

NeuroMLR

**Robust & reliable route
recommendation on road networks**

Matías Etcheverry, Martín González y Blanca Romero





Contexto



Fundamental para apps de navegación



La mejor ruta no siempre es la más corta



Métodos actuales no garantizan llegar a destino



Sufren al recomendar rutas no conocidas



NeuroMLR



Problema de recomendación

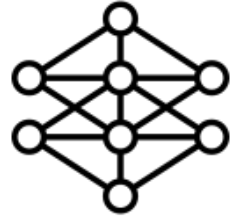
- Mejor ruta en red vial
- Predecir la ruta más probable
 - Trayectorias históricas
 - Tráfico
 - Topología del mapa



Contribución

- Mejor desempeño que modelos actuales
- Soluciona problemas actuales:
 - Adapta mejor a datos no vistos
 - Más probabilidad de alcanzar el destino

CssRnn

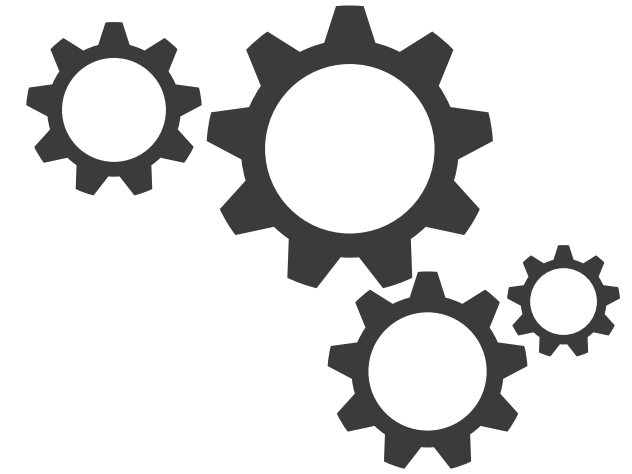


- Constrained State Space Recurrent Neural Network
- Secuencias de nodos
- Aprende patrones

DeepST



- Deep Spacial Transition
- Modelo probabilístico profundo
- Predice el siguiente segmento de la ruta

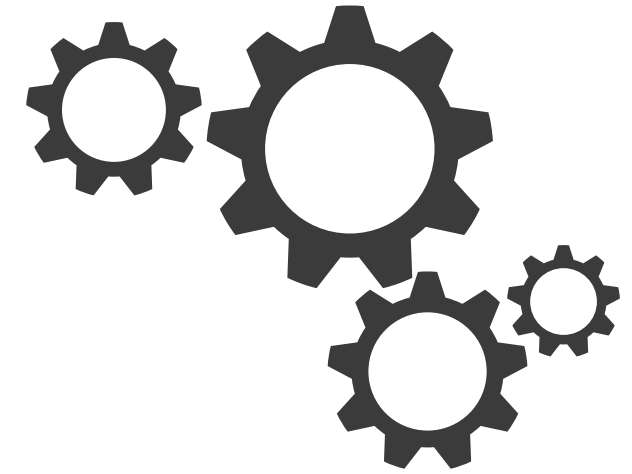


Estado del Arte y Marco Teórico

Limitaciones



- Generalizan mal a caminos nuevos
 - CnnRnn: representación fija de nodos
 - DeepST: No conoce probabilidades a nuevos segmentos
- No garantizan llegar al destino
 - Distintos destinos con mismas probabilidades



Estado del Arte y Marco Teórico

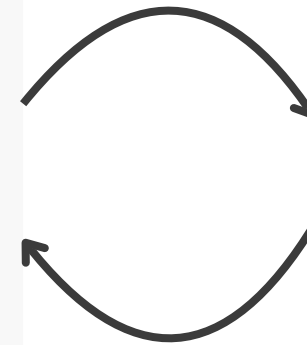
Solución



Búsqueda

Dijkstra
Greedy

Aseguran que se alcance el destino



Asignación de probabilidades

Red neuronal → Predecir la probabilidad de transición de cada nodo

Solución

Dado un grafo $G = (V, E, \delta, \tau_t)$

V = nodos (intersecciones)

E = aristas (segmentos de calle)

$\delta(e)$ = largo de segmento de calle

$\tau_t(e)$ = tiempo promedio por segmento de calle a la hora t

+ D un dataset con trayectos históricos sobre este grafo

Representación de tráfico

\mathbf{r}_t = Vector de velocidades a la hora t

- vector de tamaño $|E|$



top **5%** de caminos más frecuentados



PCA



\mathbf{z}_t = Representación del tráfico a la hora t

- vector de tamaño k

Representación de nodos

Lipschitz embedding: $\mathbf{v}()$



Tomar $\{a_1, \dots, a_k\}$, k
nodos de V al azar



$d(u, v)$ = distancia entre nodos u y v



$\mathbf{v}(u) = [x_1, \dots, x_k]$ donde $x_i = d(u, a_i)$

Modelo - GCN

Generalizar para
nodos no vistos

Propósito:
Afinar Lipschitz embeddings

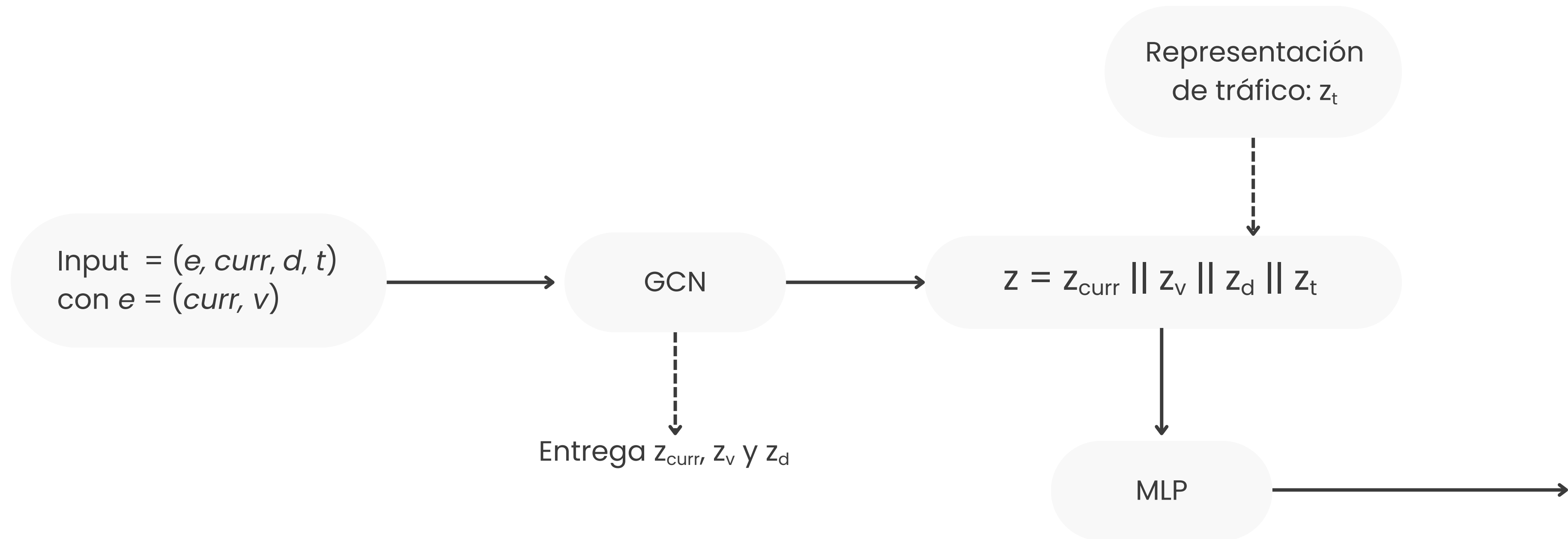
Los embeddings $\mathbf{v}(u)$ se
usan como input inicial

$$h_u^l = \sigma \left(\mathbf{W}_l \sum_{v \in N(u) \cup u} \frac{h_v^{l-1}}{\sqrt{(|N(v)| + 1)(|N(u)| + 1)}} \right)$$

Cada capa genera \mathbf{h}_u^l
 $N(u)$ = vecinos del nodo u en el grafo

Última capa $\rightarrow \mathbf{h}_u^L = \mathbf{z}_u$
nuevo embedding

Modelo



Modelo

softmax sobre todas las
otras posibles transiciones
desde *curr*

$$Q(e \mid curr, d, t; \theta)$$

$f(v, curr, d, t)$
no normalizado

Todo el modelo se entrena
con el dataset D

$\Theta \rightarrow$ *cross entropy* loss

Evaluación/Resultados

Métricas utilizadas:

$$\text{Precision} = \frac{\sum_{e \in (\mathcal{R} \cap \mathcal{R}^*)} \delta(e)}{\sum_{e \in \mathcal{R}^*} \delta(e)},$$

$$\text{Recall} = \frac{\sum_{e \in (\mathcal{R} \cap \mathcal{R}^*)} \delta(e)}{\sum_{e \in \mathcal{R}} \delta(e)}$$

$$\text{Reachability} = \begin{cases} 1 & \text{if } d = d^* \\ 0 & \text{otherwise} \end{cases},$$

$$\text{Reachability Distance} = \textit{Haversine}(d, d^*)$$

Evaluación/Resultados

| | Precision (%) | Recall (%) | Reachability (%) | Reachability Distance (km) |
|------------|---------------|------------|------------------|----------------------------|
| NeuroMLR-D | 74.0% | 66.4% | - | - |
| NeuroMLR-G | 69.8% | 64.4% | 98.1 | 0.02 |
| CssRnn | 48.7% | 57.7% | 79.0% | 1.0 |
| DeepSt | 59.9% | 31.8% | 7.8% | 1.9 |
| SP | 55.9% | 46.8% | - | - |
| QP | 46.6% | 40.9% | - | - |

Tabla 1: Métricas de desempeño sobre distintos métodos

Impacto de la distancia de viaje

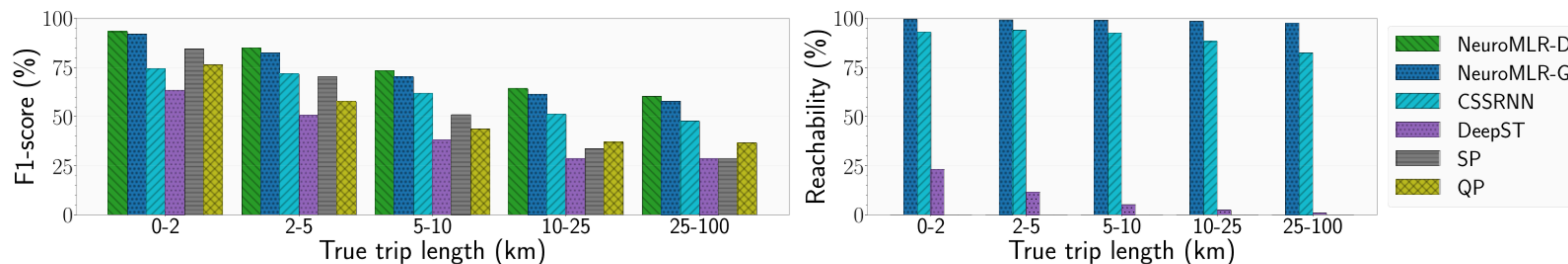


Figura 3: Impacto de distancia del viaje en puntaje F1 y porcentaje de nodos de destino alcanzados dentro del set de datos de Beijing

Desempeño en data no vista

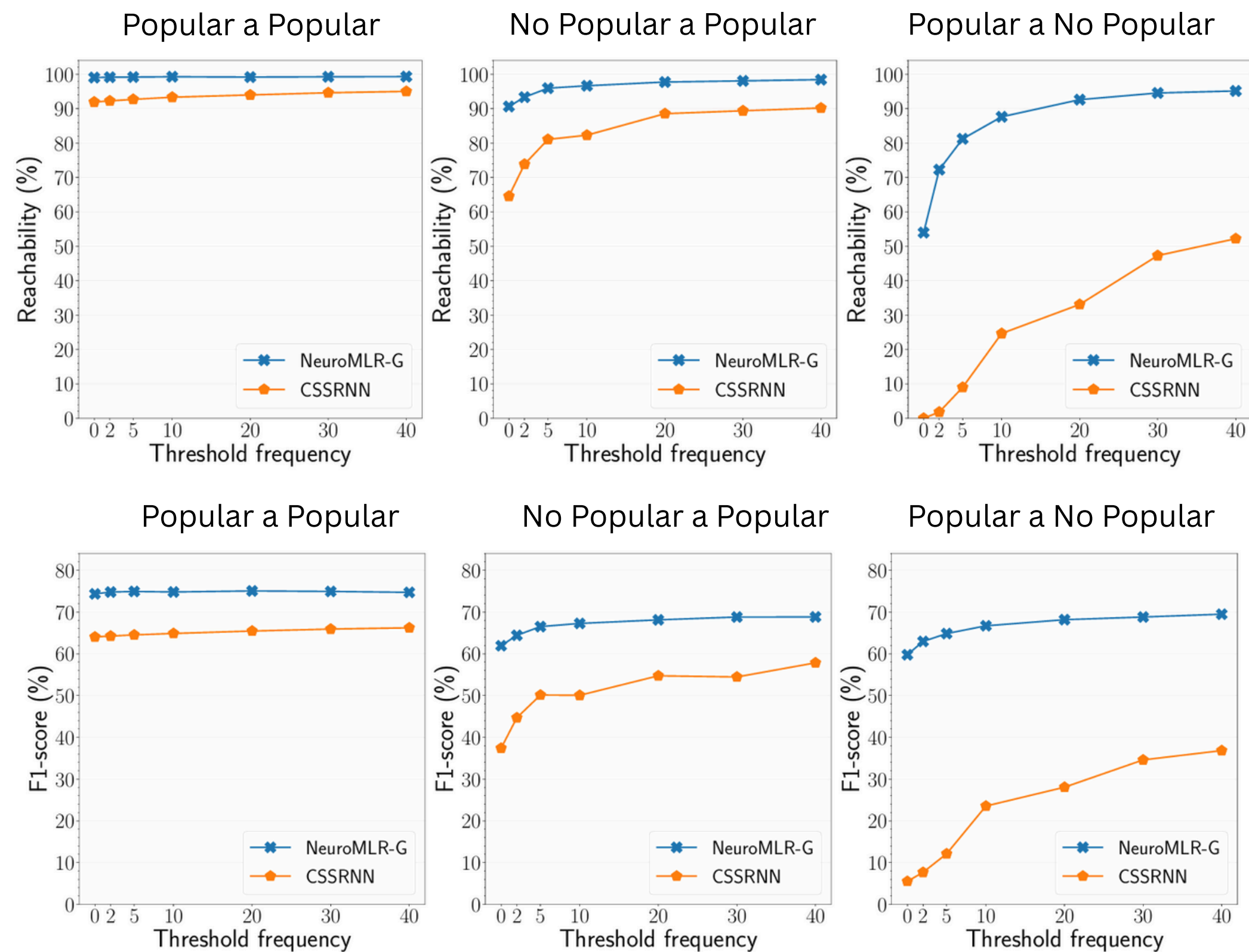


Figura 1: Variación de los nodos alcanzados y del puntaje F1 con el umbral de frecuencia para diferentes categorías de popularidad

Desempeño con distintos volúmenes de datos

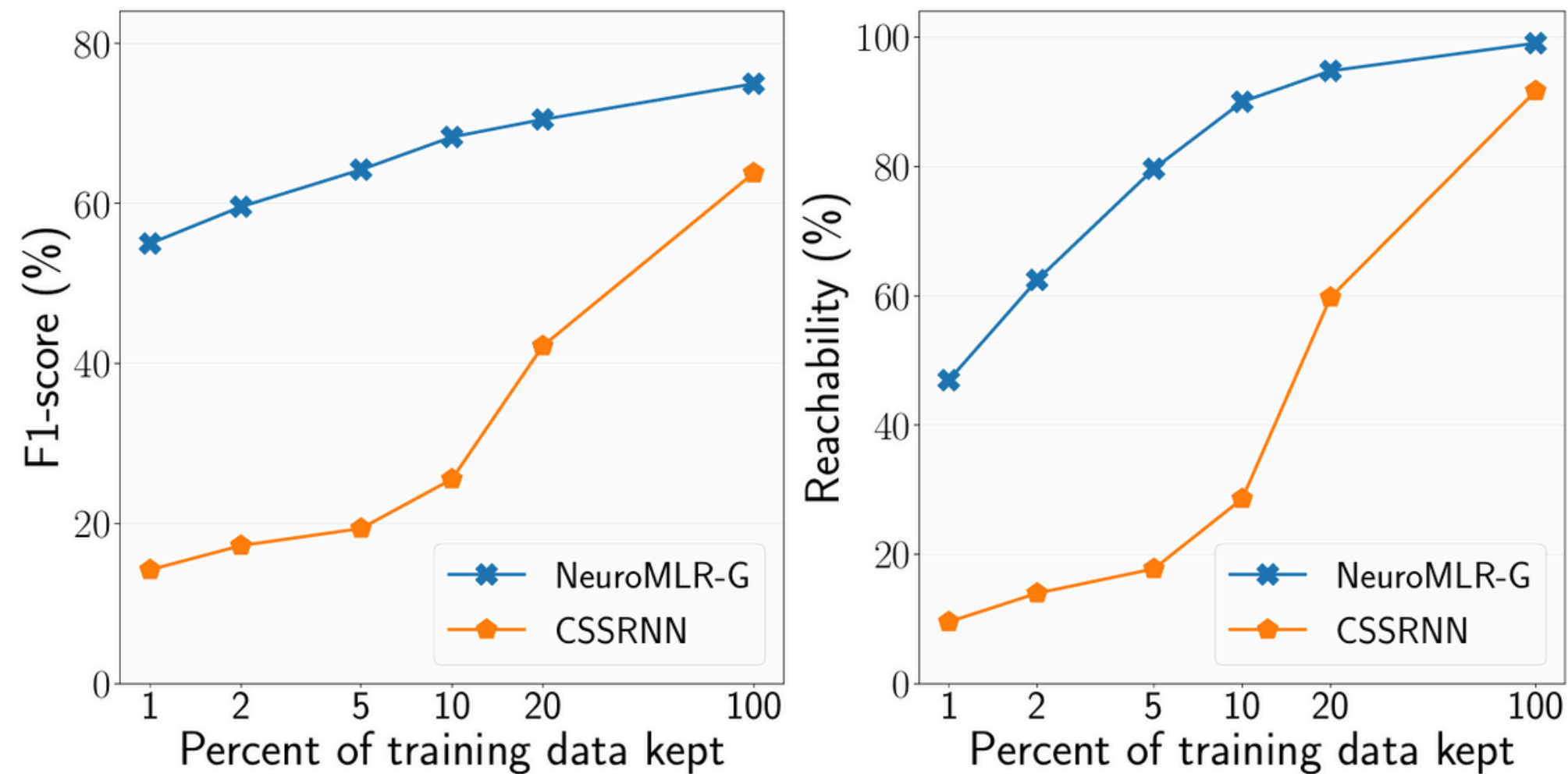


Figura 2: Desempeño de NeuroMLR-G y CsrRnn con diferentes porcentajes de los datos de aprendizaje

Tiempos de inferencia

| Algorithm | Chengdu | Porto | Harbin | Beijing | CityIndia |
|------------|-------------|-------------|-------------|-------------|-------------|
| NEUROMLR-D | 11.38 | 10.11 | 7.62 | 1.08 | 0.28 |
| NEUROMLR-G | 5612 | 4539 | 3286 | 2084 | 1522 |
| CSSRNN | 4425 | 1684 | 1355 | 522 | 467 |

Tabla 2: Viajes procesados por segundo

Conclusiones

Explicabilidad del modelo

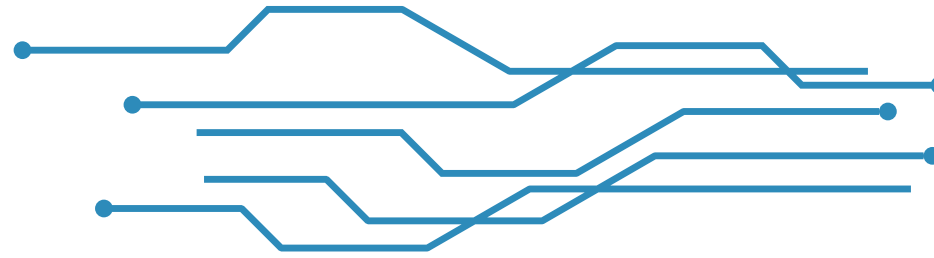
- El usuario no sabe por qué se le recomienda cierta ruta.

Código no ejecutable

- Existen dependencias conflictivas y clases que no existen.

Posible método: RL

- Aprendizaje de secuencia de decisiones
- Estados, acciones, recompensas, objetivo y restricciones



NeuroMLR

**Robust & reliable route
recommendation on road networks**

Matías Etcheverry, Martín González y Blanca Romero

