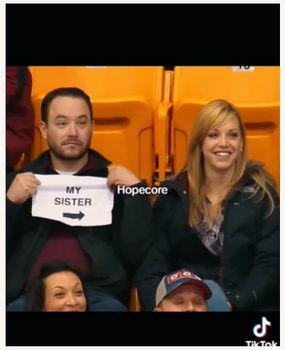
2. Fine Grained User Preference Graph

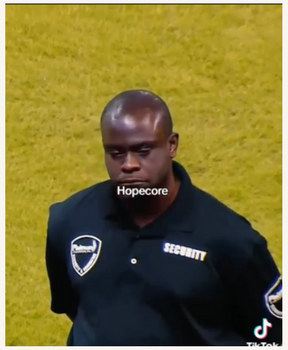


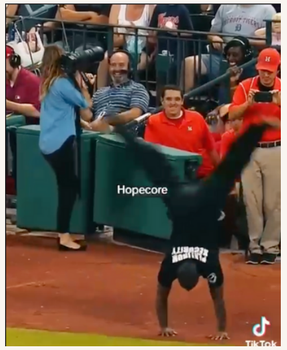






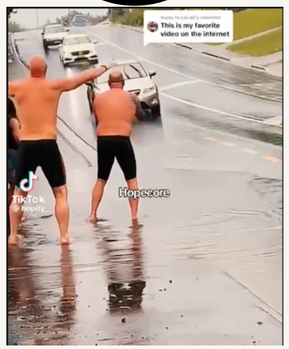


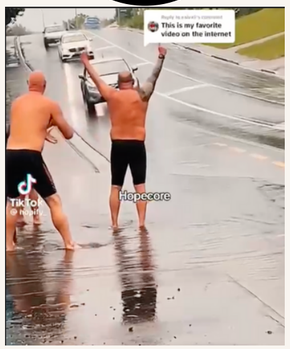


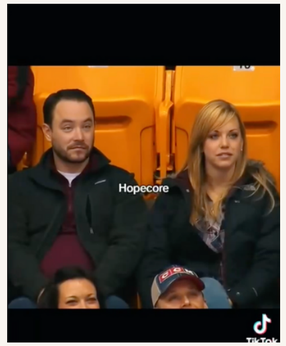


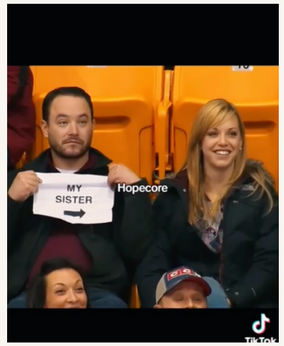
FRAME

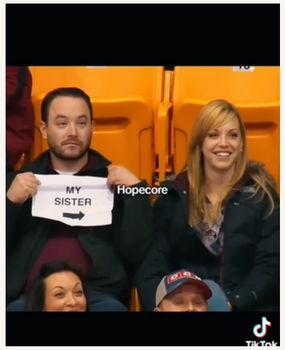
2. Fine Grained User Preference Graph

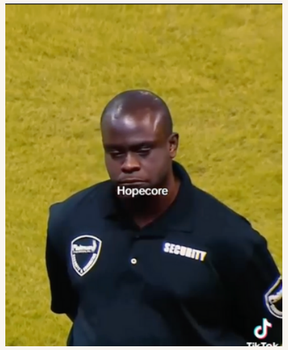


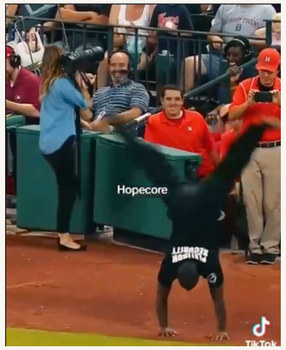








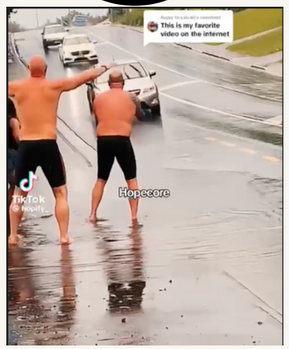


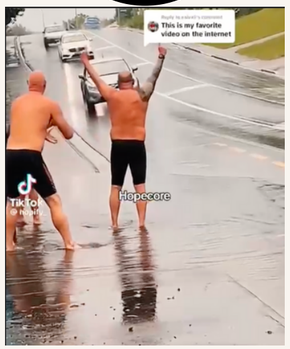


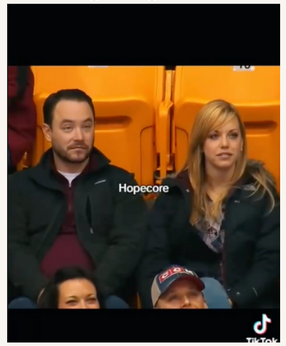
FRAME

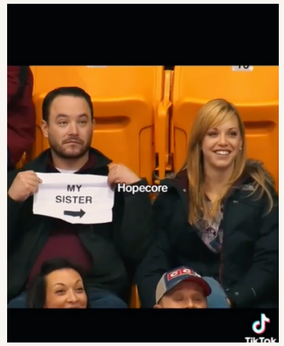
2. Fine Grained User Preference Graph

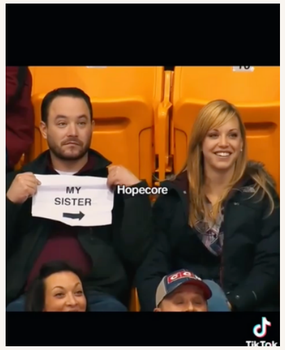
SKIP

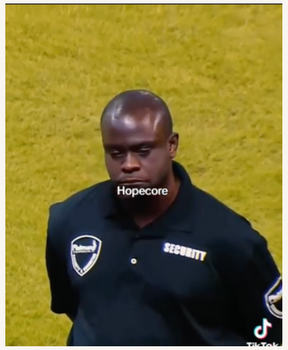


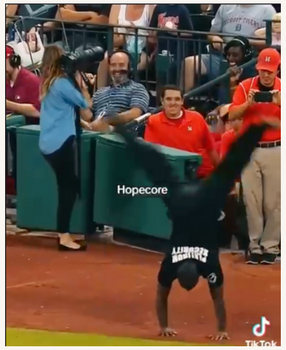






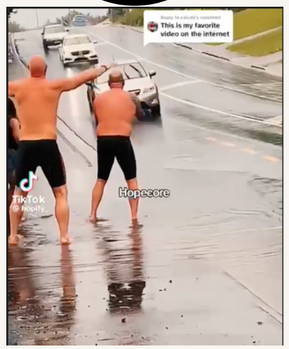


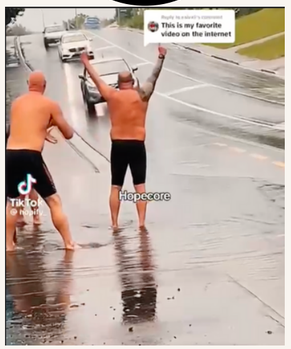


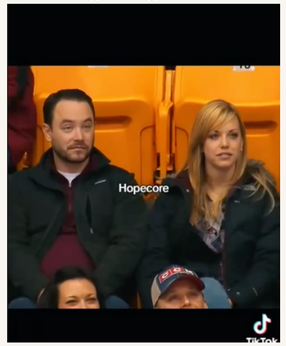


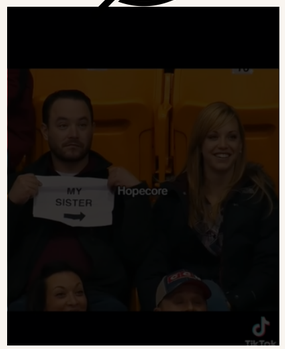
FRAME

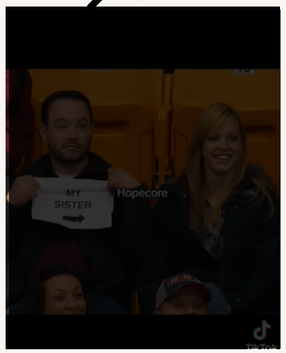
2. Fine Grained User Preference Graph

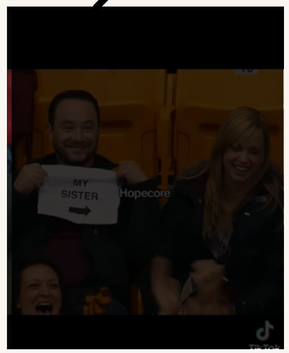


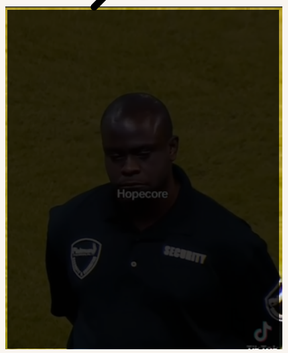


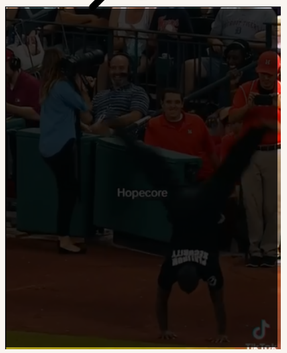












FRAME

2. Fine Grained User Preference Graph

Clips vistos pero de videos Skipped

Clips no vistos

NO son clasificados ni como positivos ni negativos

FRAME

2. Fine Grained User Preference Graph

Solo el clip donde ocurre el SKIP es clasificados como negativo

$$C_u^n = \{c_1^n, c_2^n, \cdots, c_{N_-^{(u)}}^n\},$$

C_u^n: conjunto de clips negativos del usuario u

c_1^n: clip del video 1 donde pasó skip

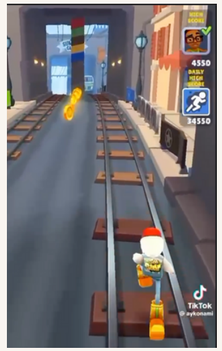
$N_-^{(u)}$: cantidad de videos skipped

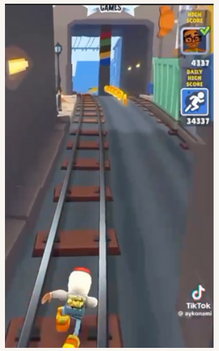
FRAME

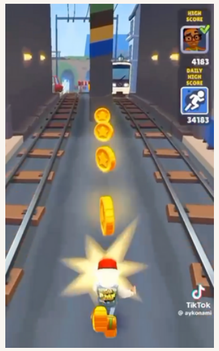
2. Fine Grained User Preference Graph

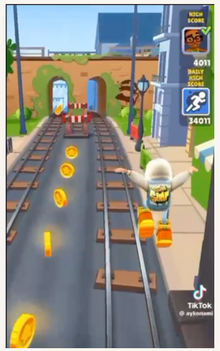
FRAME

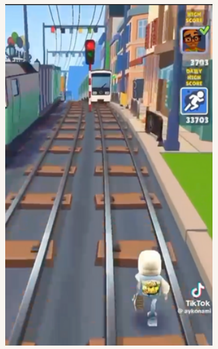
2. Fine Grained User Preference Graph

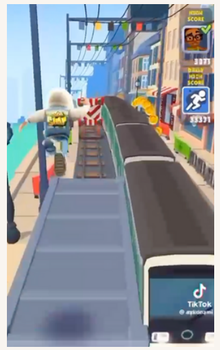


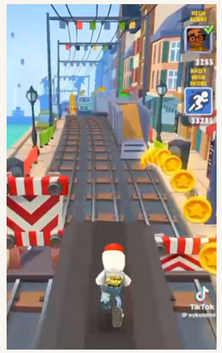


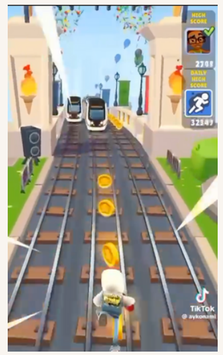






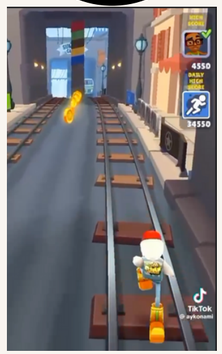
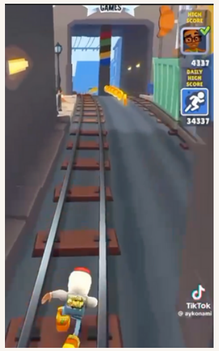
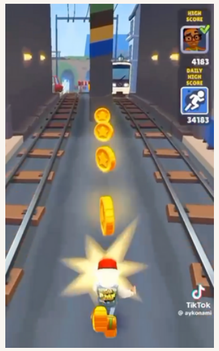
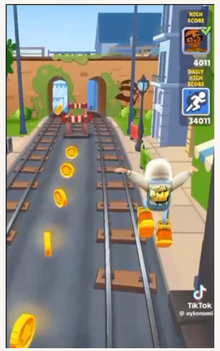
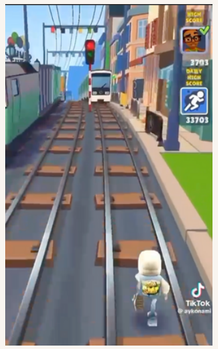
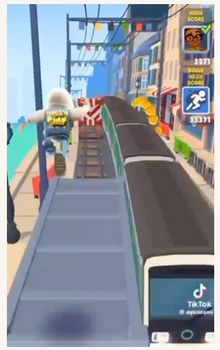
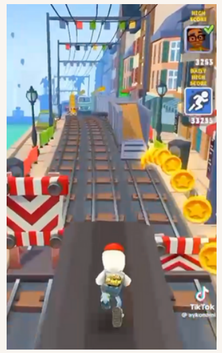


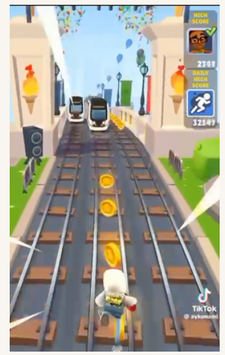




FRAME

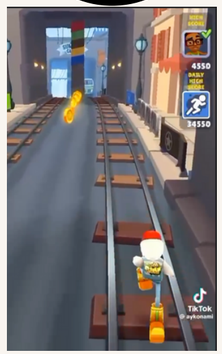
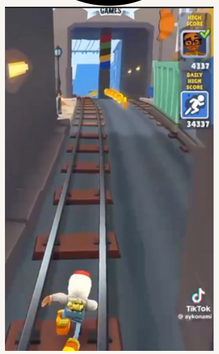
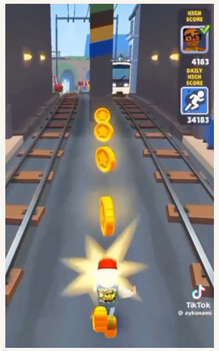
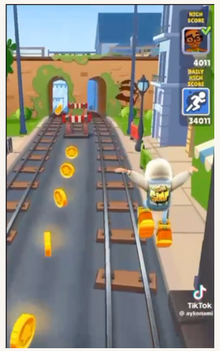
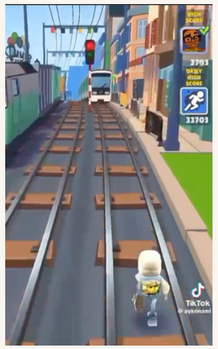
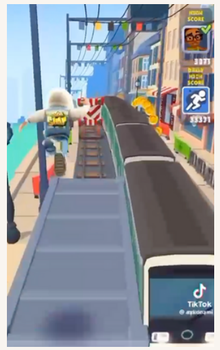
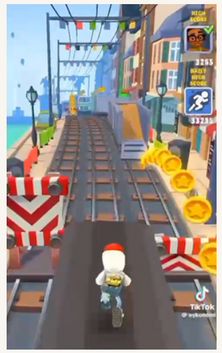
2. Fine Grained User Preference Graph

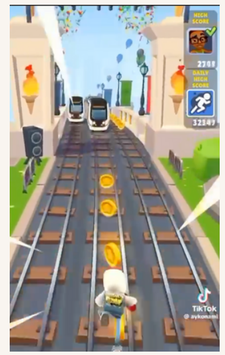










FRAME

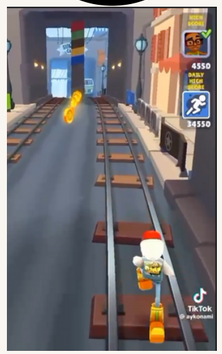
2. Fine Grained User Preference Graph

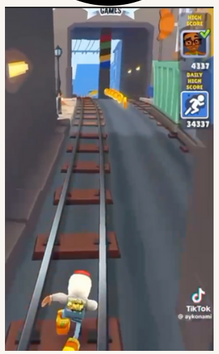








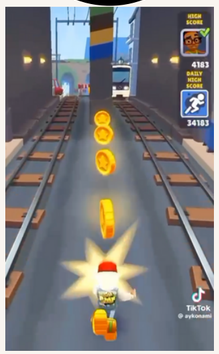


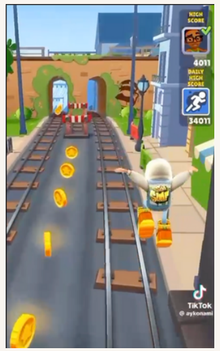
FRAME

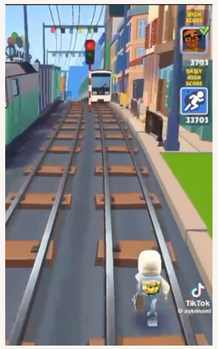
2. Fine Grained User Preference Graph

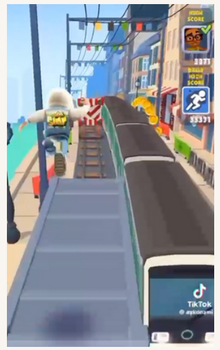


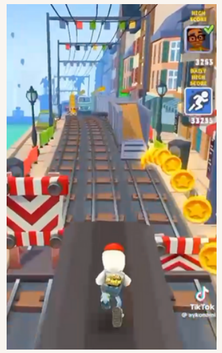


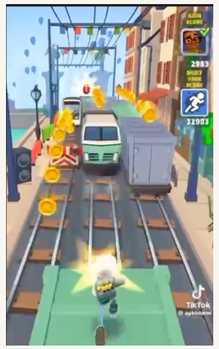


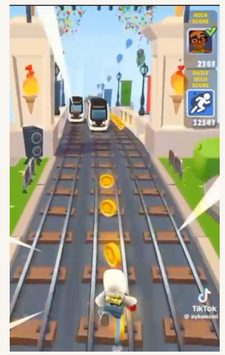






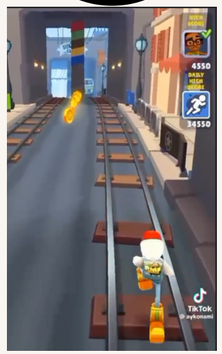
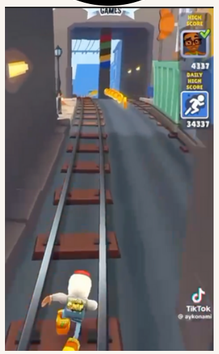
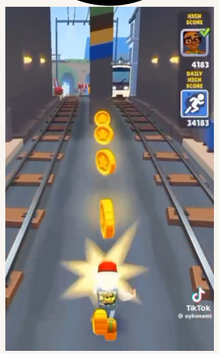
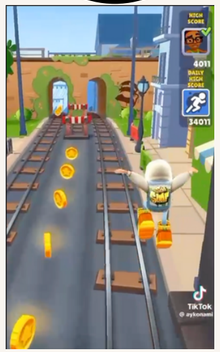
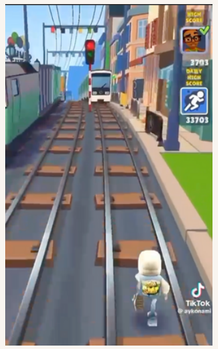
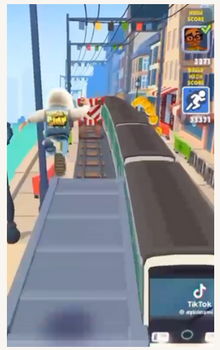
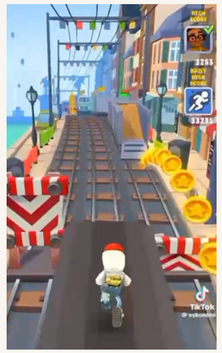


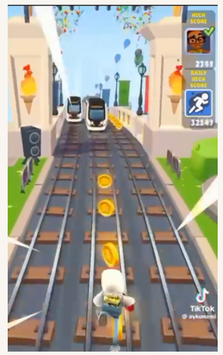




FRAME

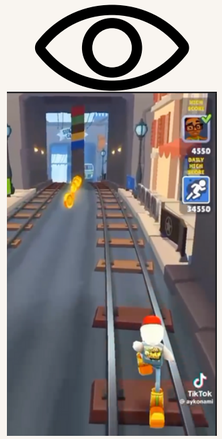
2. Fine Grained User Preference Graph

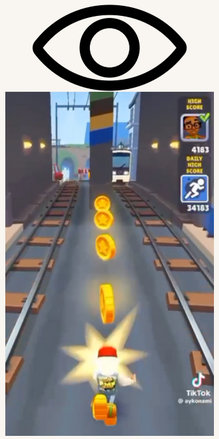








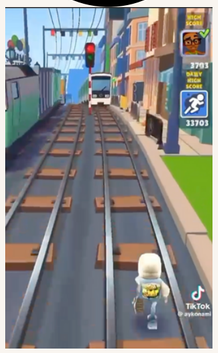
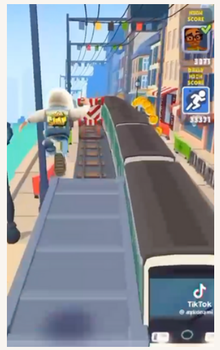
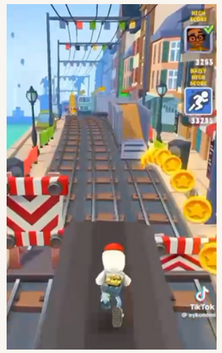


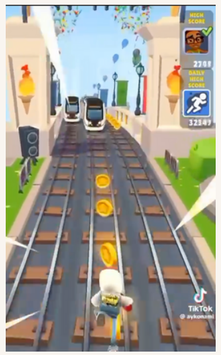
FRAME

2. Fine Grained User Preference Graph



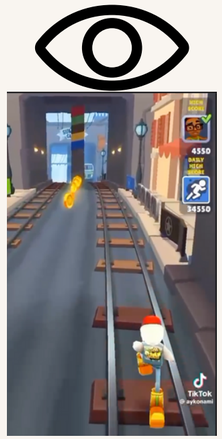


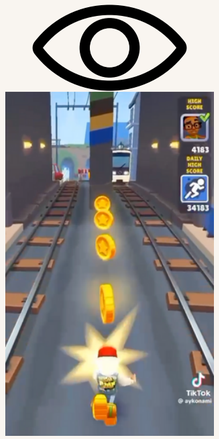




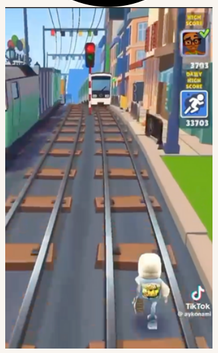
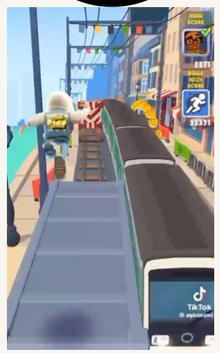
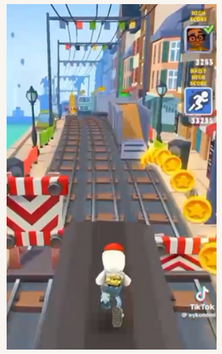


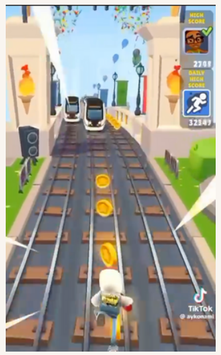
FRAME

2. Fine Grained User Preference Graph



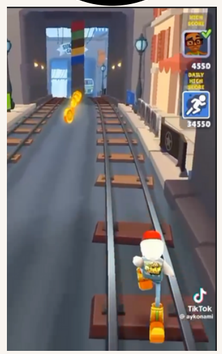
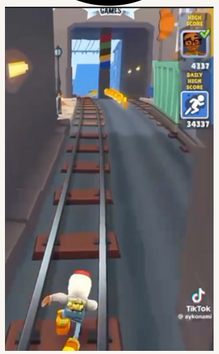
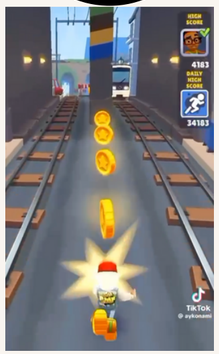
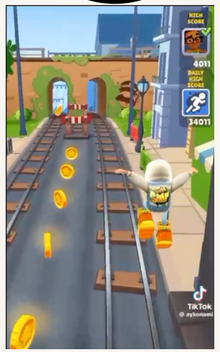
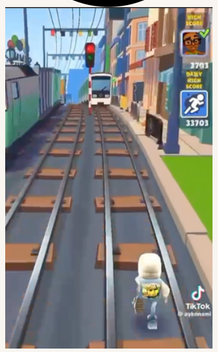
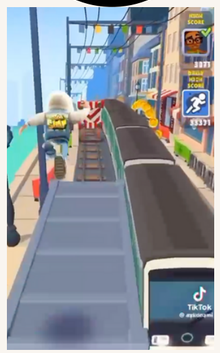
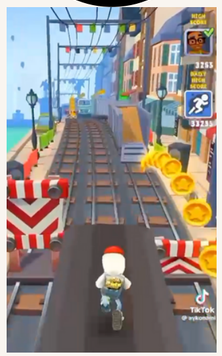


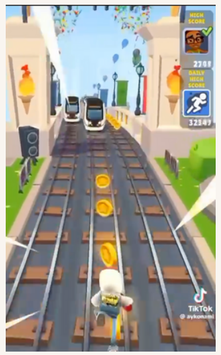






FRAME

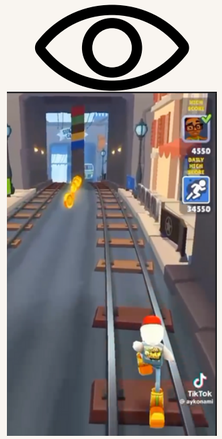
2. Fine Grained User Preference Graph

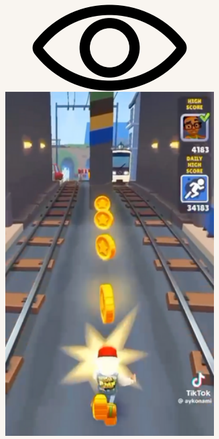








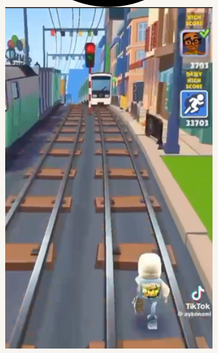
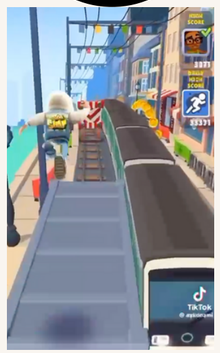
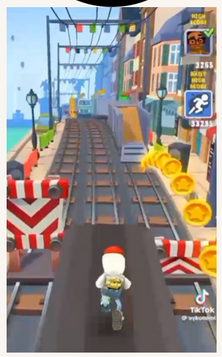
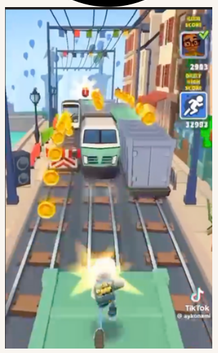
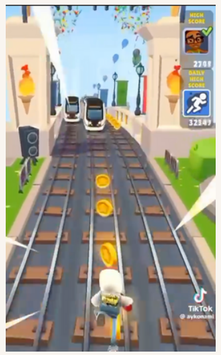


FRAME

2. Fine Grained User Preference Graph

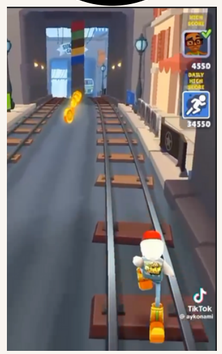
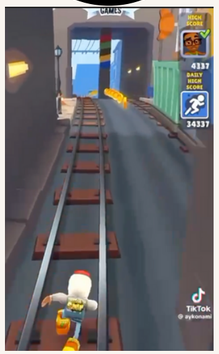
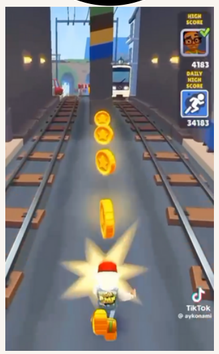
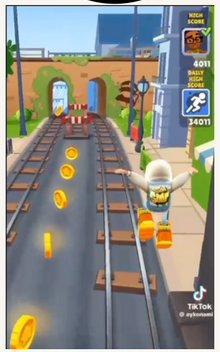
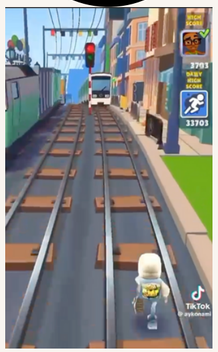
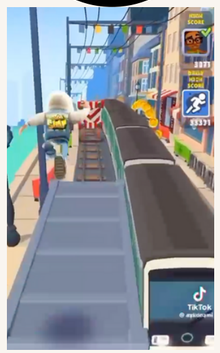
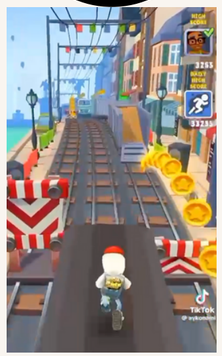
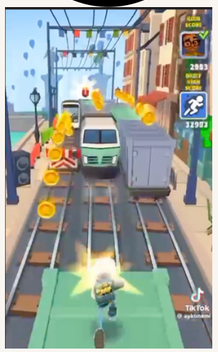
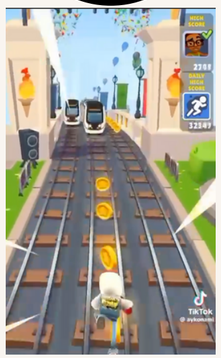




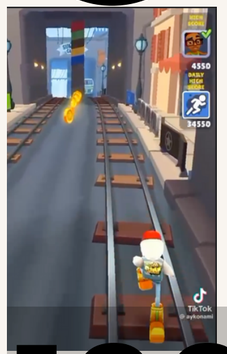
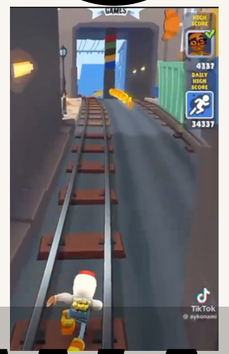
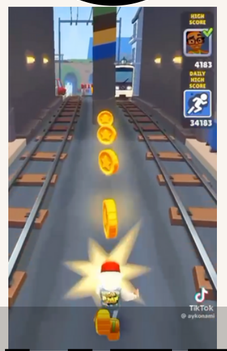
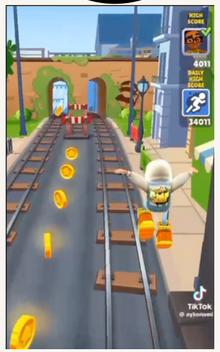
FRAME

2. Fine Grained User Preference Graph

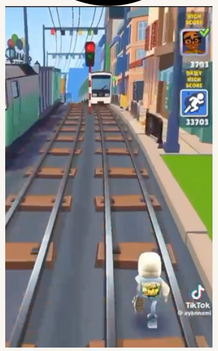
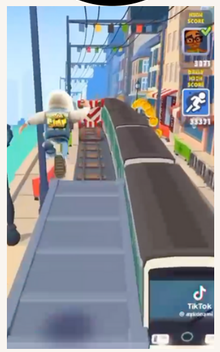
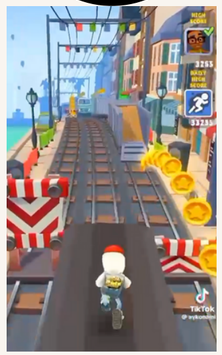
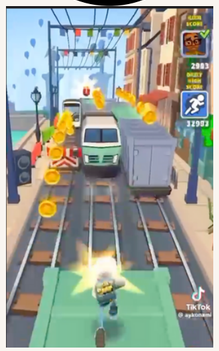
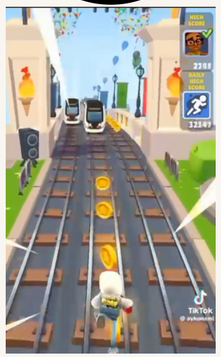










FRAME

2. Fine Grained User Preference Graph

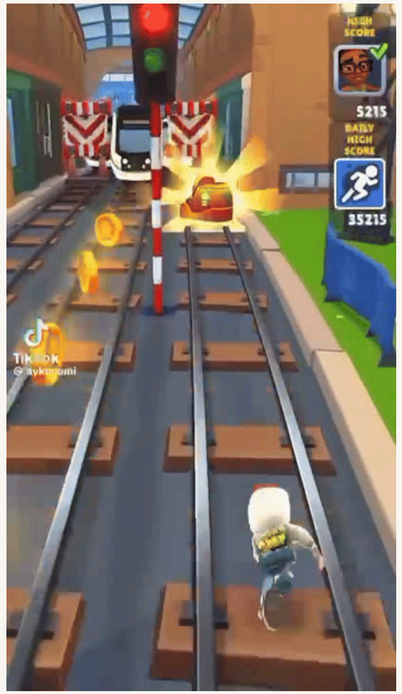





100% WATCHED

FRAME

2. Fine Grained User Preference Graph



Todos los clips de un video 100% visto son clasificados como positivos

$$C_u^p = \bigcup_{i=1}^{N_+^{(u)}} \{c_{i,1}^p, c_{i,2}^p, \cdots, c_{i,N_c}^p\},$$

$N_+^{(u)}$: cantidad de videos 100% vistos

N_c: cantidad de clips en el video i

$c_{i,1}^p$: clip 1 del video i (positivo)

FRAME

2. Fine Grained User Preference Graph

$$R_{ij}^{p} = \begin{cases} 1, & c_j \in C_{u_i}^{p}; \\ 0, & \text{otherwise}; \end{cases}$$

Matriz de Interacción Positiva

$$R_{ij}^{n} = \begin{cases} 1, & c_j \in C_{u_i}^{n}; \\ 0, & \text{otherwise}; \end{cases}$$

Matriz de Interacción Negativa

FRAME

2. Fine Grained User Preference Graph

$$\begin{bmatrix} 0.4 \\ -1.1 \\ 0.85 \\ 0.95 \end{bmatrix} \quad f_c$$

Se tienen Visual Embeddings
de cada clip

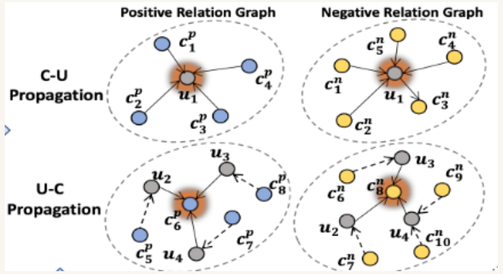
FRAME

2. Fine Grained User Preference Graph

$$\begin{bmatrix} 0.4 \\ -1.1 \\ 0.85 \\ 0.95 \end{bmatrix} \quad \mathbf{f_c}$$

Se tienen Visual Embeddings de cada clip

	C1	C2	C3	C4
U1	1	1	0	0
U2	0	1	1	0
U3	1	0	0	1



Se tienen grafos con los intereses de los usuarios

FRAME

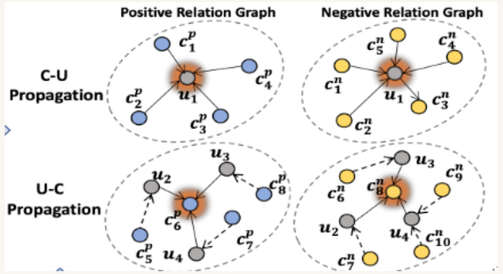
2. Fine Grained User Preference Graph

$$\begin{bmatrix} 0.4 \\ -1.1 \\ 0.85 \\ 0.95 \end{bmatrix} \quad \mathbf{f_c}$$

Se tienen Visual Embeddings de cada clip

	C1	C2	C3	C4
U1	1	1	0	0
U2	0	1	1	0
U3	1	0	0	1

Se tienen grafos con los intereses de los usuarios



¿Cómo se combinan estos elementos?

FRAME

2. Fine Grained User Preference Graph

GCN Layer

Graph Convolutional Network: Embeddings se obtienen por propagación al ir incorporando embeddings de los vecinos

Positive/Negative User Interest Embedding Matrix

$$\mathbf{H}_u^p = \sigma\left(\tilde{\mathbf{R}}^p \mathbf{H}_c^{(0)} \mathbf{W}_p^{(1)}\right)$$

$$\mathbf{H}_u^n = \sigma\left(\tilde{\mathbf{R}}^n \mathbf{H}_c^{(0)} \mathbf{W}_n^{(1)}\right)$$

σ : Función de Activación No-linear

$\tilde{\mathbf{R}}$: Matriz de Interacción User-Clip (normalizada)

$\mathbf{H}_c^{(0)}$: Clip Embedding Matrix

$\mathbf{W}^{(1)}$: Trainable Weight Matrix (propio del GCN)

FRAME

2. Fine Grained User Preference Graph

Ya se tienen matriz de Embedding con los intereses de los usuarios...

FRAME

2. Fine Grained User Preference Graph

Ya se tienen matriz de Embedding con los intereses de los usuarios...

Pero en la vida real nos puede interesar saber relaciones Clip-Clip

FRAME

2. Fine Grained User Preference Graph

Ya se tienen matriz de Embedding con los intereses de los usuarios...

Pero en la vida real nos puede interesar saber relaciones Clip-Clip

Si 2 clips están conectados al mismo usuario, esto es información útil: los clips pueden ser similares

FRAME

2. Fine Grained User Preference Graph

Para ello se lleva a cabo un Two-Hop Embedding propagation para obtener nuevos embeddings de clips:

$$\mathbf{H}_c^{p(1)} = \sigma\left(\left(\tilde{\mathbf{R}}^p\right)^\top \mathbf{H}_u^p \mathbf{W}_p^{(2)}\right)$$

$$\mathbf{H}_c^{n(1)} = \sigma\left(\left(\tilde{\mathbf{R}}^n\right)^\top \mathbf{H}_u^n \mathbf{W}_n^{(2)}\right)$$

σ : Función de Activación No-linear

$\tilde{\mathbf{R}}$: Matriz de Interacción User-Clip (normalizada)

$\mathbf{H}_u^p$ $\mathbf{H}_u^n$: Positive/Negative User Interest Embedding Matrix

$\mathbf{W}^{(2)}$: Trainable Weight Matrix de la segunda capa del GCN

FRAME

2. Fine Grained User Preference Graph

Debido a que se quitaron las preferencias de los usuarios de la matriz, ya no es relevante la separación entre clips positivos y negativos

$$\mathbf{H}_c^{(1)} = \mathbf{Mean}\left(\mathbf{H}_c^{p(1)}, \mathbf{H}_c^{n(1)}\right).$$

FRAME

2. Fine Grained User Preference Graph

Se tienen las siguientes matrices con embeddings de clips, las cuales se combinan mediante coeficientes:

$\mathbf{H}_c^{(0)}$: Clip Embedding Matrix

$\mathbf{H}_c^{(1)}$: Clip Embedding Matrix (con otra información)

$$\mathbf{H}_c = \alpha_0 \mathbf{H}_c^{(0)} + \alpha_1 \mathbf{H}_c^{(1)}$$

FRAME

2. Fine Grained User Preference Graph

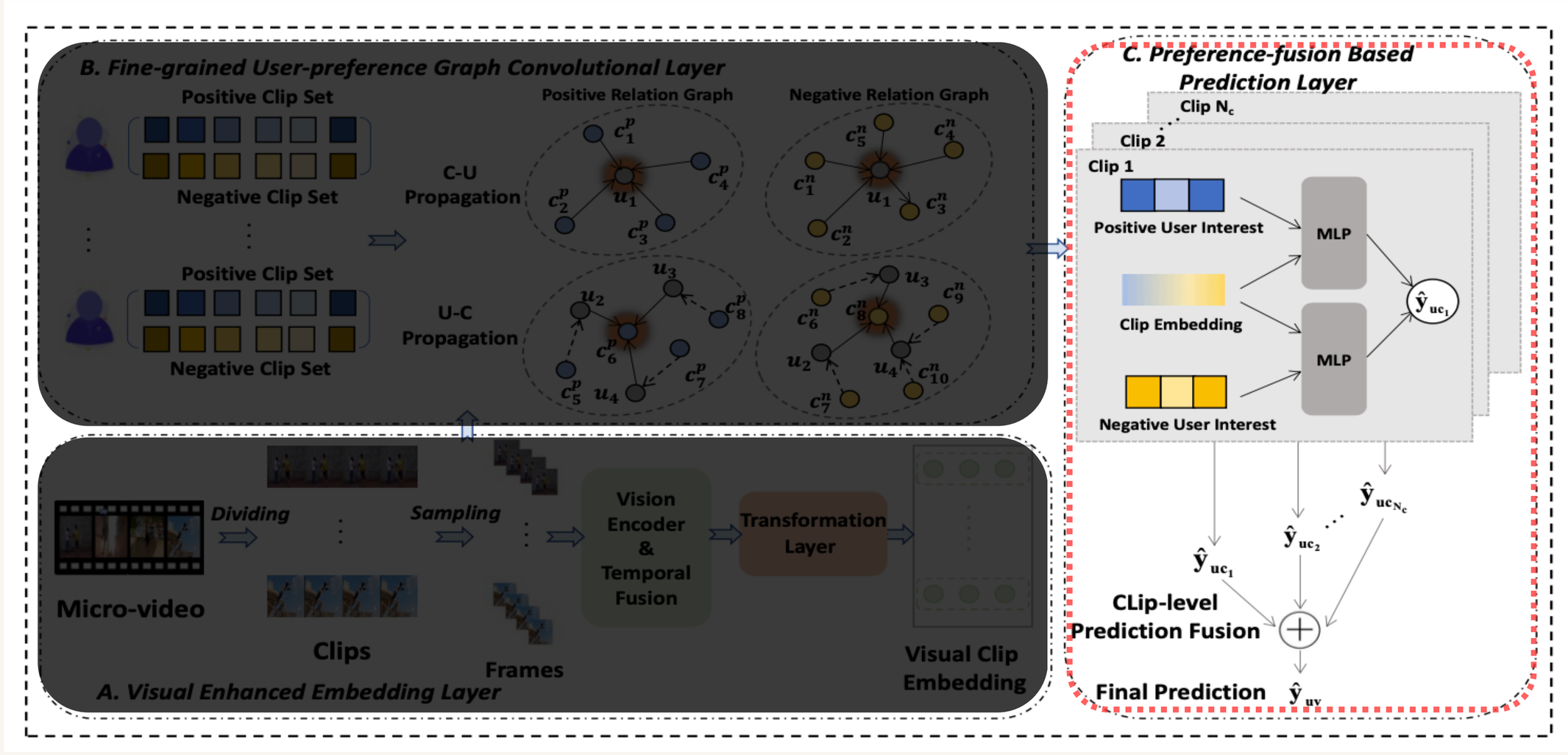
Con todo lo anterior, se logró obtener:

$\mathbf{H}_c$: Clip Embedding Matrix

$\mathbf{H}_u^p$: Positive User Interest Embedding Matrix

$\mathbf{H}_u^n$: Negative User Interest Embedding Matrix

FRAME


B. Fine-grained User-preference Graph Convolutional Layer
Positive Clip Set
Negative Clip Set
C-U Propagation
Positive Relation Graph
Negative Relation Graph
Positive Clip Set
Negative Clip Set
U-C Propagation
C. Preference-fusion Based Prediction Layer
Clip N_c
Clip 2
Clip 1
Positive User Interest
Clip Embedding
Negative User Interest
MLP
MLP
CLip-level Prediction Fusion
Final Prediction
Dividing
Sampling
Micro-video
Clips
Frames
Vision Encoder & Temporal Fusion
Transformation Layer
Visual Clip Embedding
A. Visual Enhanced Embedding Layer

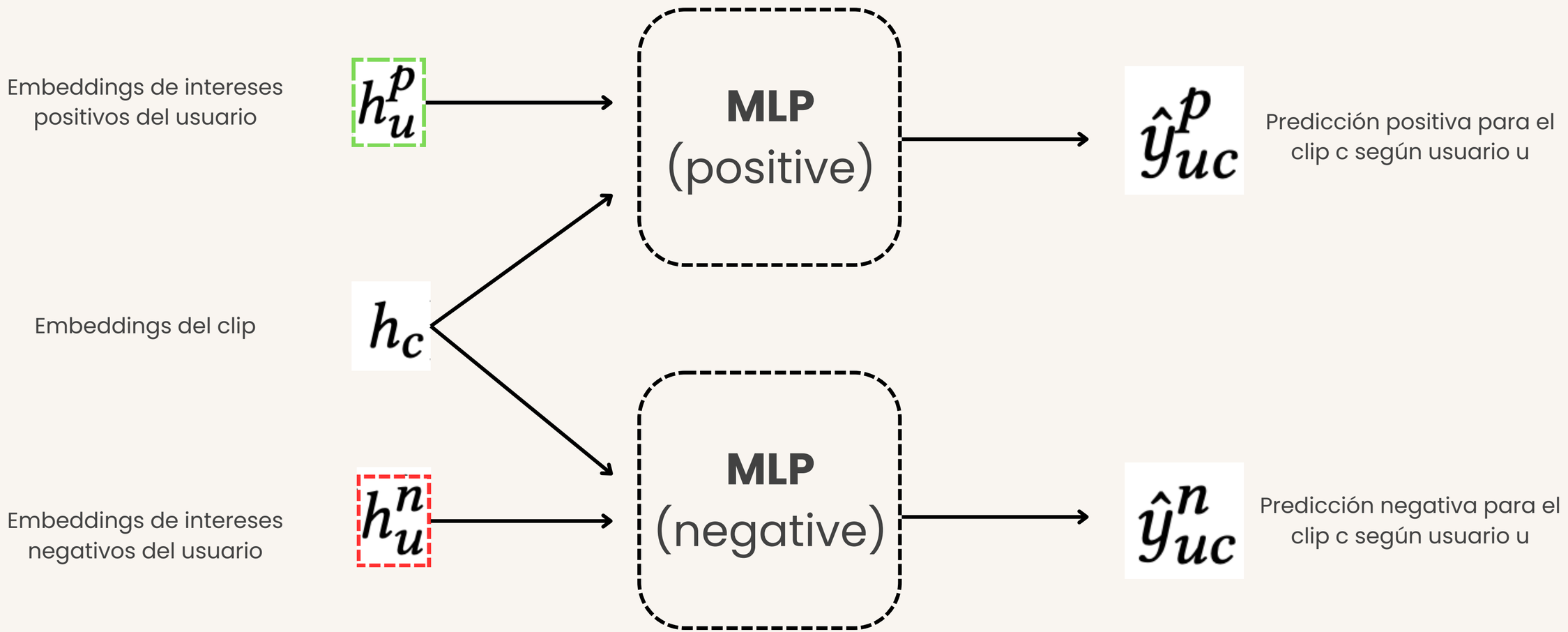
FRAME

$$\hat{y}_{uc}^{p} = \mathbf{MLP}^{p}(h_u^{p} \parallel h_c),$$

$$\hat{y}_{uc}^{n} = \mathbf{MLP}^{n}(h_u^{n} \parallel h_c).$$

3. Preference-Fusion Based Prediction Layer

Dado un usuario u y un clip c, se obtiene la siguiente predicción:



FRAME

3. Preference-Fusion Based Prediction Layer

Finalmente la predicción a nivel de clip esta dada por la siguiente formula:

$$\hat{y}_{uc} = \alpha_p \hat{y}^p_{uc} - \alpha_n \hat{y}^n_{uc}$$

Donde $\hat{y}_{uc}$ es la probabilidad de que el usuario u no se salte el clip c.

FRAME

3. Preference-Fusion Based Prediction Layer

Con un promedio simple entre todos los clips, se obtiene la **probabilidad de que el usuario u termine de ver el video v**

$$\hat{y}_{uv} = \frac{1}{N_c} \sum_{i=1}^{N_c} \hat{y}_{uc_i}$$

N_c : Cantidad de clips del video v

$\hat{y}_{uc_i}$: probabilidad de que el usuario u no salte el clip

EXPERIMENTO Y RESULTADOS

PREGUNTAS DE INVESTIGACIÓN

RQ1 - ¿Cómo es el **desempeño** del método **propuesto** en comparación con los métodos **anteriores**?

RQ2 - ¿Cómo afecta en el desempeño el utilizar **fine-grained** modeling en comparación con **coarse-grained** modeling?

DATASET

Se seleccionaron 2 dataset de distintas apps, con usuarios distintos

Dataset	#Users	#Videos	#Interactions
Micro-video-A	12,739	58,291	342,694
Micro-video-B	22,049	61,903	2,111,566

user_id
video_id
playing_time
duration_time
interaction_timestamp
like
follow
forward

DESEMPEÑO (RQ1)

Table 3: Overall performance comparison (all models have the same input for fair comparison; values are the average of three running instances with different random seeds; best baseline is marked with underline).

Method Type	Method	Micro-video-A						Micro-video-B					
		AUC	Logloss	Recall@3	NDCG@3	Recall@5	NDCG@5	AUC	Logloss	Recall@3	NDCG@3	Recall@5	NDCG@5
Feature-based recommender model	DeepFM	0.5834	0.8675	0.0563	0.1975	0.0832	0.2348	0.6566	0.8319	0.0785	0.2498	0.1064	0.3172
	NFM	0.6049	0.7835	0.0682	0.2374	0.0886	0.2591	0.6673	0.8125	0.0876	0.2631	0.1296	0.3319
	AutoInt	0.6279	0.7931	0.0694	0.2244	0.0907	0.2652	0.6865	0.7847	0.0842	0.2653	0.1287	0.3574
	DIFM	0.6338	0.7531	0.0749	0.2448	0.1104	0.3086	0.6913	0.7957	0.0861	0.2710	0.1256	0.3724
	AFN	0.6376	0.7438	0.0762	0.2683	0.1085	0.3106	0.6808	0.8045	0.0903	0.2794	0.1376	0.3830
Micro-video recommender model	ALPINE	0.6218	0.7692	0.0701	0.2324	0.0925	0.2886	0.6995	0.7834	0.0922	0.2748	0.1320	0.3783
	MTIN	0.6427	0.7228	0.0836	0.2715	0.1125	0.3327	0.7489	0.7647	0.1036	0.3081	0.1417	0.3938
Our model	**FRAME**	**0.7039**	**0.6732**	**0.1083**	**0.3219**	**0.1378**	**0.3722**	**0.7870**	**0.7619**	**0.1149**	**0.3296**	**0.1796**	**0.4358**

> *El método propuesto obtiene el mejor desempeño de manera consistente en comparación con los baselines.*

EFECTIVIDAD DE MODELO FINE-GRAINED CLIPPING (RQ2)

Table 4: Ablation study of the fine-grained clip dividing.

Dataset	Model	AUC	Logloss	Recall@3	NDCG@3
Micro-video-A	w/o clip dividing ($N_c = 1$)	0.6329	0.7658	0.0736	0.2715
	w/ clip dividng ($N_c = 4$)	0.6967	0.6903	0.1058	0.3145
	w/ clip dividng ($N_c = 8$)	**0.7039**	**0.6732**	**0.1083**	**0.3219**
Micro-video-B	w/o clip dividing ($N_c = 1$)	0.7318	0.8054	0.0910	0.2684
	w/ clip dividng ($N_c = 4$)	0.7739	0.7715	0.1127	0.3173
	w/ clip dividng ($N_c = 8$)	**0.7870**	**0.7619**	**0.1149**	**0.3296**

CONCLUSIONES Y FUTURO

1. **El feedback del usuario es continuo**: es por ello que un enfoque fine-grained al fijarse en clips puede obtener mejores resultados más que solo like/dislike.

2. FRAME demuestra ser **consistentemente mejor** en resultados que otros modelos coarse-grained

3. A futuro se piensa llevar a cabo experimentos con **mayor potencia computacional (clips más cortos)** y llevar a cabo A/B testing a partir de las recomendaciones entregadas por Frame

4. Como grupo creemos que se podría generar un **ensamble** que combine **FRAME con modelos coarse-grained** y experimentar según eso.

GRACIAS

REFERENCIAS

[1] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In Proceedings of the 27th ACM International Conference on Multimedia. 1437–1445
[2] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-scale Time-aware User Interest Modeling for Micro-video Recommendation. In MM. 3487–3495.
[3] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In MM. 1464–1472.
[4] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou, Zhiwei Xiong, and Yan Li. 2018. Temporal hierarchical attention at category-and item-level for micro-video click-through prediction. In MM. 1146–1153.
[5] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, and Chunyan Miao. 2021. SEMI: A Sequential Multi-Modal Information Transfer Network for E-Commerce Micro-Video Recommendations. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 3161–3171.
[6] Yujie Lu, Yingxuan Huang, Shengyu Zhang, Wei Han, Hui Chen, Zhou Zhao, and Fei Wu. 2021. Multi-trends Enhanced Dynamic Micro-video Recommendation. arXiv preprint arXiv:2110.03902 (2021)

REFERENCIAS

[7] IBM - Convolutional Neural Networks. https://www.ibm.com/es-es/topics/convolutional-neural-networks
[8] Thomas Kipf, Google Deepmind - Graph Convolutional Networks. https://tkipf.github.io/graph-convolutional-networks/
[9] scikit-learn. Neural Network Models (supervised) - https://scikit-learn.org/stable/modules/neural_networks_supervised.html
[10] Google developers. Machine learning. ROC and AUC - https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
[11] Geeks for Geeks - https://www.geeksforgeeks.org/binary-cross-entropy-log-loss-for-binary-classification/