Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding

Yushun Dong, Sihang Jiang, Simon Zhu, Hanzhi Zhou

UVA ENGINEERING

ROADMAP

1

Background & Motivation

What problem to solve?
Why need a new model?

2

Methodology

What's our model?
How does it solve the problem?

3

Experiment Design

How to evaluate?
What're the datasets and baselines?

4

Results & Other Conclusions

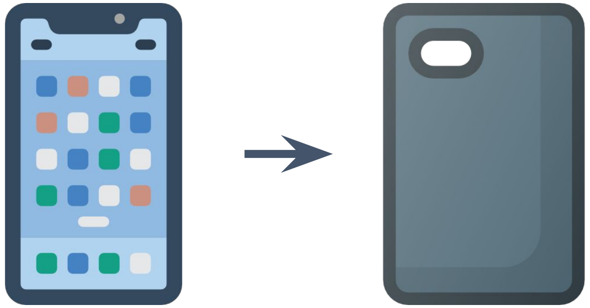
How does our model perform?
Future works

Background & Motivation

Top-N sequential recommendation captures customers' sequential patterns aside from general behavior

Customers' General Preferences vs. Sequential Patterns

- e.g: Buying iPhone accessory after buying an IPhone.

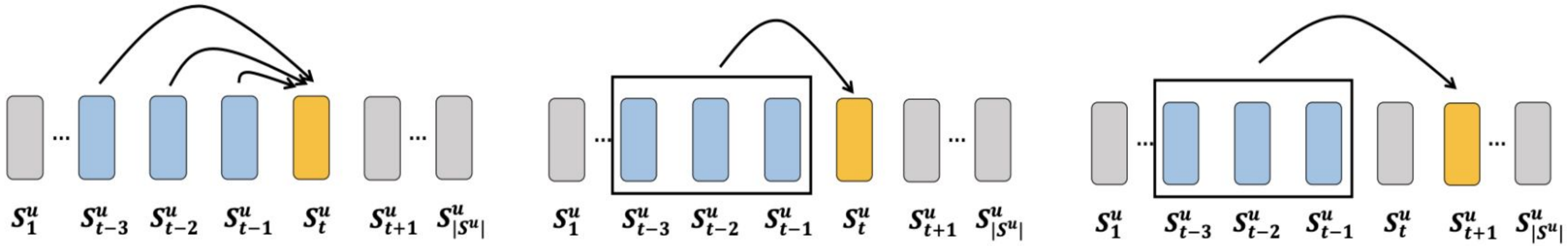


What Top-N sequential recommendation does:

- **recommends each user N items that maximize his/her future needs**, by considering both general preferences and sequential patterns. Unlike conventional top-N recommendation, top-N sequential recommendation **models the user behavior as a sequence of items**, instead of a set of items.

Current sequential recommendation systems leave out union-level and skip patterns

- The current Markov Chain models only point-level sequential patterns.

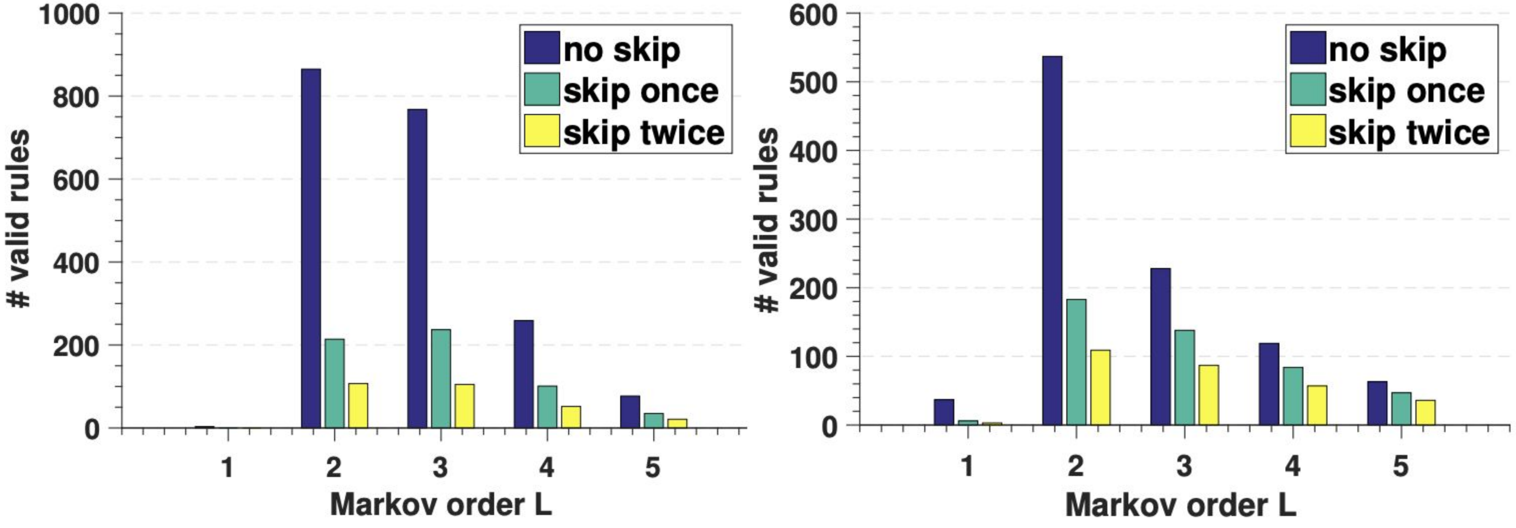


(a) point-level (b) union-level, no skip (c) union-level, skip once

Figure 1: An example of point and union level dynamic pattern influences, the order of Markov chain $L = 3$

Data exploration reveals common union-level and skip patterns

One Rule: $(\mathcal{S}^u_{t-L}, \cdots, \mathcal{S}^u_{t-2}, \mathcal{S}^u_{t-1}) \rightarrow \mathcal{S}^u_t.$



(a) MovieLens (b) Gowalla

Figure 2: The number of association rules vs L and skip steps. The minimum support count = 5 and the minimum confidence = 50%.

New model (Caser) gets the best of both worlds

ConvolutionAl Sequence Embedding Recommendation Model (Caser)

- **Caser models both users' general preferences and sequential patterns**, and generalizes several existing state-of-the-art methods in a single unified framework.
- **Caser capture sequential patterns at point-level, union-level, and of skip behaviors** with horizontal and vertical convolutional filters.

Methodology

How can we build a better model?

Questions to be answered:

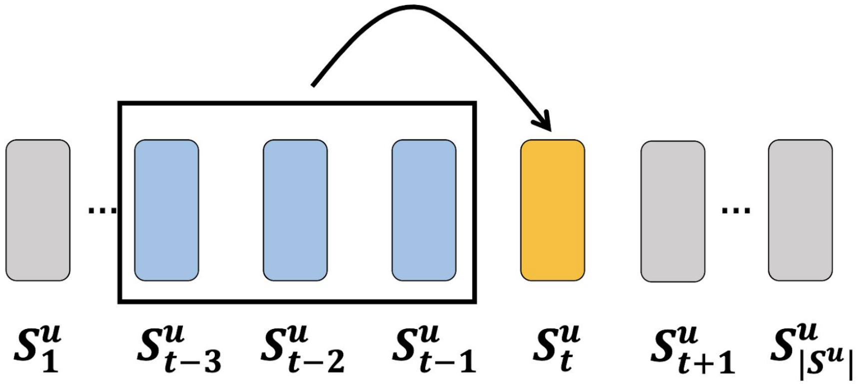
- How to model **union-level** influences?
- How to model **point-level** influences?
- How to incorporate **skip behaviors** in the model?

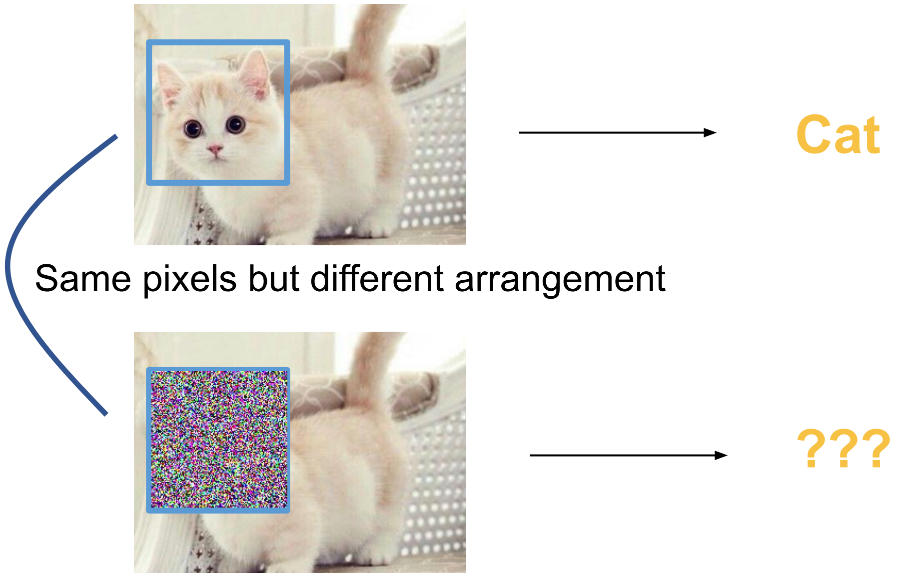
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

A basic intuition of union-level influence is that the **union exerts influence as a whole**, which is similar to the localized characteristic of an image.



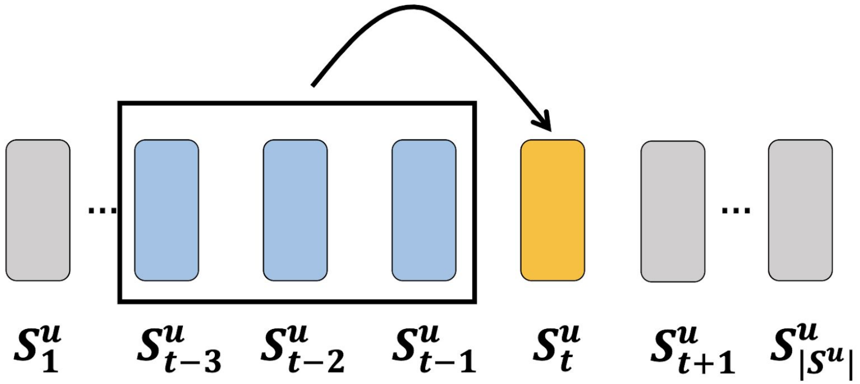


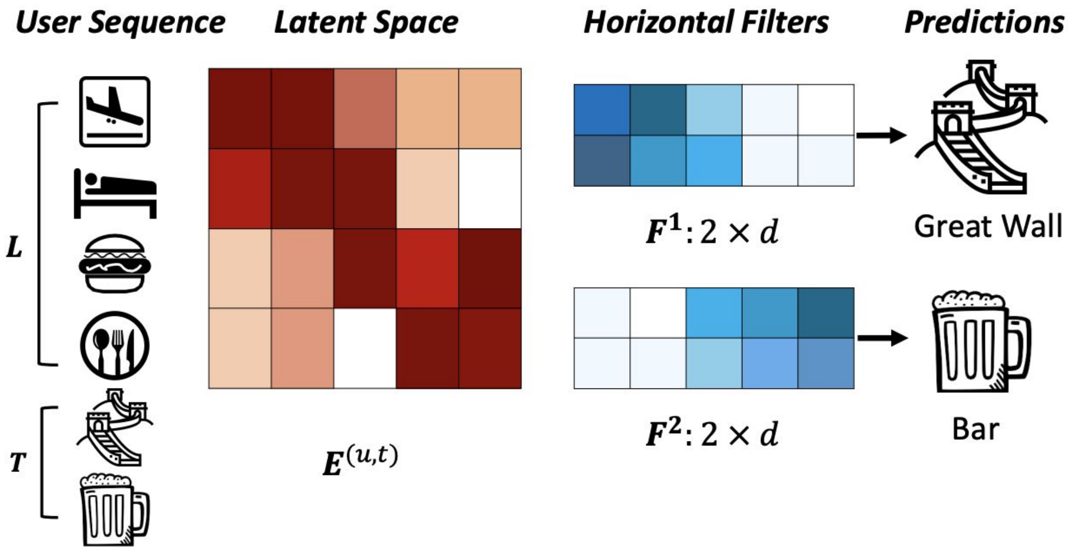
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

Such similarity motivates us to utilize convolutional kernel (i.e., filters) to extract such localized union information.

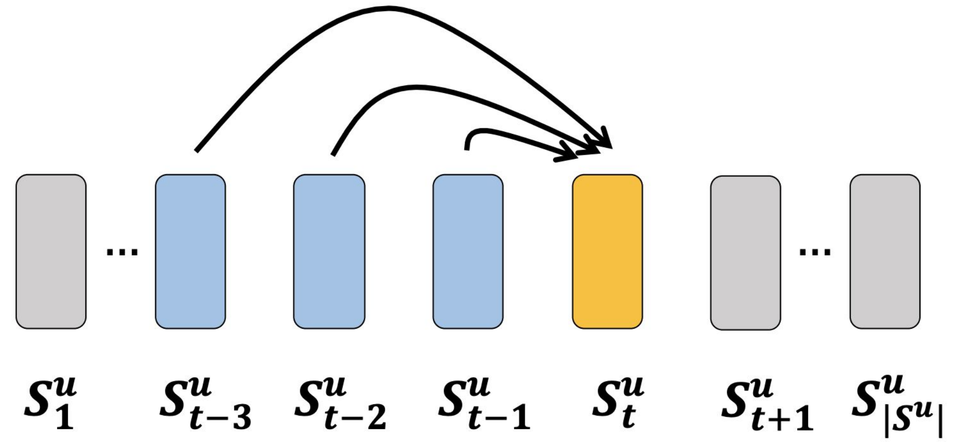




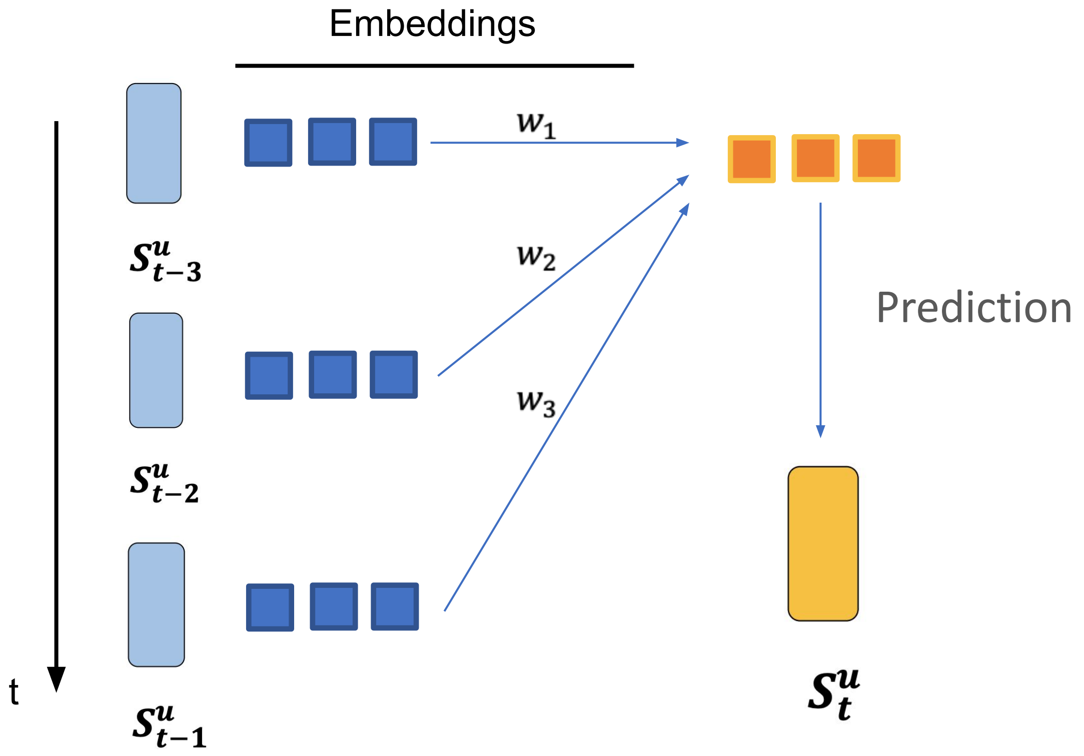
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?



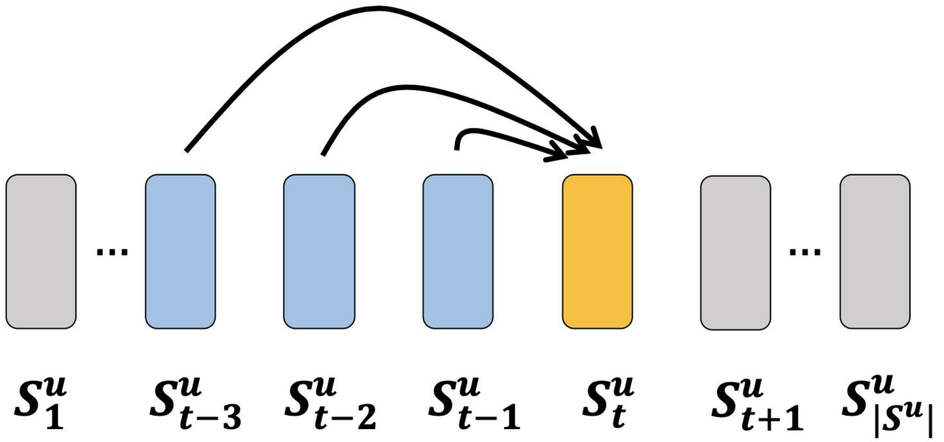
Point-level can be modeled using traditional weighted-summing.



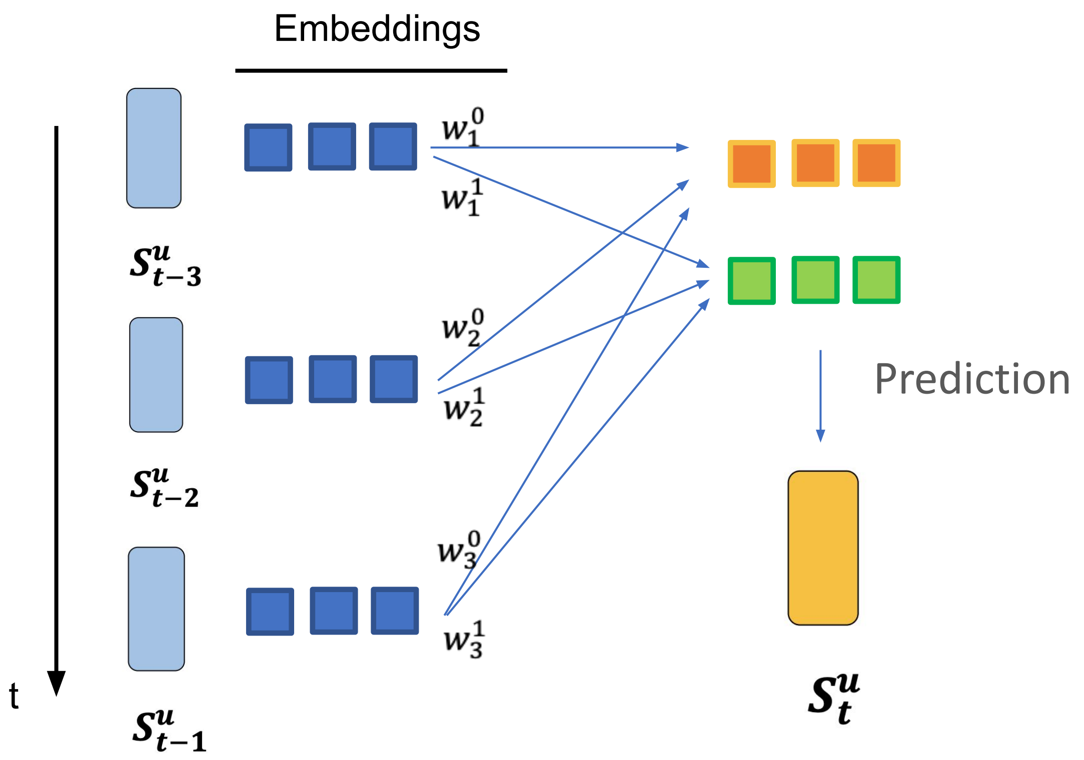
How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

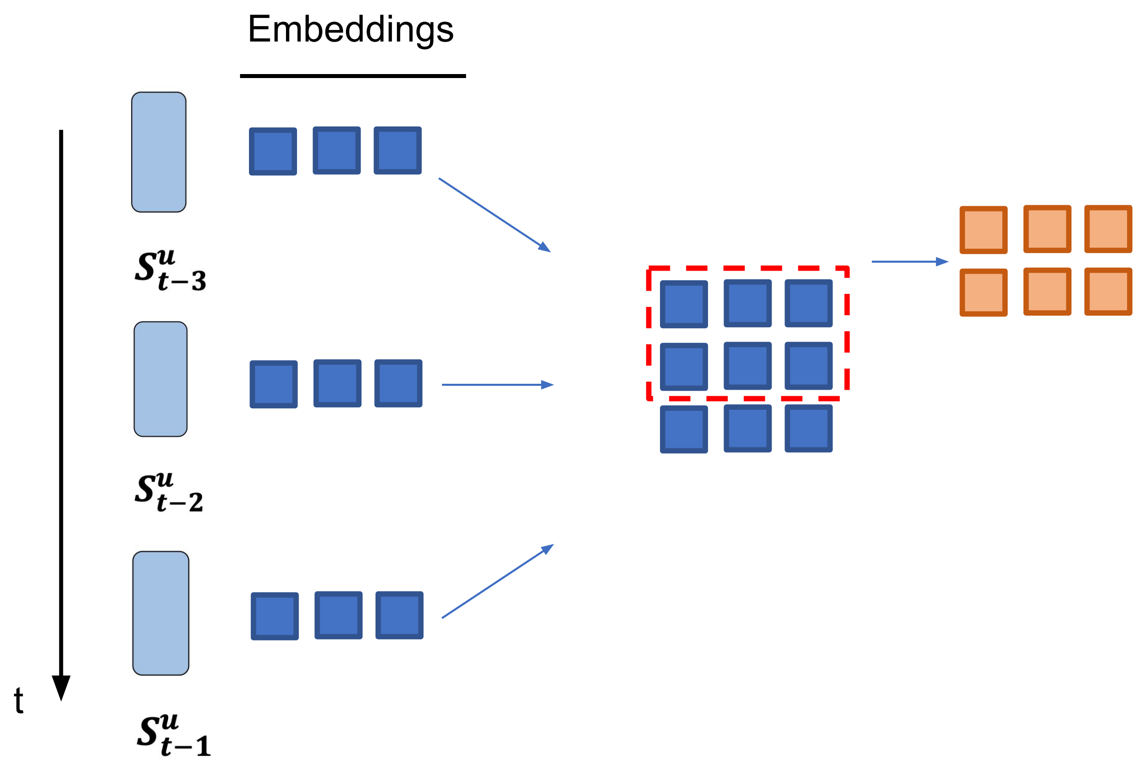


Point-level can be modeled using traditional weighted-summing.



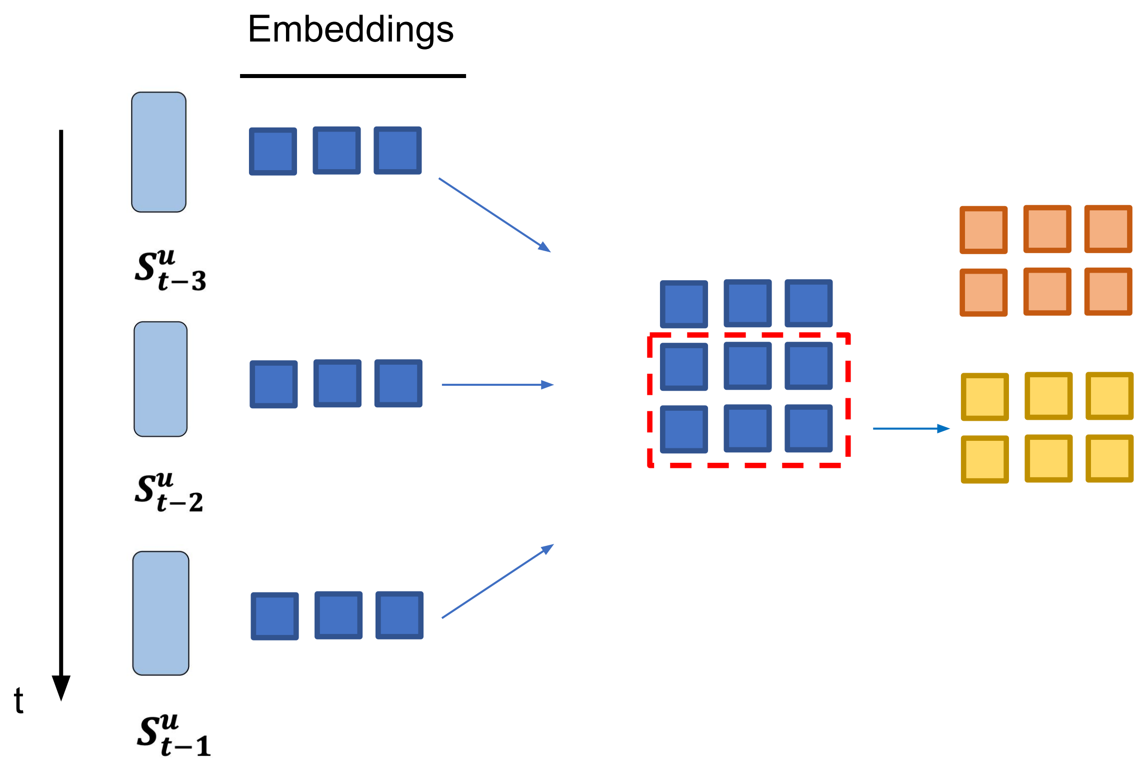
How can we build a better model?

Process overview:



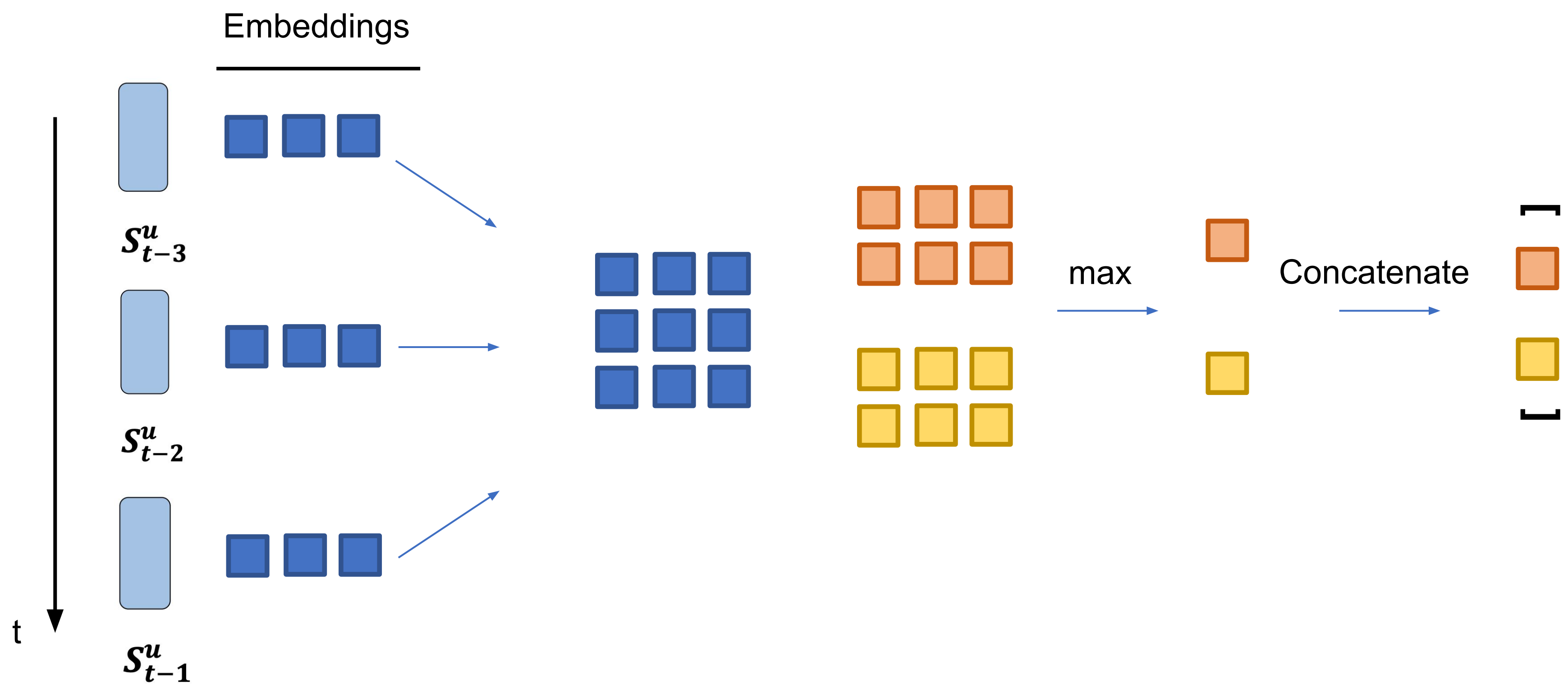
How can we build a better model?

Process overview:



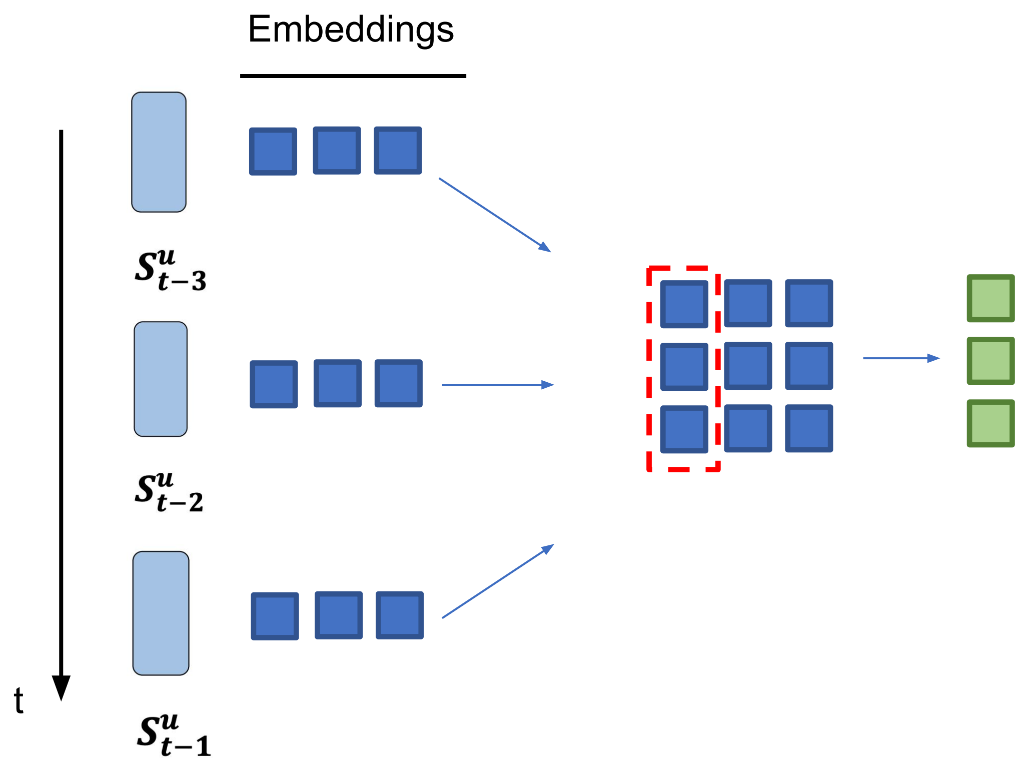
How can we build a better model?

Process overview:



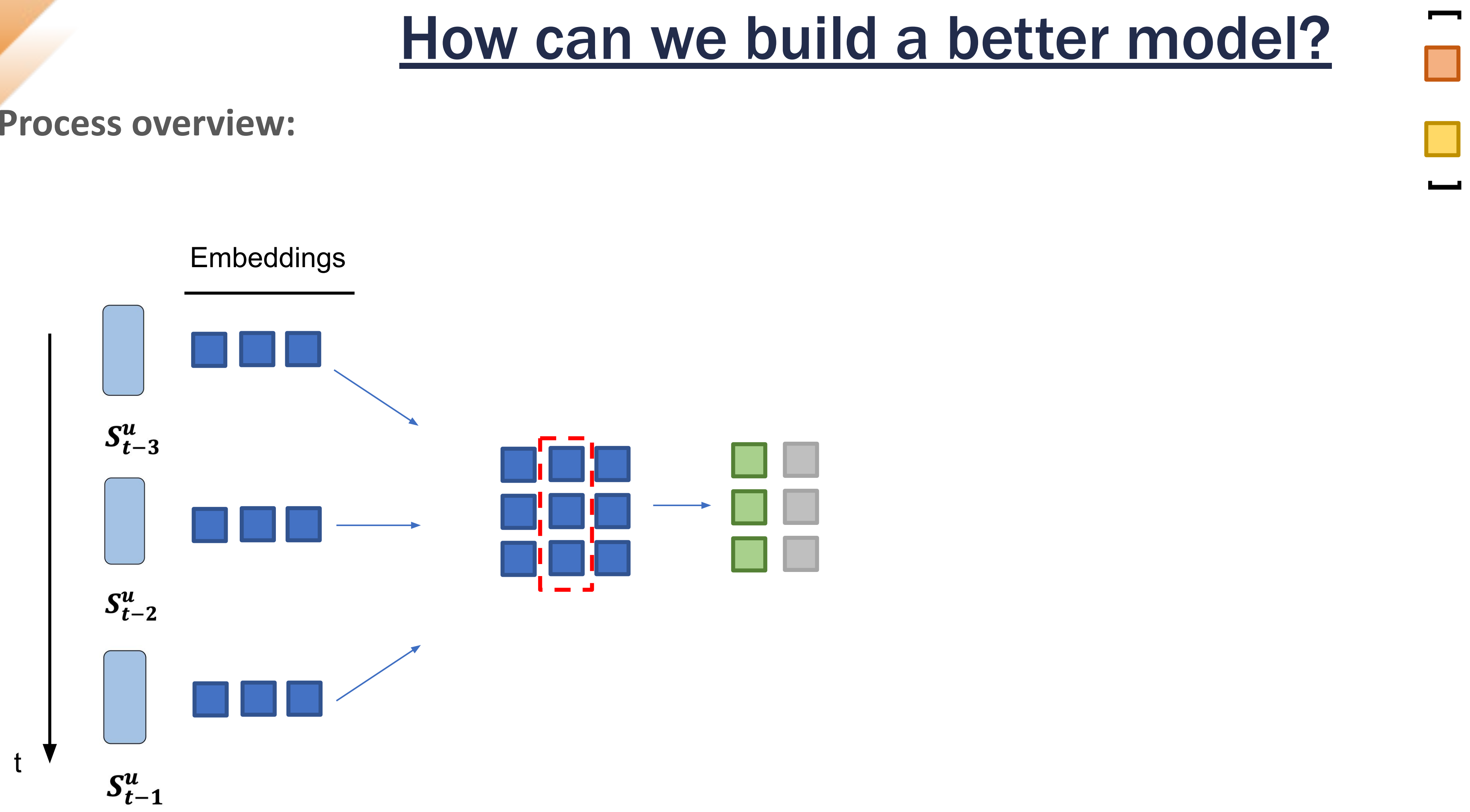
How can we build a better model?

Process overview:



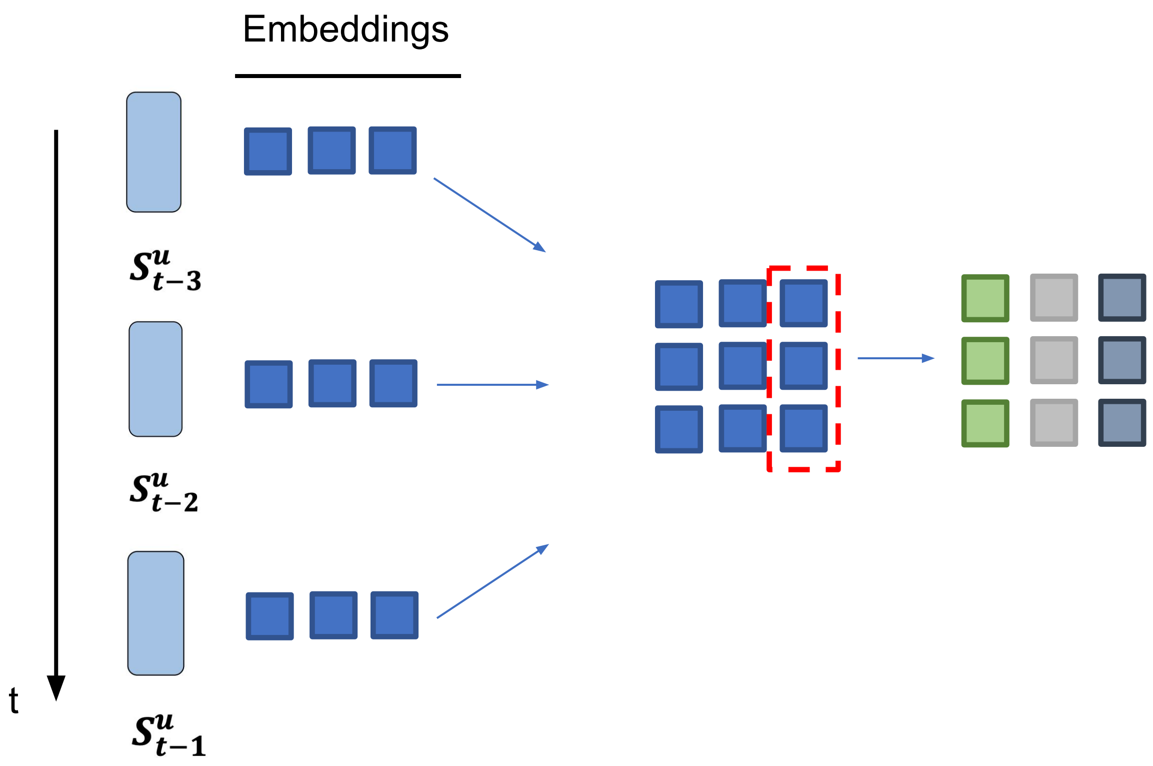
How can we build a better model?

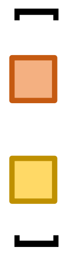
Process overview:



How can we build a better model?

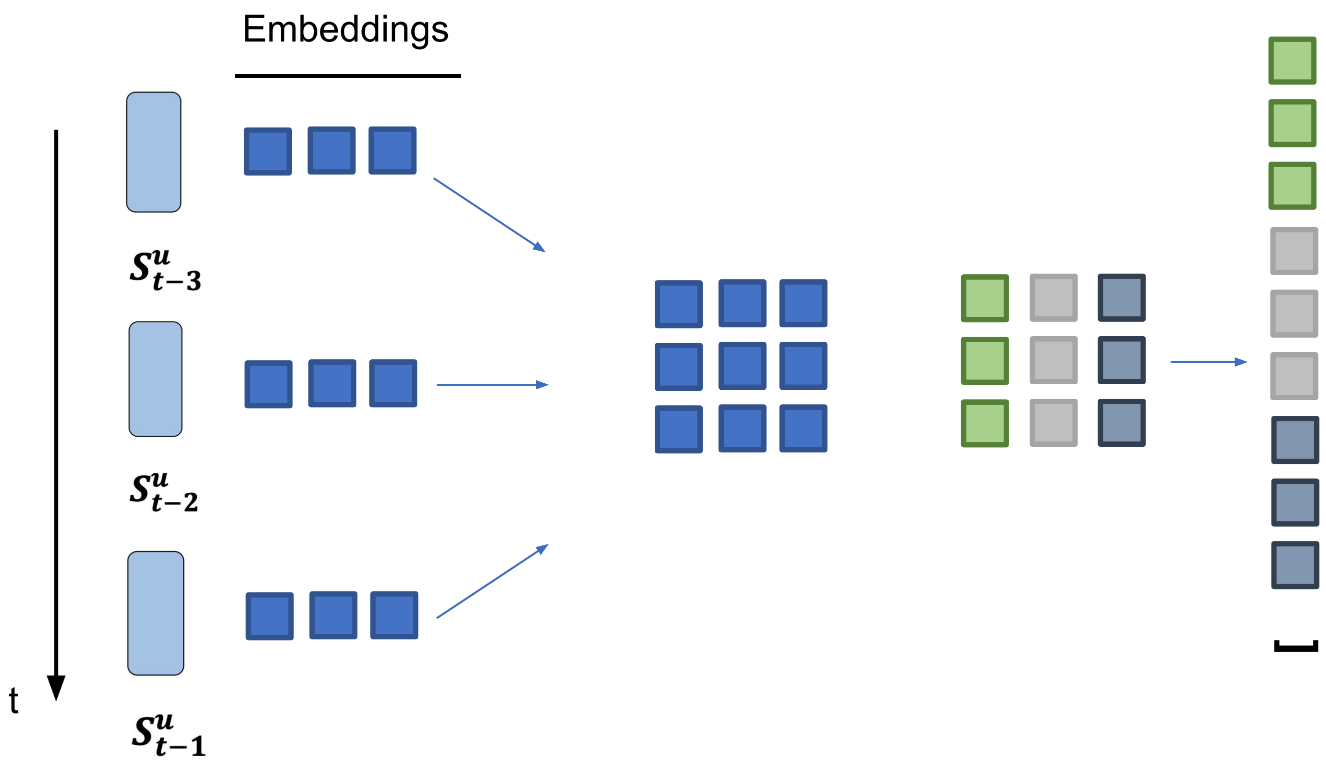
Process overview:





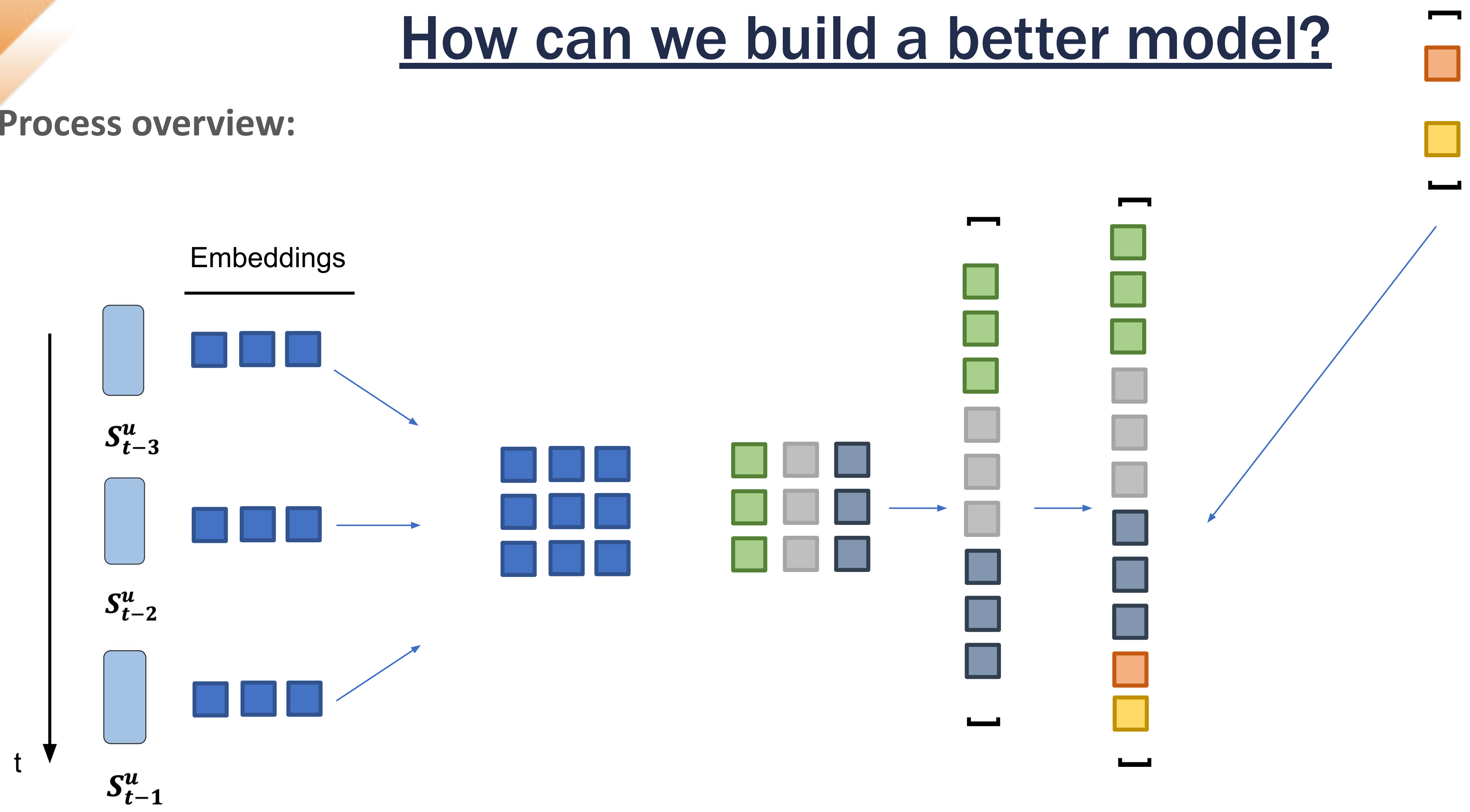
How can we build a better model?

Process overview:



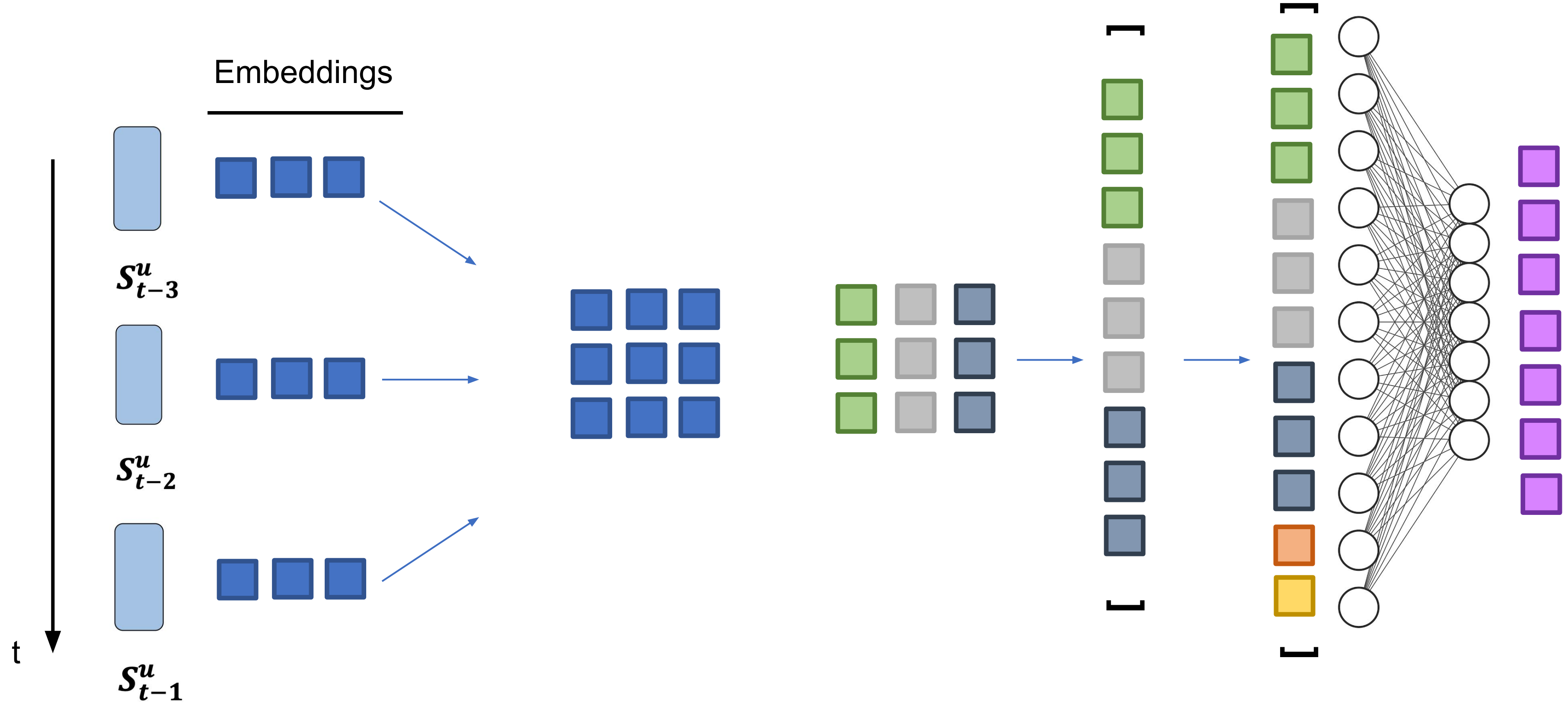
How can we build a better model?

Process overview:



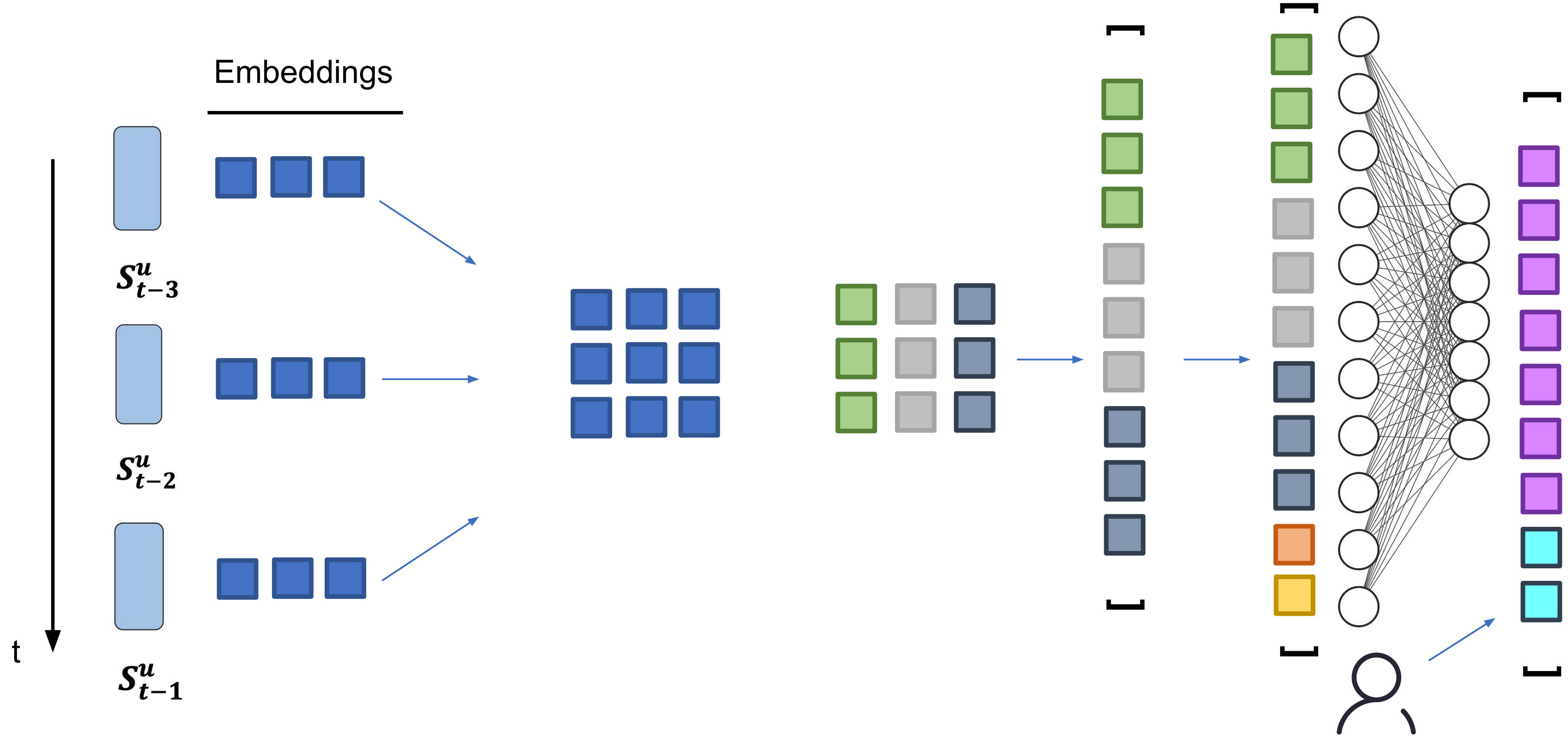
How can we build a better model?

Process overview:



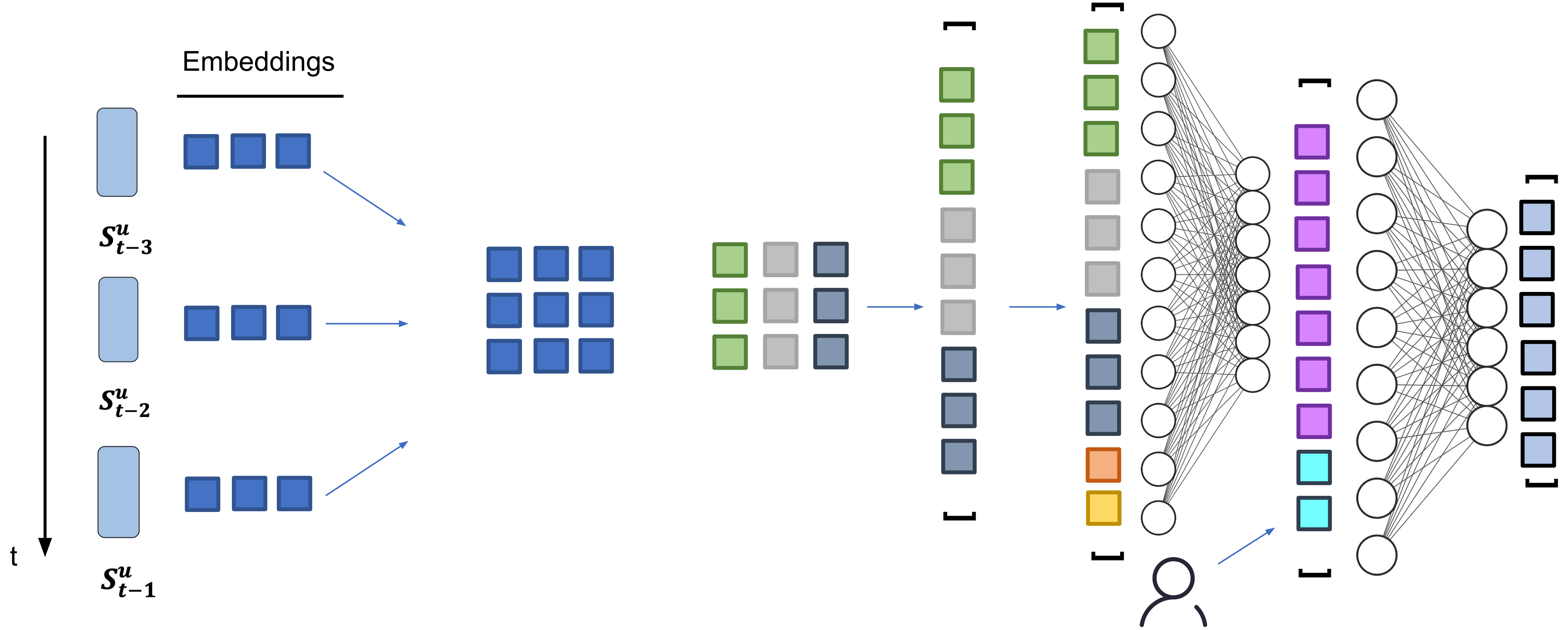
How can we build a better model?

Process overview:



How can we build a better model?

Process overview:

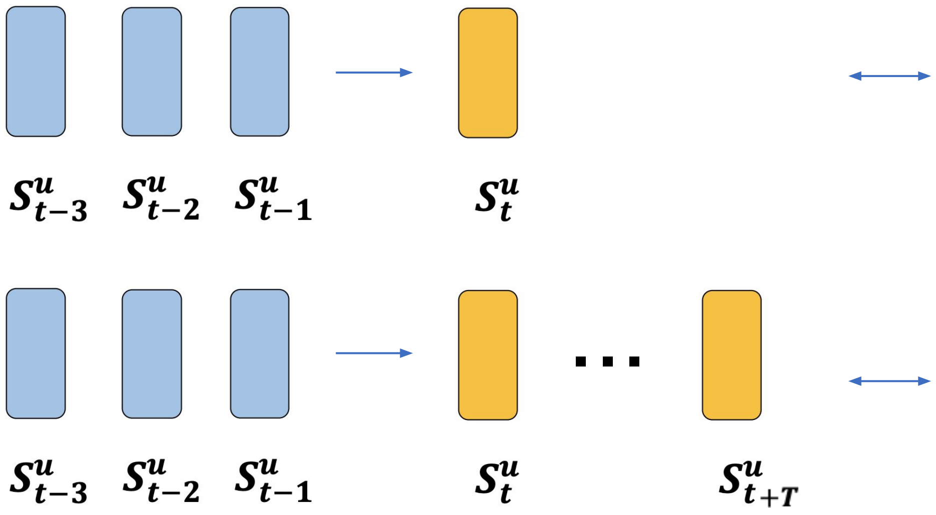


How can we build a better model?

Questions to be answered:

- How to model union-level influences?
- How to model point-level influences?
- How to incorporate skip behaviors in the model?

The solution can be straightforward: directly turn predicting the next item into predicting the next item set (including T items).



$$p(\mathcal{S}|\Theta) = \prod_u \prod_{t \in C^u} \sigma(\boldsymbol{y}^{(u,t)}_{\mathcal{S}^u_t}) \prod_{j \neq \mathcal{S}^u_t} (1 - \sigma(\boldsymbol{y}^{(u,t)}_j))$$

$$\ell = \sum_u \sum_{t \in C^u} \sum_{i \in \mathcal{D}^u_t} -\log(\sigma(\boldsymbol{y}^{(u,t)}_i)) + \sum_{j \neq i} -\log(1 - \sigma(\boldsymbol{y}^{(u,t)}_j))$$

$$\mathcal{D}^u_t = \{\mathcal{S}^u_t, \mathcal{S}^u_{t+1}, \ldots, \mathcal{S}^u_{t+T}\}$$
$$C^u = \{L+1, L+2, \ldots, |\mathcal{S}^u|\}$$

UVA ENGINEERING

Experiment Design

Datasets & Train/Test Split

Datasets:

MovieLens: user id, movie, rating

Gowalla: user id, check-in time, location

Foursquare (non-public): user id, check-in time, location

Tmall (non-public): user id, purchase time, item

Selected because of acceptable "sequential intensity" according to the authors

Train/test Split:

70% train, 10% validation (tune), 20% test

Dataset Characteristics

Table 1: Statistics of the datasets

Datasets	Sequential Intensity	#users	#items	avg. actions per user	Sparsity
MovieLens	0.3265	6.0k	3.4k	165.50	95.16%
Gowalla	0.0748	13.1k	14.0k	40.74	99.71%
Foursquare	0.0378	10.1k	23.4k	30.16	99.87%
Tmall	0.0104	23.8k	12.2k	13.93	99.89%

Define a rule $X_L \to Y$ to be $(S_{t-L}, ..., S_t) \to S_{t+1}$

X_L : L previous items, Y : next item

$$\text{Sequential Intensity } (SI) = \frac{\#\text{rules}}{\#\text{users}}.$$

$$\#\text{rules} = \sum_{L=1}^{5} \left| \left\{ X_L \mid \frac{sup(X_L Y)}{sup(X_L)} > 0.5 \text{ and } sup(X_L Y) \geq 5 \right\} \right|$$

where $sup(X_L Y)$(support) is the number of rules that have Y follows X_L and $sup(X_L)$ is the number of occurrences of X_L

A measure of the intensity of sequential patterns

Evaluation Metrics

Rhat: Predicted top N items the user want next

R: Ground truth (actual items the user want next)

Rel(N): 0 or 1, indicates whether the item predicted N exists in the ground truth

$$\text{Prec@}N = \frac{|R \cap \hat{R}_{1:N}|}{N},$$

$$\text{Recall@}N = \frac{|R \cap \hat{R}_{1:N}|}{|R|}.$$

$$\text{AP} = \frac{\sum_{N=1}^{|\hat{R}|} \text{Prec@}N \times \text{rel}(N)}{|\hat{R}|},$$

Results & Other Conclusions

Performance comparison on data sets

Caser performs better in most cases with different metrics.

Dataset	Metric	POP	BPR	FMC	FPMC	Fossil	GRU4Rec	Caser	Improv.
MovieLens	Prec@1	0.1280	0.1478	0.1748	0.2022	0.2306	**0.2515**	0.2502	-0.5%
	Prec@5	0.1113	0.1288	0.1505	0.1659	0.2000	0.2146	**0.2175**	1.4%
	Prec@10	0.1011	0.1193	0.1317	0.1460	0.1806	0.1916	**0.1991**	4.0%
	Recall@1	0.0050	0.0070	0.0104	0.0118	0.0144	**0.0153**	0.0148	-3.3%
	Recall@5	0.0213	0.0312	0.0432	0.0468	0.0602	0.0629	**0.0632**	0.5%
	Recall@10	0.0375	0.0560	0.0722	0.0777	0.1061	0.1093	**0.1121**	2.6%
	MAP	0.0687	0.0913	0.0949	0.1053	0.1354	0.1440	**0.1507**	4.7%
Gowalla	Prec@1	0.0517	0.1640	0.1532	0.1555	0.1736	0.1050	**0.1961**	13.0%
	Prec@5	0.0362	0.0983	0.0876	0.0936	0.1045	0.0721	**0.1129**	8.0%
	Prec@10	0.0281	0.0726	0.0657	0.0698	0.0782	0.0571	**0.0833**	6.5%
	Recall@1	0.0064	0.0250	0.0234	0.0256	0.0277	0.0155	**0.0310**	11.9%
	Recall@5	0.0257	0.0743	0.0648	0.0722	0.0793	0.0529	**0.0845**	6.6%
	Recall@10	0.0402	0.1077	0.0950	0.1059	0.1166	0.0826	**0.1223**	4.9%
	MAP	0.0229	0.0767	0.0711	0.0764	0.0848	0.0580	**0.0928**	9.4%
Foursquare	Prec@1	0.1090	0.1233	0.0875	0.1081	0.1191	0.1018	**0.1351**	13.4%
	Prec@5	0.0477	0.0543	0.0445	0.0555	0.0580	0.0475	**0.0619**	6.7%
	Prec@10	0.0304	0.0348	0.0309	0.0385	0.0399	0.0331	**0.0425**	6.5%
	Recall@1	0.0376	0.0445	0.0305	0.0440	0.0497	0.0369	**0.0565**	13.7%
	Recall@5	0.0800	0.0888	0.0689	0.0959	0.0948	0.0770	**0.1035**	7.9%
	Recall@10	0.0954	0.1061	0.0911	0.1200	0.1187	0.1011	**0.1291**	7.6%
	MAP	0.0636	0.0719	0.0571	0.0782	0.0823	0.0643	**0.0909**	10.4%
Tmall	Prec@1	0.0010	0.0111	0.0197	0.0210	0.0280	0.0139	**0.0312**	11.4%
	Prec@5	0.0009	0.0081	0.0114	0.0120	0.0149	0.0090	**0.0179**	20.1%
	Prec@10	0.0007	0.0063	0.0084	0.0090	0.0104	0.0070	**0.0132**	26.9%
	Recall@1	0.0004	0.0046	0.0079	0.0082	0.0117	0.0056	**0.0130**	11.1%
	Recall@5	0.0019	0.0169	0.0226	0.0245	0.0306	0.0180	**0.0366**	19.6%
	Recall@10	0.0026	0.0260	0.0333	0.0364	0.0425	0.0278	**0.0534**	25.6%
	MAP	0.0030	0.0145	0.0197	0.0212	0.0256	0.0164	**0.0310**	21.1%

Caser components

Table 3: MAP vs. Caser Components

	MovieLens	Gowalla
Caser-p	0.0935	0.0777
Caser-h	0.1304	0.0805
Caser-v	0.1403	0.0841
Caser-vh	0.1448	0.0856
Caser-ph	0.1372	0.0911
Caser-pv	0.1494	0.0921
Caser-pvh	0.1507	0.0928

Different components of Caser model (such as horizontal and vertical filters) all exhibit effectiveness.

Numerical Results

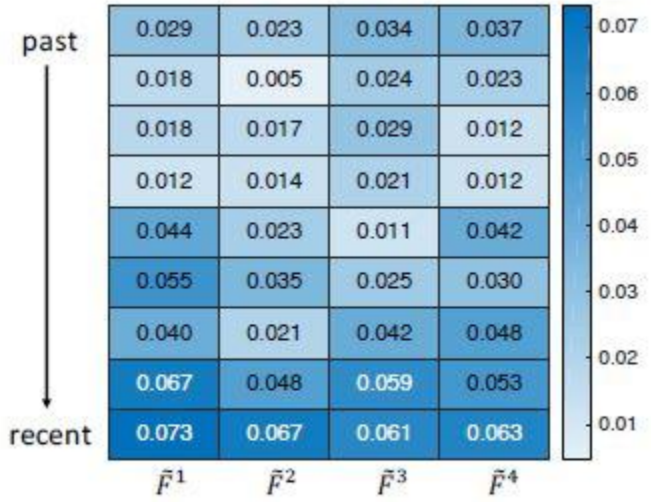


Figure 7: Visualization for four vertical convolutional filters of a trained model on MovieLens data when $L = 9$.

Recent items have higher weight.

Capture of union-level sequential patterns

(b)

Masking Items	New Rank of $\hat{R}_3$ after masking
S_1, S_2	2
S_3	32
S_4	117
S_5	77
S_3, S_4, S_5	513

Figure 8: Horizontal convolutional filters's effectiveness of capturing union-level sequential patterns on MovieLens data.

Thank you for listening!

Q&A