Sequential Data Augmentation for Generative Recommendation

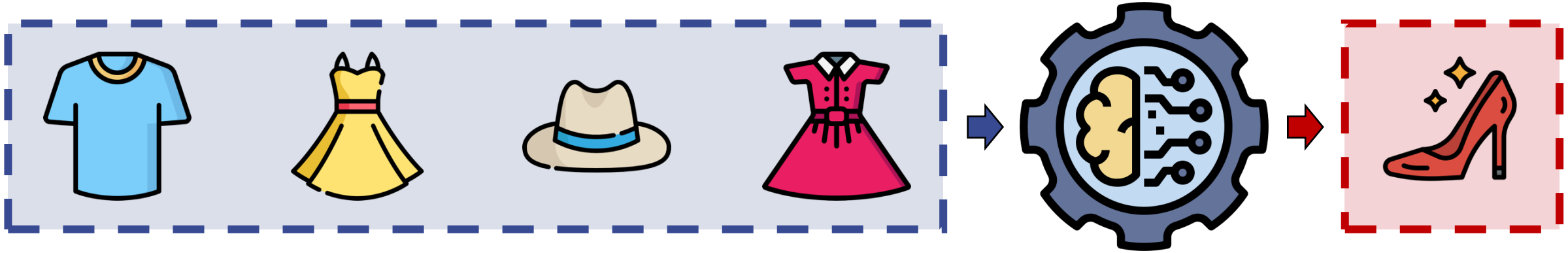
Geon Lee, Bhuvesh Kumar, Clark Mingxuan Ju, Tong Zhao,
Kijung Shin, Neil Shah, and Liam Collins

Overview

1. **Introduction**
2. Common Strategies
3. Analysis: Empirical Study
4. Analysis: Training Distribution
5. Proposed Framework
6. Experiments
7. Conclusions

Generative Recommendation

- **Generative recommendation (GR)** aims to predict next item from user behavior sequences.

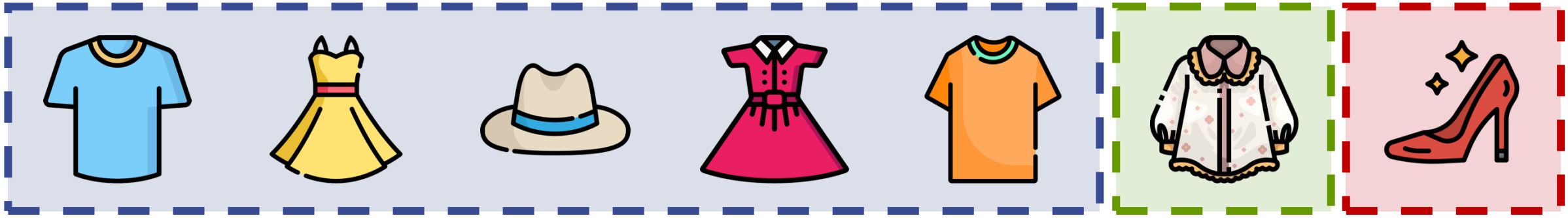


User Behavior Sequence (Input)

Predicted Item (Output)

Problem Setup

- **Given:** A user interaction sequence $(i_1, i_2, \ldots, i_n)$
- **Goal:** The next item i_{n+1}
- **Evaluation Protocol:** **Leave-last-out** evaluation



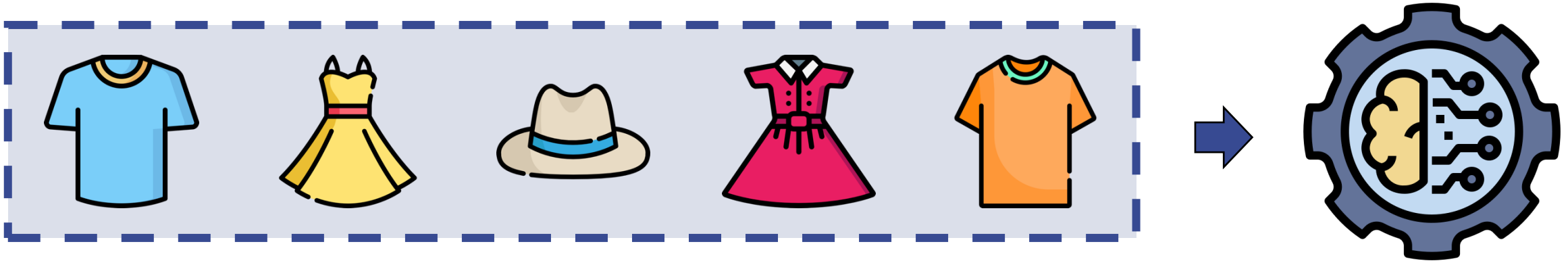
Training Data

Validation Item

Test Item

Data Augmentation

- How to effectively construct the training data (i.e., **data augmentation**)?
- **Note:** This process is often simplified, applied inconsistently, or treated as a minor implementation detail.



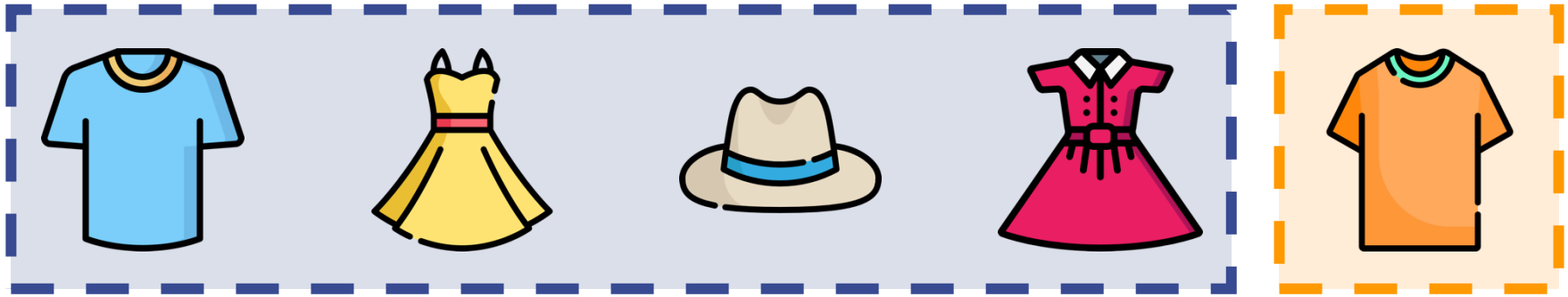
Training Data

Overview

1. Introduction
2. **Common Strategies**
3. Analysis: Empirical Study
4. Analysis: Training Distribution
5. Proposed Framework
6. Experiments
7. Conclusions

Common Strategies – Last-Target

- **Last-Target (LT):** Generate a single training sample per user using the last item in the training sequence as the prediction target.

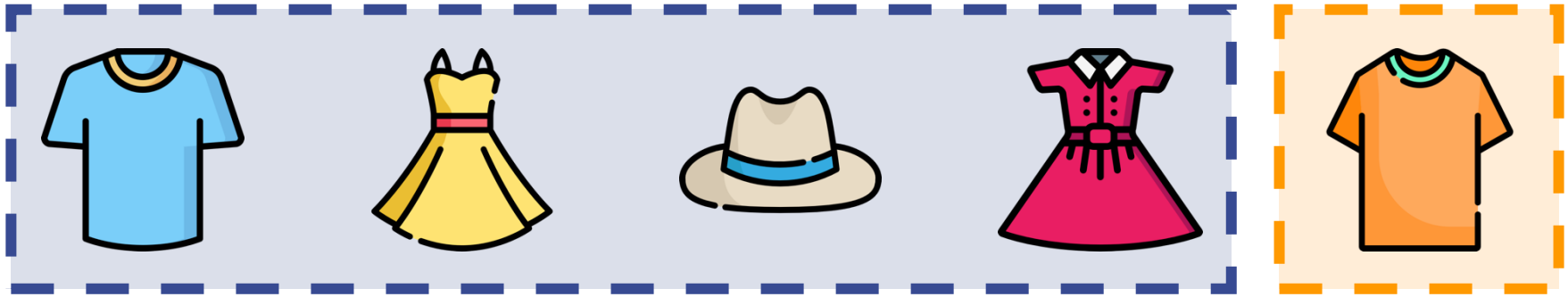


Training Input

Training Target

Common Strategies – Multi-Target

- **Multi-Target (MT):** Predict multiple targets within each user sequence.

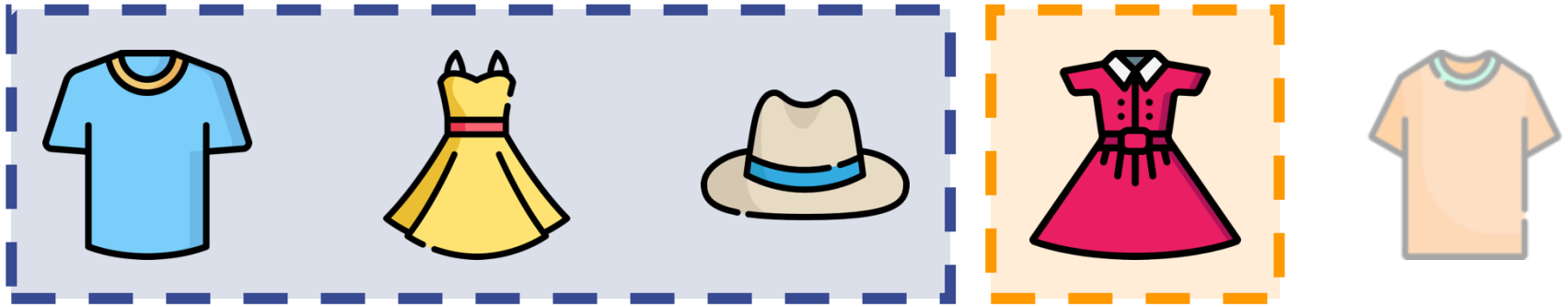


Training Input

Training Target

Common Strategies – Multi-Target

- **Multi-Target (MT):** Predict multiple targets within each user sequence.

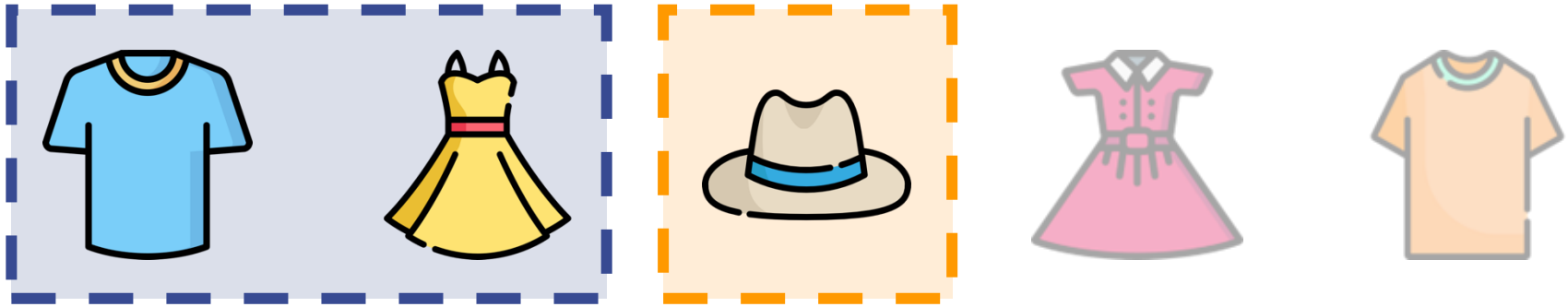


Training Input

Training Target

Common Strategies – Multi-Target

- **Multi-Target (MT):** Predict multiple targets within each user sequence.



Training Input

Training Target

Common Strategies – Multi-Target

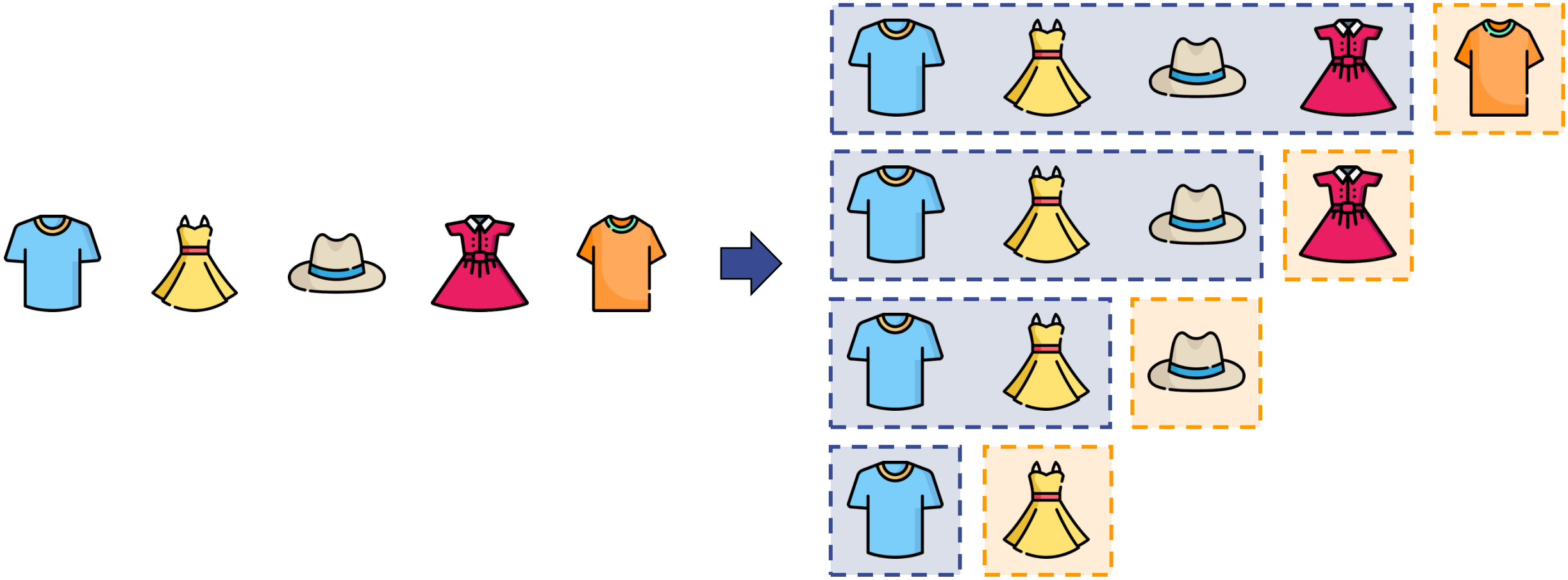
- **Multi-Target (MT):** Predict multiple targets within each user sequence.

Training Input

Training Target

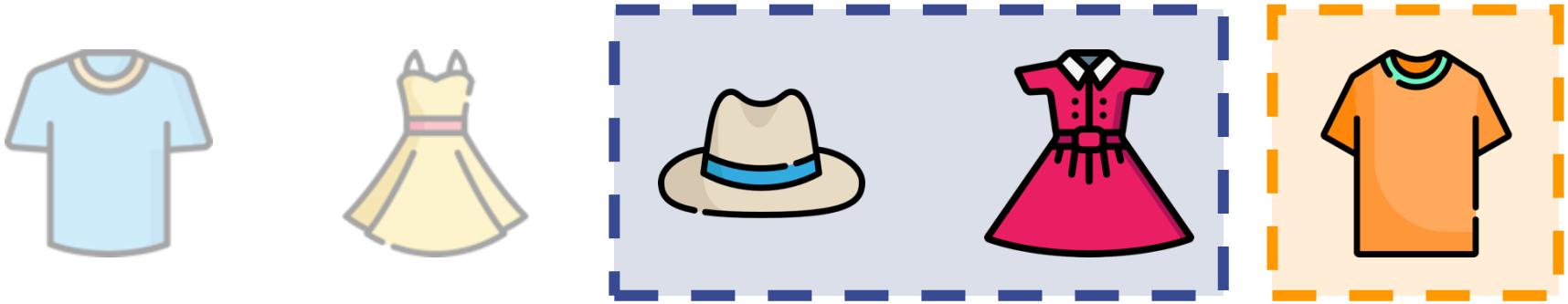
Common Strategies – Multi-Target

- **Multi-Target (MT):** Predict multiple targets within each user sequence.



Common Strategies – Slide Window

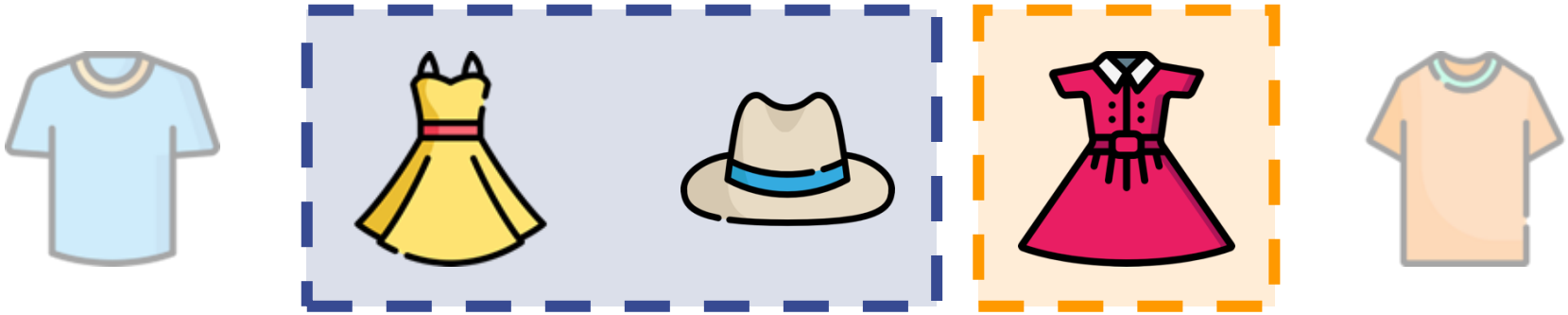
- **Slide-Window (SW):** Extract all possible size-$\{2, \ldots, |s|\}$ contiguous subsequences from each user sequence.



Window size w = 2 for example

Common Strategies – Slide Window

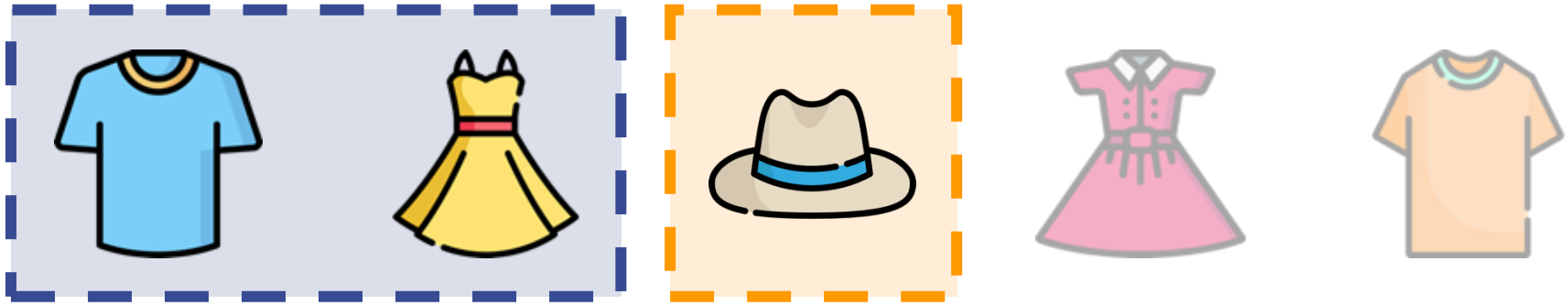
- **Slide-Window (SW):** Extract all possible size-$\{2, \ldots, |s|\}$ contiguous subsequences from each user sequence.



Window size w = 2 for example

Common Strategies – Slide Window

- **Slide-Window (SW):** Extract all possible size-$\{2, \ldots, |s|\}$ contiguous subsequences from each user sequence.



Training Input Training Target

Window size w = 2 for example

Common Strategies – Slide Window

- **Slide-Window (SW):** Extract all possible size-$\{2, \ldots, |s|\}$ contiguous subsequences from each user sequence.

Overview

1. Introduction
2. Common Strategies
3. **Analysis: Empirical Study**
4. Analysis: Training Distribution
5. Proposed Framework
6. Experiments
7. Conclusions

Empirical Analysis of Target Distribution

- How do data augmentation strategies **reshape training distributions**?

Augmented Training Data $\mathcal{D}_{\text{train}}$

Target Distribution

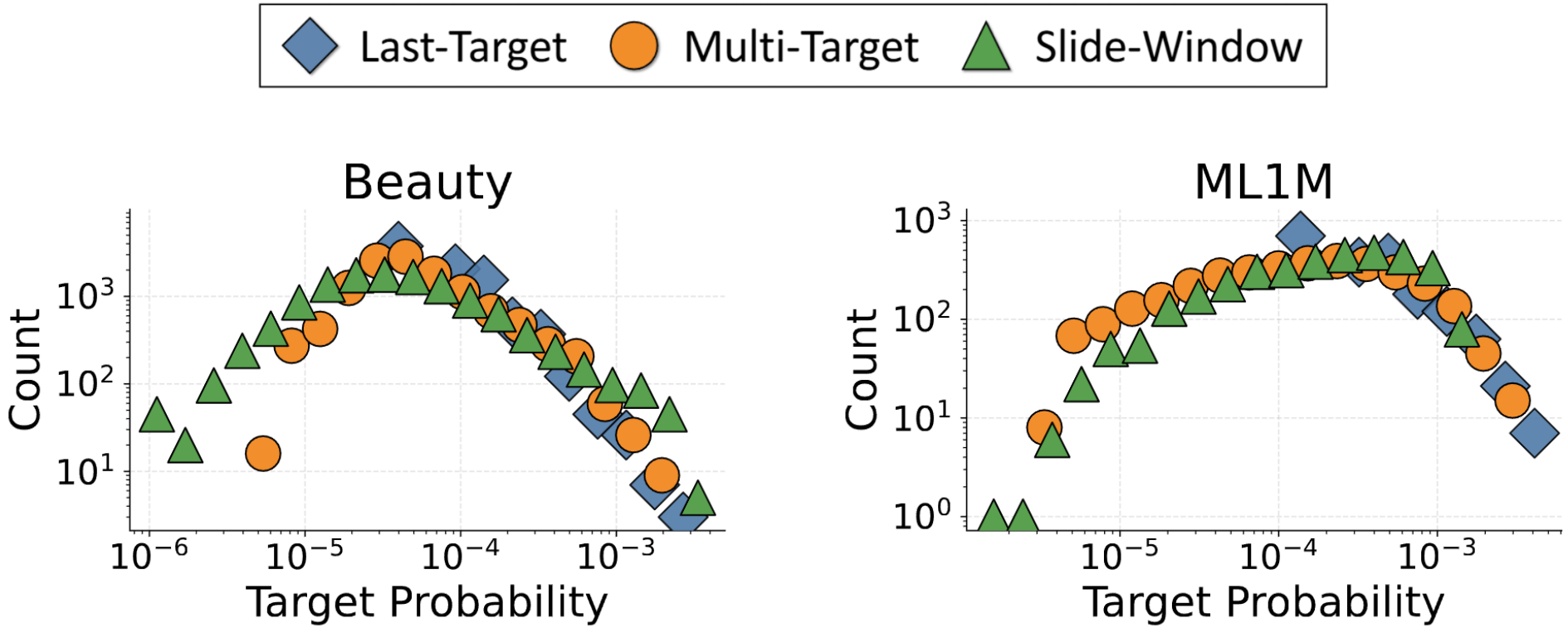
$$p_{\text{train}}(\text{  }) = \frac{2}{3}$$

$$p_{\text{train}}(\text{  }) = \frac{1}{3}$$

How frequently does the item appear as the target?

Empirical Analysis of Target Distribution

- Different strategies yield distinct **target distributions**.



Empirical Analysis of Target Distribution

- Different strategies yield distinct **target distributions**.
- Ranking (w.r.t. target probability) agreement of items.

	LT	MT	SW
LT	1.000	0.728	0.658
MT		1.000	0.791
SW			1.000

Beauty

	LT	MT	SW
LT	1.000	0.746	0.732
MT		1.000	0.936
SW			1.000

ML-1M

Empirical Analysis of Target Distribution

- Different strategies yield distinct **target distributions**.
- Average rank (w.r.t. target probability) differences between pairs of items.

	LT	MT	SW
LT	0	2263	2762
MT		0	1754
SW			0

Beauty

	LT	MT	SW
LT	0	557	587
MT		0	157
SW			0

ML-1M

Empirical Analysis of Input-Target Distribution

- How do data augmentation strategies **reshape training distributions**?

Augmented Training Data $\mathcal{D}_{\text{train}}$

Conditional Input|Target Distribution

$$|p_{\text{train}}(\cdot \mid \text{  })| = 2$$

$$|p_{\text{train}}(\cdot \mid \text{  })| = 1$$

How diverse are the inputs associated with each target?

Empirical Analysis of Input-Target Distribution

- Different strategies yield distinct **input-target distributions**.

Empirical Importance of Data Augmentation

- As a result, generative recommendation model performance (NDCG@10) is **highly sensitive to the data augmentation strategy**.

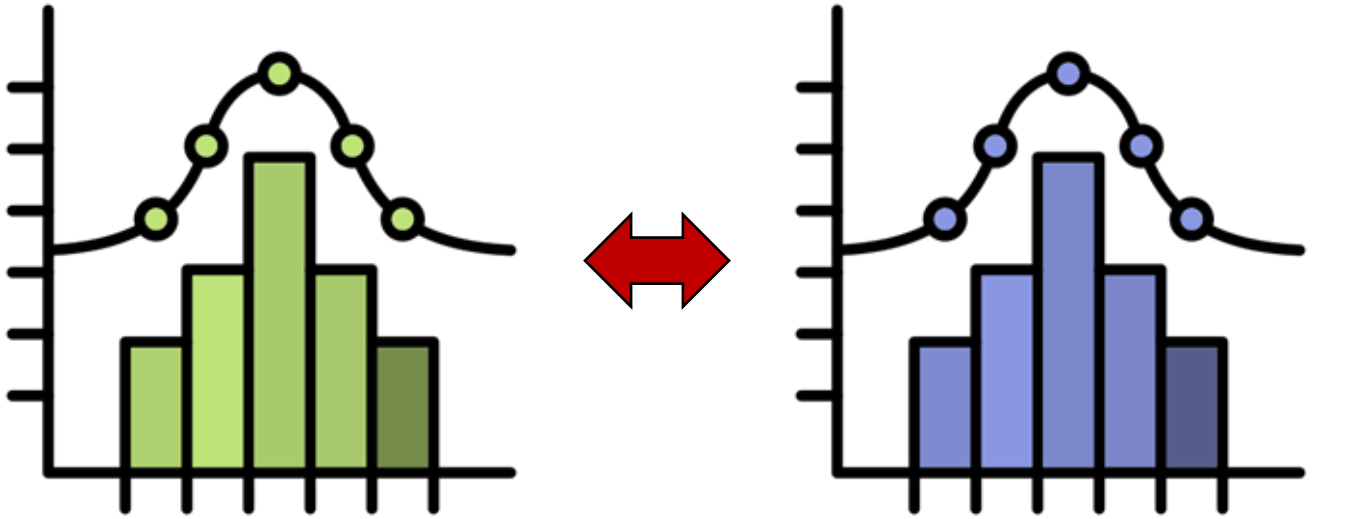
Model	Strategy	**Beauty**	**Toys**	**Sports**	**ML1M**	**ML20M**
SASRec	Last-Target	0.0124	0.0121	0.0037	0.0136	0.0628
	Multi-Target	**0.0372**	**0.0378**	**0.0162**	**0.1194**	**0.0995**
	Slide-Window	0.0323	0.0354	0.0149	0.1022	0.0526
	Δ (Best/Worst)	200.0%	212.4%	337.8%	777.9%	89.2%
TIGER	Last-Target	0.0213	0.0212	0.0150	0.0147	0.0559
	Multi-Target	0.0319	**0.0303**	**0.0194**	**0.1299**	**0.1147**
	Slide-Window	**0.0321**	0.0273	0.0171	0.1105	0.0321
	Δ (Best/Worst)	50.7%	42.9%	29.3%	783.7%	257.3%

Overview

1. Introduction
2. Common Strategies
3. Analysis: Empirical Study
4. **Analysis: Training Distribution**
5. Proposed Framework
6. Experiments
7. Conclusions

Analysis of Target Distribution

- For effective **generalization**, the **training** **target distribution** should closely align with that at **test** time.



Target Distribution (**Training**; p_{train})

Target Distribution (**Test**; p_{test})

$$\mathrm{KL}(p_{\text{train}} \parallel p_{\text{test}}) = \sum_{y \in I} p_{\text{train}}(y) \log \frac{p_{\text{train}}(y)}{p_{\text{test}}(y)}$$

Analysis of Target Distribution

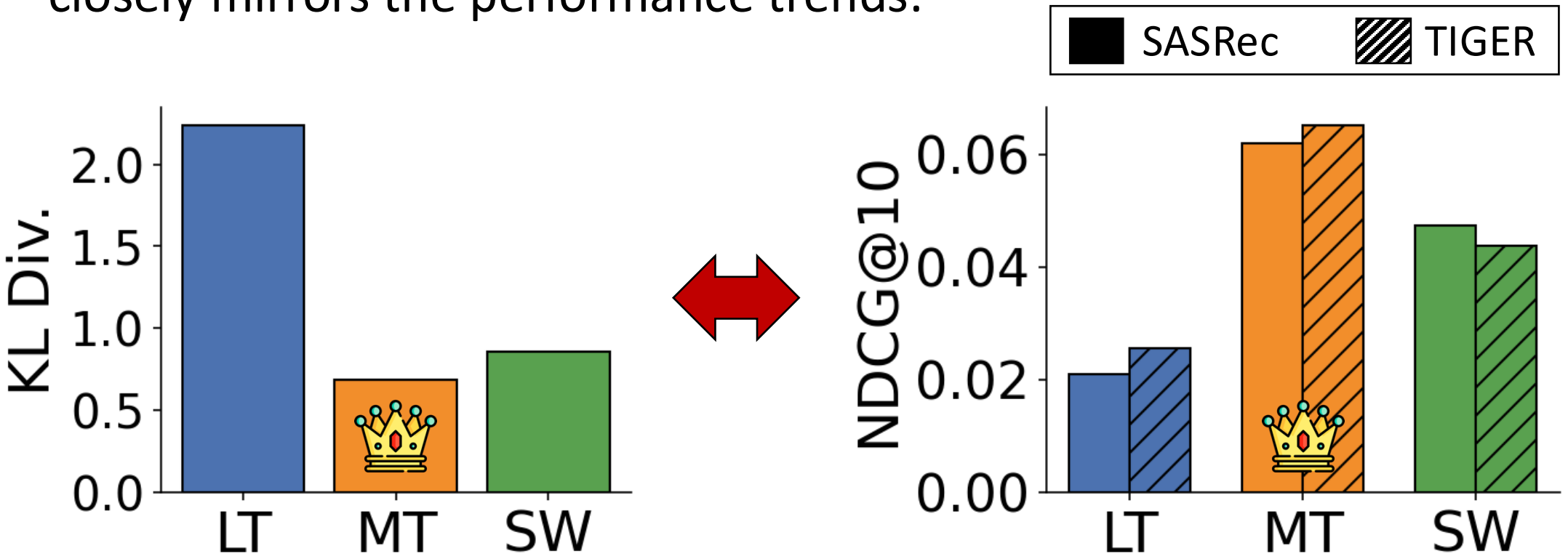
- **Multi-Target** yields the ***lowest*** KL divergence (**strong alignment**);
  Last-Target has the ***highest*** KL divergence (**misalignment**).

	Beauty	Toys	Sports	ML1M	ML20M
Last-Target (LT)	2.768	3.147	2.737	2.198	0.343
Multi-Target (MT)	**0.898**	**1.062**	**0.819**	**0.495**	**0.168**
Slide-Window (SW)	1.158	1.349	0.910	0.563	0.312

KL Divergence of Target Distributions

Analysis of Target Distribution

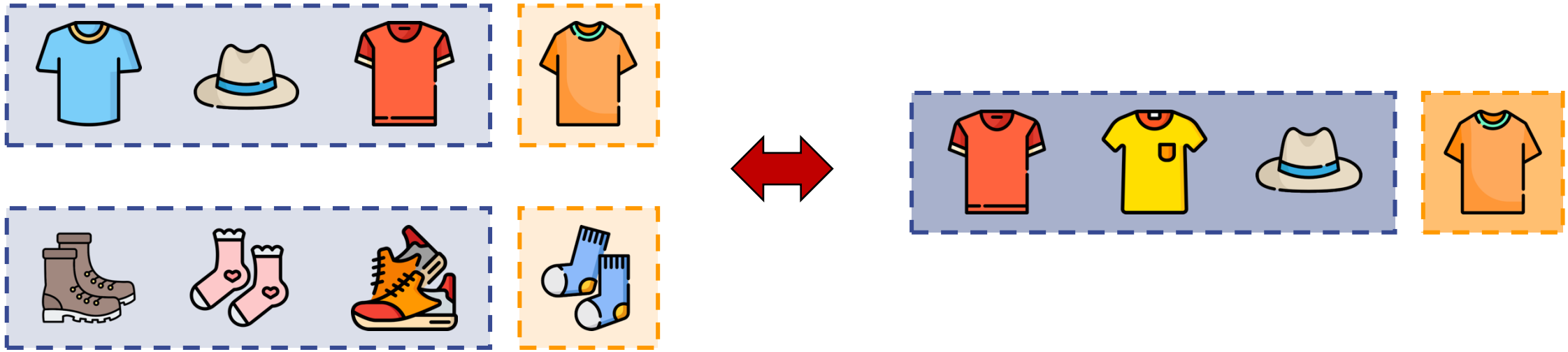
- The **alignment** between training and test **target distributions** closely mirrors the performance trends.



* Average values across five datasets

Analysis of Input-Target Distribution

- For effective **generalization**, the **training** **input-target distribution** should closely align with that at **test** time.

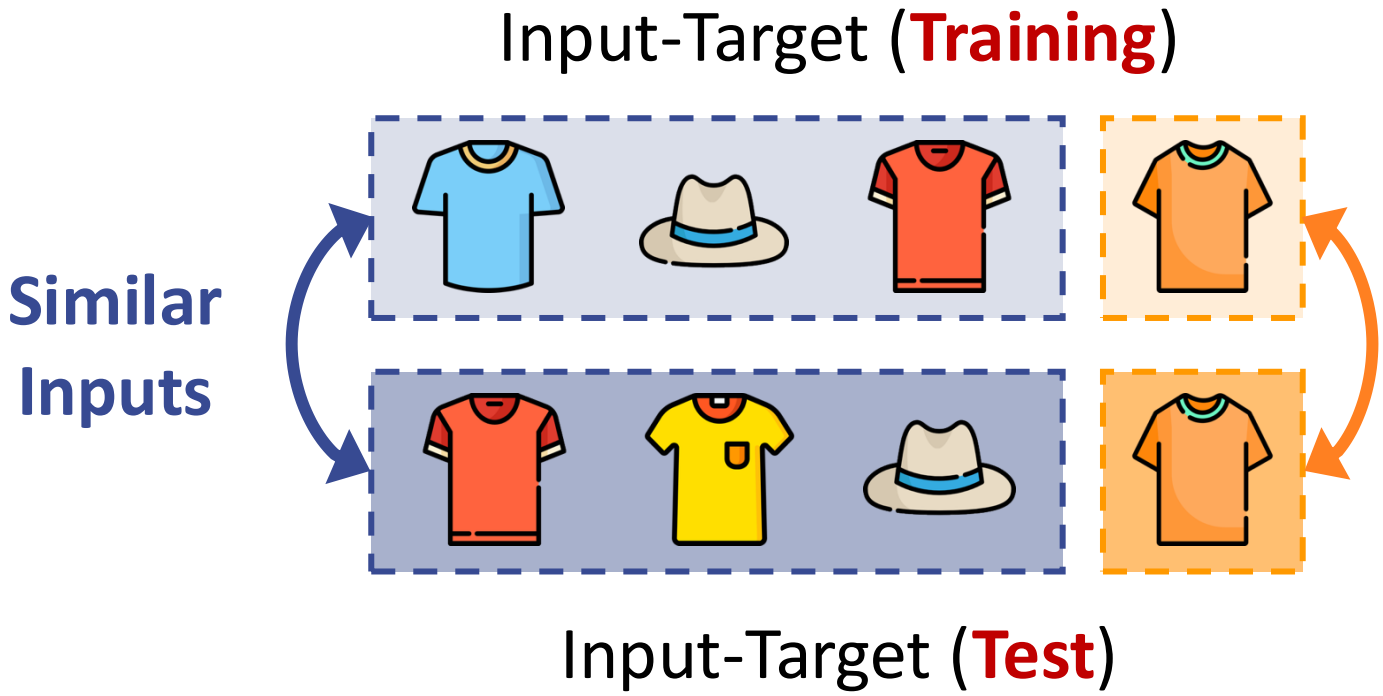


Input-Target
(**Training**)

Input-Target
(**Test**)

Analysis of Input-Target Distribution

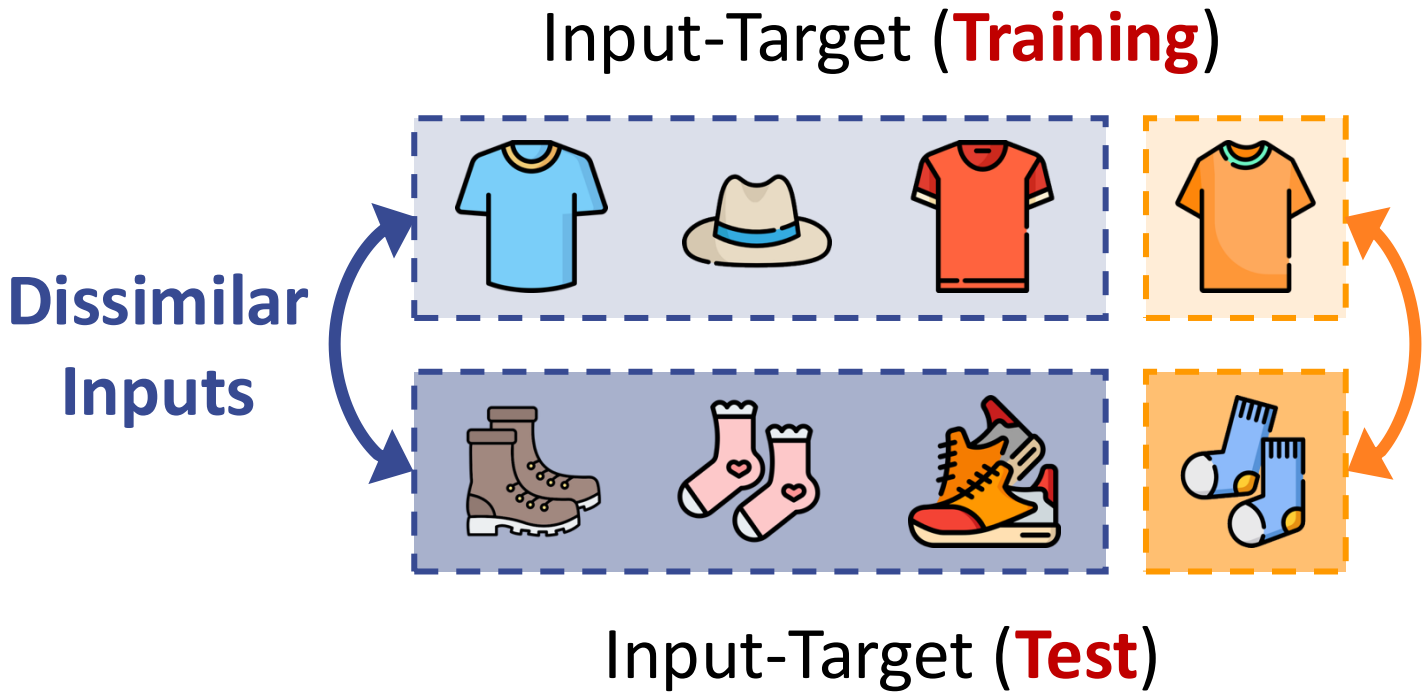
- **Alignment:** Inputs associated with the ***same*** target should exhibit ***similar*** structural patterns across training and test.



$$\mathbb{E}_y \mathbb{E}_{\substack{(x,y)\sim\mathcal{D}_{\text{test}} \\ (x',y)\sim\mathcal{D}_{\text{train}}}} \text{sim}(x, x')$$

Analysis of Input-Target Distribution

- **Discrimination:** Inputs associated with ***different*** targets should exhibit ***dissimilar*** structural patterns across training and test.



$$\mathbb{E}_{y \neq y'} \mathbb{E}_{\substack{(x,y)\sim\mathcal{D}_{\text{test}} \\ (x',y')\sim\mathcal{D}_{\text{train}}}} \text{sim}(x, x')$$

Analysis of Input-Target Distribution

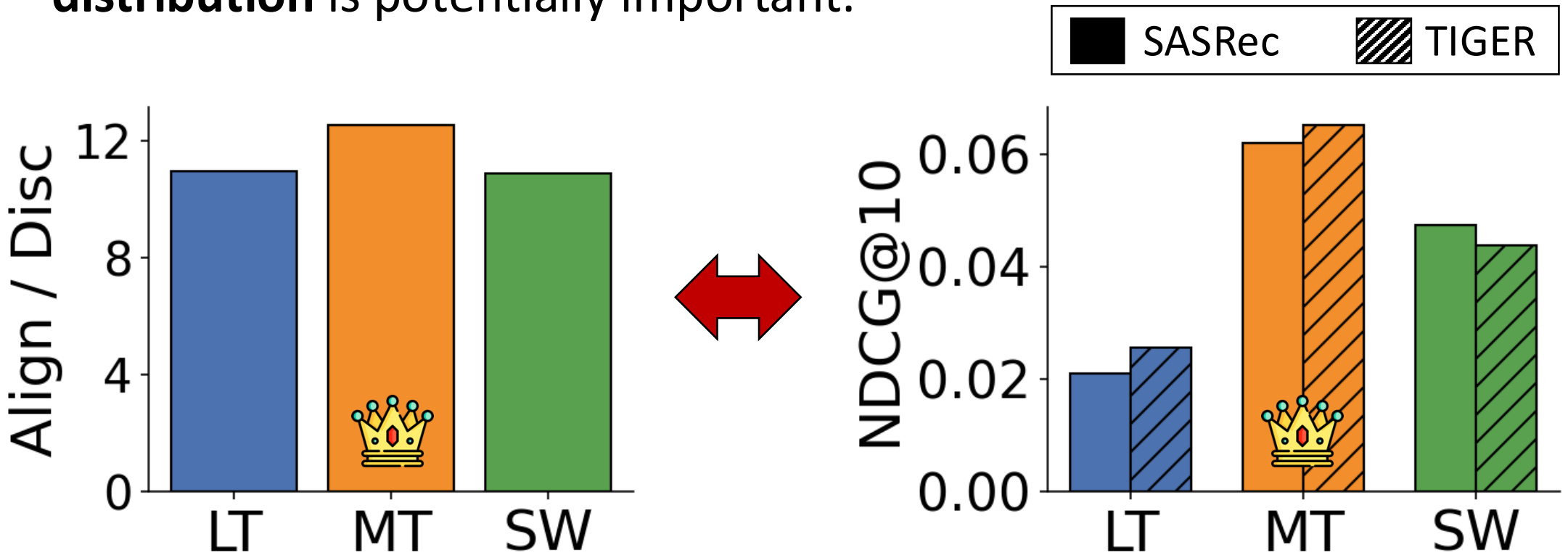
- **Multi-Target** yields the ***highest*** alignment-to-discrimination ratio than Last-Target and Slide-Window across datasets.

	Beauty	Toys	Sports	ML1M	ML20M
Last-Target (LT)	14.54	19.61	8.52	3.91	8.21
Multi-Target (MT)	**16.42**	**20.11**	**9.09**	**4.32**	**12.77**
Slide-Window (SW)	14.86	19.88	8.78	3.57	7.37

Alignment-to-Discrimination Ratio of Input-Target Distributions

Analysis of Input-Target Distribution

- Good trade-off between **alignment** and **discrimination** in **input-target distribution** is potentially important.



* Average values across five datasets

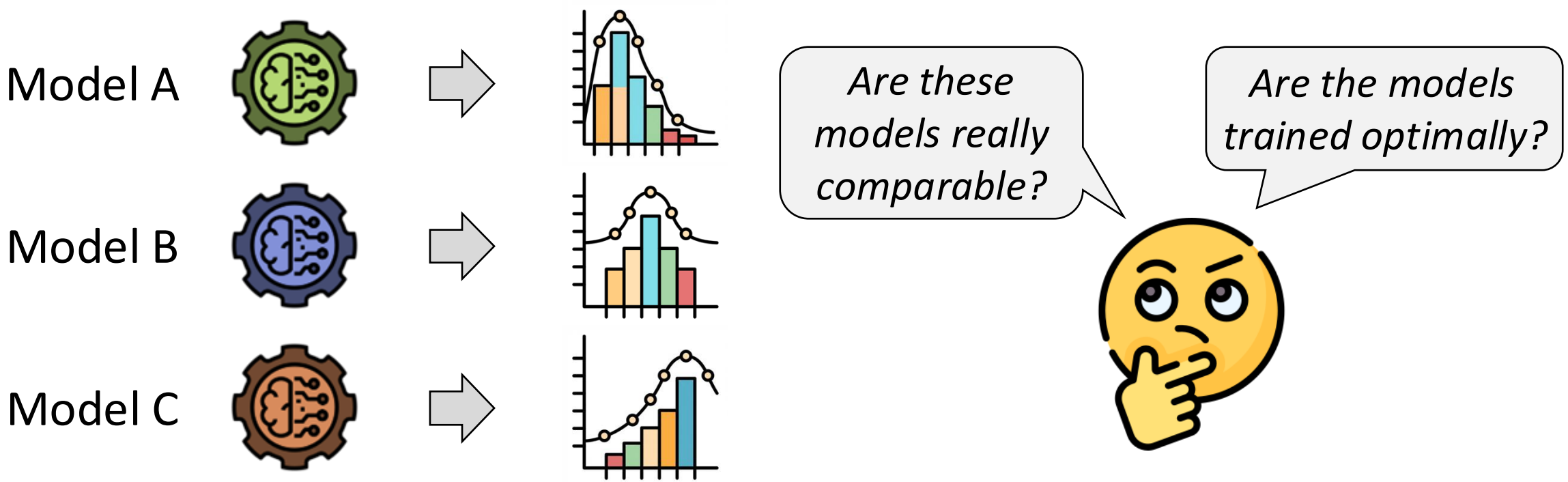
Overview

1. Introduction
2. Common Strategies
3. Analysis: Empirical Study
4. Analysis: Training Distribution
5. **Proposed Framework**
6. Experiments
7. Conclusions

GENPAS: Generalized Data Augmentation

- Despite its importance, generative recommendation models are often trained under **inconsistent or restrictive training distributions**.

→ It is difficult to accurately assess and compare model effectiveness.



GENPAS: Generalized Data Augmentation

- **GENPAS** is a ***generalized*** and ***principled*** framework for data augmentation.
- It unifies existing strategies into a **three-step sampling** process.

(1) Sequence Sampling

⇩

(2) Target Sampling

⇩

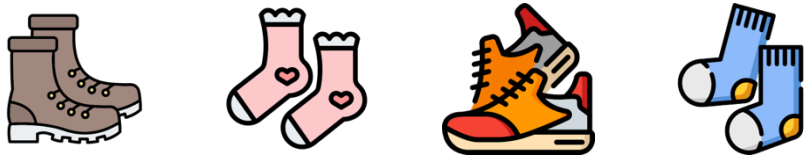
(3) Input Sampling

GENPAS: Generalized Data Augmentation

(1) **Sequence Sampling**: Samples a user sequence $s \in \mathcal{S}$ with probability:

$$p_\alpha(s) = \frac{(|s| - 1)^\alpha}{\sum_{s' \in \mathcal{S}} (|s'| - 1)^\alpha}$$

Sample a sequence



- $\alpha < 0$ → Favors *shorter* sequences
- $\alpha = 0$ → Samples all sequences *uniformly*
- $\alpha > 0$ → Favors *longer* sequences

GENPAS: Generalized Data Augmentation

(2) **Target Sampling**: Samples a target position $k \in \{2, \dots |s|\}$ with probability:

$$p_\beta(k \mid s) = \frac{(k-1)^\beta}{\sum_{k'=2}^{|s|}(k'-1)^\beta}$$

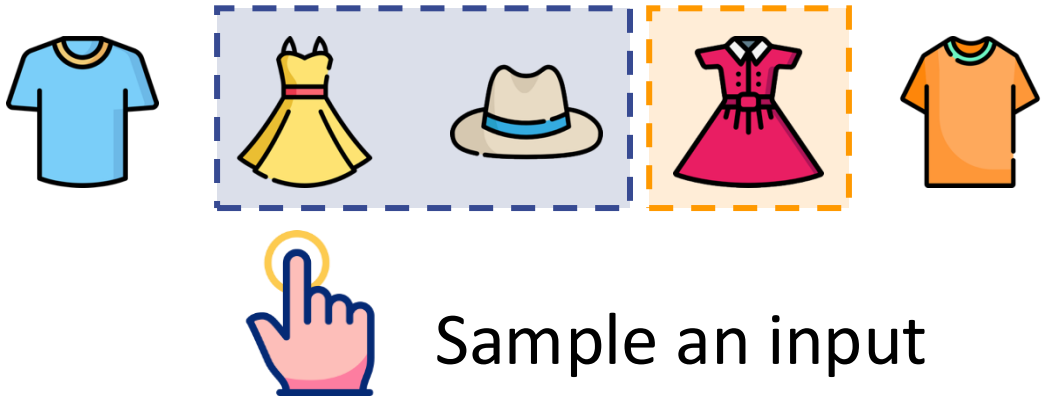
Sample a target 

- $\beta < 0$ → Favors *earlier* targets
- $\beta = 0$ → Samples all targets *uniformly*
- $\beta > 0$ → Favors *recent* targets

GENPAS: Generalized Data Augmentation

(3) **Input Sampling**: Samples a start position $j \in \{1, \dots k-1\}$ with probability:

$$p_\gamma(j \mid k, s) = \frac{j^\gamma}{\sum_{j'=1}^{k-1} (j')^\gamma}$$



Sample an input

- $\gamma < 0$ → Favors *longer* inputs
- $\gamma = 0$ → Samples all inputs *uniformly*
- $\gamma > 0$ → Favors *shorter* inputs

GENPAS: Generalized Data Augmentation

- The joint distribution over an **input-target pair** $(\tilde{x}, \tilde{y})$ factorizes as:

$$p(\tilde{x}, \tilde{y}) = p(s, k, j) = p_\alpha(s) \cdot p_\beta(k \mid s) \cdot p_\gamma(j \mid k, s)$$

(1) Sequence Sampling

⇩

(2) Target Sampling

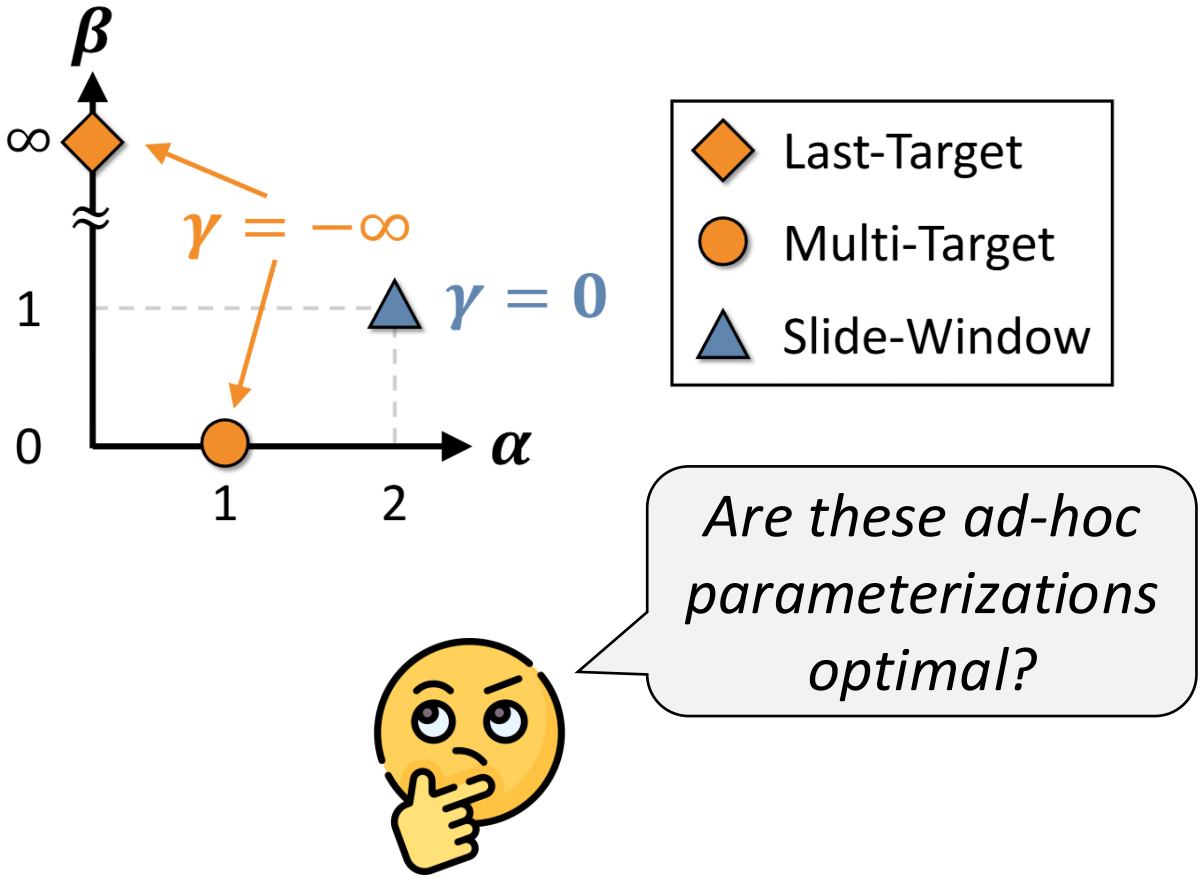
⇩

(3) Input Sampling

GENPAS: Generalized Data Augmentation

- Common augmentation strategies are *special cases* of **GENPAS**.

Strategy	α	β	γ
Last-Target	0.0	∞	$-\infty$
Multi-Target	1.0	0.0	$-\infty$
Slide-Window	2.0	1.0	0.0



Overview

1. Introduction
2. Common Strategies
3. Analysis: Empirical Study
4. Analysis: Training Distribution
5. Proposed Framework
6. **Experiments**
7. Conclusions

Experimental Settings

- **Datasets:** Five public datasets (**Amazon** & **MovieLens**) and a large-scale internal dataset collected from **Snapchat**.
- **Models:** Two representative GR models **SASRec** & **TIGER**

Dataset	Beauty	Toys	Sports	ML1M	ML20M	Internal
# Users	22,363	19,412	35,598	6,040	138,493	69.38M
# Items	12,101	11,924	18,357	3,416	26,744	42,805
# Interactions	198,502	167,597	296,337	999,611	20,000,263	1.527B
# Avg. Length	8.88	8.63	8.32	165.50	144.41	22.01
Sparsity	99.93%	99.93%	99.95%	95.16%	99.46%	99.94%

Overall Performance

- **GENPAS** *consistently* and *significantly* outperforms common data augmentation strategies (i.e., Last-Target, Multi-Target, & Slide-Window).

		Beauty		Toys		Sports		ML1M		ML20M	
		NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10
SASRec	Last-Target	0.0124 ±0.0005	0.0237 ±0.0013	0.0121 ±0.0003	0.0237 ±0.0008	0.0037 ±0.0003	0.0073 ±0.0005	0.0136 ±0.0009	0.0306 ±0.0016	0.0628 ±0.0010	0.1142 ±0.0011
	Multi-Target	0.0372 ±0.0006	0.0623 ±0.0008	0.0378 ±0.0014	0.0636 ±0.0023	0.0162 ±0.0006	0.0282 ±0.0008	0.1194 ±0.0046	0.2236 ±0.0066	0.0995 ±0.0015	0.1824 ±0.0028
	Slide-Window	0.0323 ±0.0004	0.0510 ±0.0009	0.0354 ±0.0003	0.0571 ±0.0006	0.0149 ±0.0001	0.0256 ±0.0007	0.1022 ±0.0062	0.1960 ±0.0082	0.0526 ±0.0024	0.1076 ±0.0050
	GENPAS	**0.0426** ±0.0003	**0.0689** ±0.0007	**0.0481** ±0.0007	**0.0771** ±0.0016	**0.0219** ±0.0003	**0.0365** ±0.0003	**0.1230** ±0.0029	**0.2288** ±0.0053	**0.1115** ±0.0015	**0.1938** ±0.0024
	Improv.	**14.52%**	**10.59%**	**27.25%**	**21.23%**	**35.19%**	**29.43%**	**3.02%**	**2.33%**	**12.06%**	**6.25%**
TIGER	Last-Target	0.0213 ±0.0020	0.0431 ±0.0035	0.0212 ±0.0002	0.0413 ±0.0011	0.0150 ±0.0003	0.0281 ±0.0009	0.0147 ±0.0012	0.0340 ±0.0028	0.0559 ±0.0019	0.1063 ±0.0028
	Multi-Target	0.0319 ±0.0003	0.0608 ±0.0007	0.0303 ±0.0011	0.0575 ±0.0019	0.0194 ±0.0001	0.0359 ±0.0002	0.1273 ±0.0024	0.2272 ±0.0050	0.1147 ±0.0059	0.1900 ±0.0091
	Slide-Window	0.0321 ±0.0014	0.0580 ±0.0009	0.0273 ±0.0007	0.0504 ±0.0008	0.0171 ±0.0004	0.0319 ±0.0010	0.1105 ±0.0038	0.1966 ±0.0049	0.0321 ±0.0077	0.0646 ±0.0147
	GENPAS	**0.0443** ±0.0010	**0.0766** ±0.0010	**0.0482** ±0.0011	**0.0822** ±0.0034	**0.0254** ±0.0009	**0.0453**±0.0013	**0.1390** ±0.0022	**0.2425**±0.0008	**0.1233** ±0.0003	**0.2009** ±0.0001
	Improv.	**38.01%**	**25.99%**	**59.08%**	**42.96%**	**30.93%**	**26.18%**	**9.19%**	**6.73%**	**7.50%**	**5.74%**

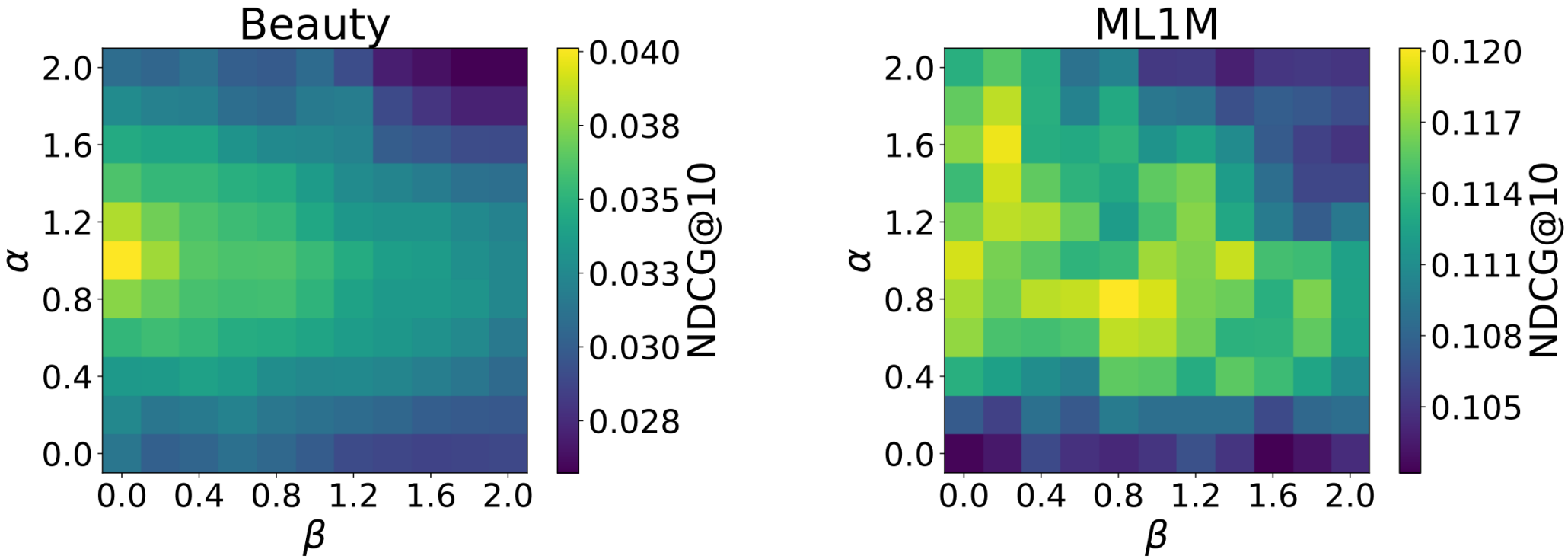
Overall Performance (vs Input-Level Augmentation)

- **GenPAS** outperforms input-level augmentation methods (e.g., insertion).
- **Note 1:** Instead of *perturbing* the input sequence, sampling *contiguous* inputs of varying lengths (by controlling γ) is more effective.
- **Note 2:** Prior work often evaluate only at $(\alpha, \beta) = (0, \infty)$.

	Beauty			Toys			Sports			ML1M			ML20M		
$(\alpha, \beta) \rightarrow$	$(0, \infty)$	$(1, 0)$	$(2, 1)$	$(0, \infty)$	$(1, 0)$	$(2, 1)$	$(0, \infty)$	$(1, 0)$	$(2, 1)$	$(0, \infty)$	$(1, 0)$	$(2, 1)$	$(0, \infty)$	$(1, 0)$	$(2, 1)$
Insert	0.0192	0.0399	0.0310	0.0217	0.0446	0.0322	0.0080	0.0203	0.0136	0.0182	0.1179	0.0984	0.0637	0.0973	0.0679
Delete	0.0147	0.0324	0.0233	0.0152	0.0349	0.0227	0.0052	0.0156	0.0092	0.0153	0.0711	0.0713	0.0531	0.0619	0.0462
Replace	0.0148	0.0308	0.0243	0.0149	0.0284	0.0183	0.0051	0.0130	0.0066	0.0176	0.1122	0.0971	0.0605	0.0893	0.0652
Reorder	0.0129	0.0353	0.0267	0.0128	0.0353	0.0225	0.0041	0.0154	0.0092	0.0147	0.1202	0.1074	0.0624	0.0969	0.0701
Sample	0.0159	0.0376	0.0299	0.0170	0.0383	0.0270	0.0060	0.0171	0.0110	0.0194	0.1143	0.0989	0.0601	0.0911	0.0649
$\gamma = -\infty$	0.0124	0.0372	0.0274	0.0121	0.0378	0.0226	0.0037	0.0162	0.0091	0.0136	0.1194	0.1100	0.0628	**0.0995**	0.0564
$\gamma = -1$	0.0221	0.0403	0.0323	0.0267	0.0455	0.0334	0.0099	0.0195	0.0133	0.0323	0.1107	0.0993	0.0693	0.0973	0.0667
$\gamma = 0$	0.0236	**0.0426**	0.0323	0.0287	**0.0481**	0.0354	0.0109	**0.0219**	0.0149	0.0382	**0.1230**	0.1022	0.0749	0.0937	0.0526
$\gamma = 1$	0.0240	0.0410	0.0323	0.0287	0.0473	0.0354	0.0112	0.0216	0.0144	0.0420	0.1041	0.0977	0.0759	0.0869	0.0625

Effectiveness of Components

- Varying α (sequence sampling) and β (target sampling) significantly affects model performance.

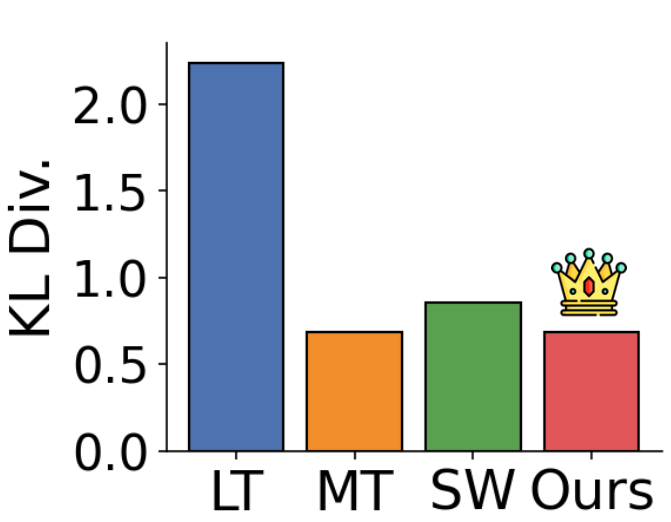


On Training Data Properties

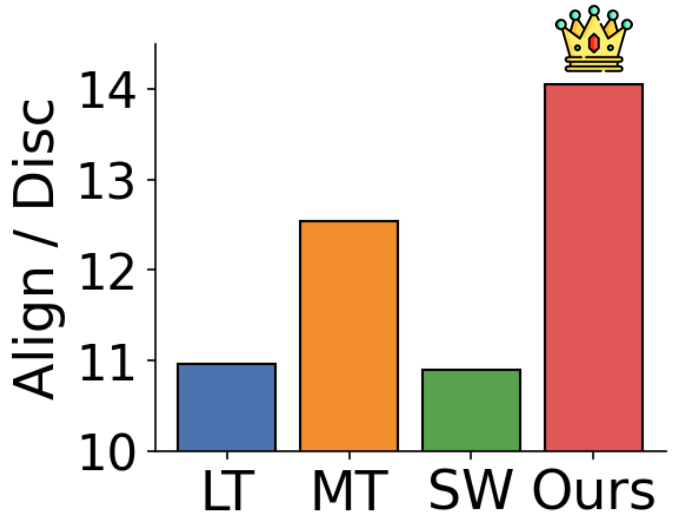
- Selected configurations (α, β, γ) for each dataset yield improved **target distributions** and **input-target distributions**.

	SASRec	TIGER
Beauty	$(1,0,0)$	$(1,0,1)$
Toys	$(1,0,0)$	$(1,0,1)$
Sports	$(1,0,0)$	$(1,1,1)$
ML1M	$(1,0,0)$	$(1,0,1)$
ML20M	$(0,2,-\infty)$	$(0,2,-\infty)$

Selected (α, β, γ) Configurations



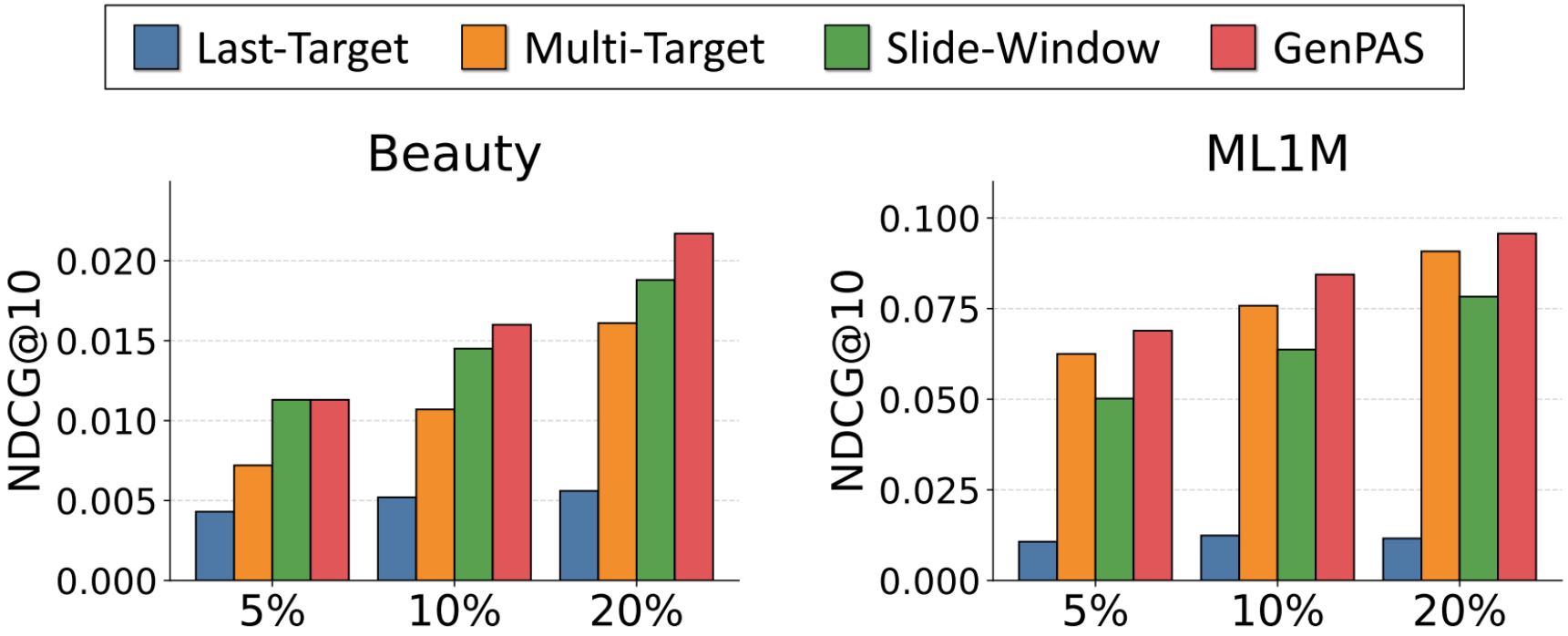
Target Distributions



Input-Target Distributions

Data Efficiency

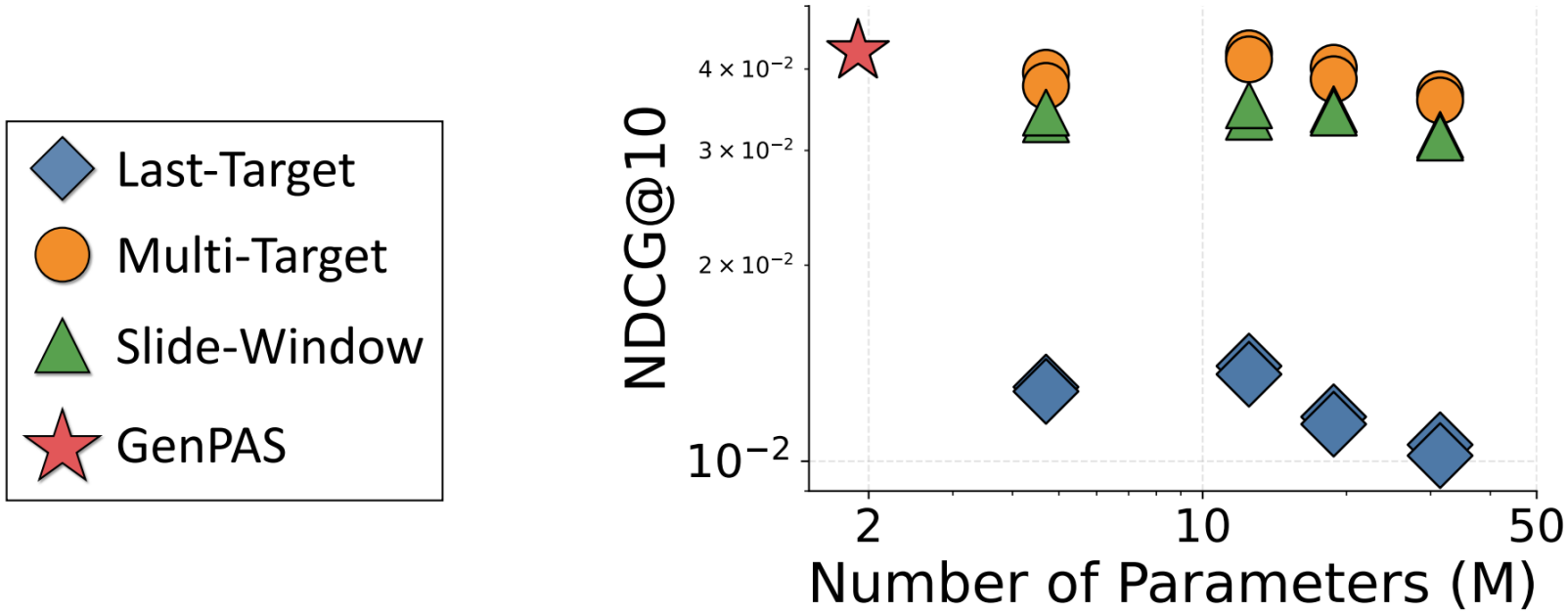
- **GENPAS** achieves strong performance even with reduced training data.
- **Note:** In ML1M, **GENPAS** with 1% training data > MT with full data



Parameter Efficiency

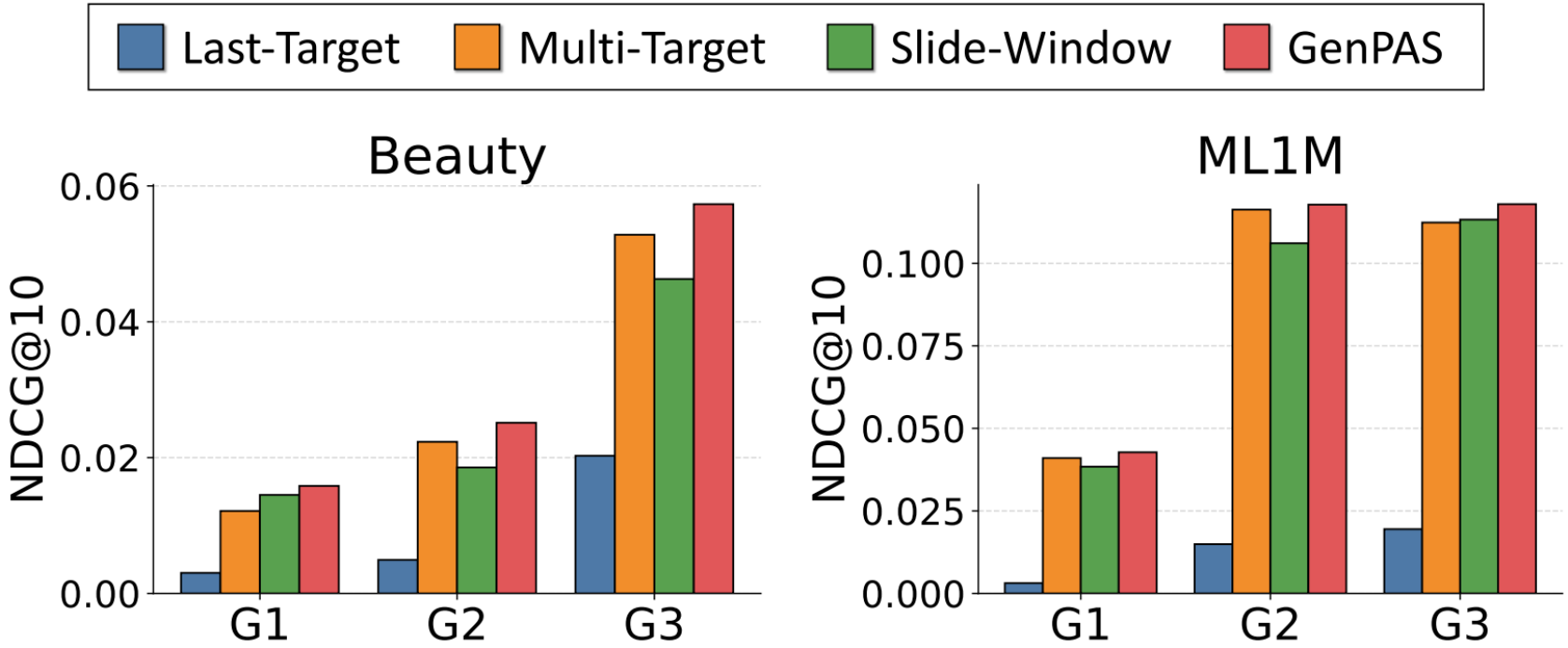
- SASRec trained on **GENPAS**-augmented data outperforms larger-parameter variants trained with other strategies.

→ Augmenting data can be more efficient than scaling model sizes.



Long Tail Performance

- **GENPAS** outperforms the other augmentation strategies across all item groups, from the *least popular* (G1) to the *most popular* (G3).



Large-Scale Data

- **GENPAS** outperforms its baseline (LT) in the industry-scale dataset.

	Transductive				Inductive (New Users)			
	Baseline (LT)		GENPAS		Baseline (LT)		GENPAS	
	N@10	R@10	N@10	R@10	N@10	R@10	N@10	R@10
	0.1904	0.3144	**0.2059**	**0.3512**	0.1961	0.3144	**0.2104**	**0.3357**
Improv.	–	–	**+8.14%**	**+11.7%**	–	–	**+7.29%**	**+6.77%**

Overview

1. Introduction
2. Common Strategies
3. Analysis: Empirical Study
4. Analysis: Training Distribution
5. Proposed Framework
6. Experiments
7. **Conclusions**

Conclusions

- We propose **GENPAS**, a principled and generalized data augmentation method for generative recommendation.

✓ **In-Depth Analysis:** We conduct a thorough analysis of widely used data augmentation strategies for generative recommendation.

✓ **Generalized Framework:** We introduce **GENPAS**, a principled three-step sampling framework for constructing training data that unifies existing ones.

✓ **Strong Performance:** Across both benchmark and industrial datasets, we show that generative recommendation models largely benefit from **GENPAS**.

Code & datasets: https://github.com/snap-research/GenPAS