# Dynamic Product Image Generation and Recommendation at Scale for Personalized E-commerce

**Ádám Czapp**
adam-tibor.c@taboola.com
@adam_czapp

**Mátyás Jani**
matyas.j@taboola.com

**Bálint Domián**
balint.d@taboola.com

**Balázs Hidasi**
balazs.h@taboola.com
@balazshidasi

Taboola Budapest

## Good looking creatives at scale

- **Representation** of the recommended item **influences click probability**

| Traditional ad campaigns | Product recommendation |
|---|---|
| Sponsored content | DPA, Marketplaces |
| Few items (creatives) | Many items (products) |
| High quality, hand designed creatives | Simple product images [+simple background] [+template graphics] |

→ Automated generation, high quality creatives

- **BGGEN**: use Stable Diffusion to **generate** appropriate **backgrounds** around recommended products in a **scalable** way to **increase user engagement** with recommendations

| Number of potential creatives: $|I| \times |P| \times |R|$ | | |
|---|---|---|
| $|I|$ | Number of products (catalog size) | Up to millions |
| $|P|$ | Number of prompts (themes) | 1-10 |
| $|R|$ | Number of ad placement resolutions | Few thousands |

## Personalizing the background

- Hard to estimate performance in advance
  - Let online optimization do it!
  - Multi-armed bandit

- Click probability might depend on the
  - user (e.g. location, gender)
  - product (e.g. subcategory, color)
  - placement (e.g. resolution)
- Utilize this information
  - Contextual bandit (LinUCB)
  - One LinUCB per item category

- Selects from a few predefined prompts

or

## Background generation pipeline

### Diffusion based image generation

- Reverse diffusion process produces images from noise
- Classifier Free Guidance
  - Influences generation during inference time

Diffusion process →
← Reverse diffusion process
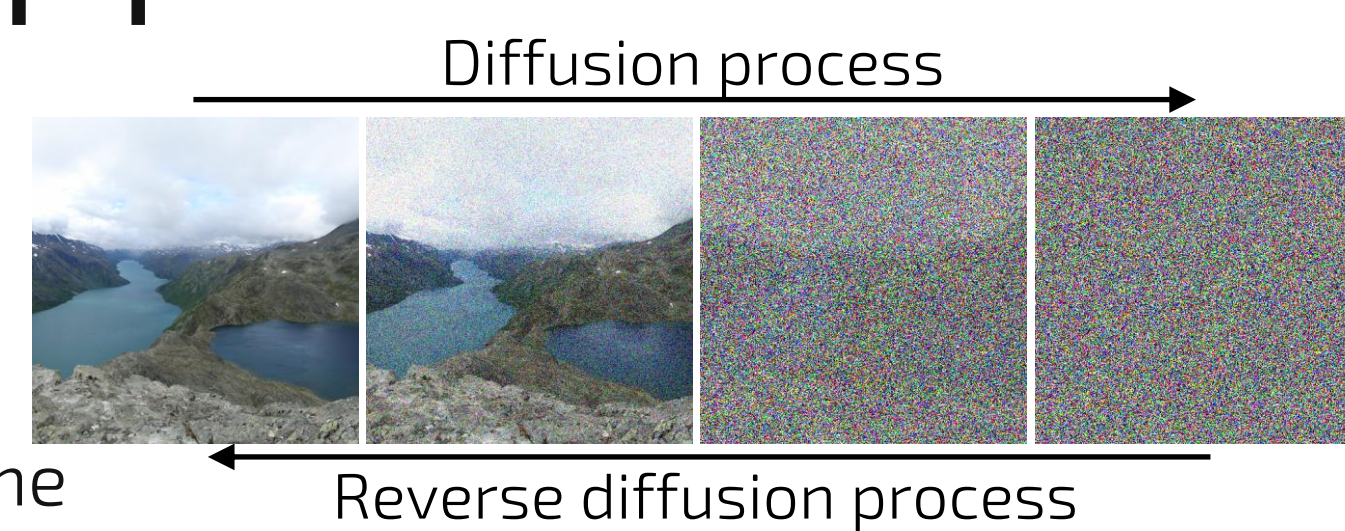
### Inpainting

- Draw outside of the masked area only
- Often produces unwanted artifacts & extends the product

### ControlNet

- Condition the generation on e.g. edges, poses

### Extra conditioning

- on the product further reduces artifacts
- outside of the product changes the color composition of the image
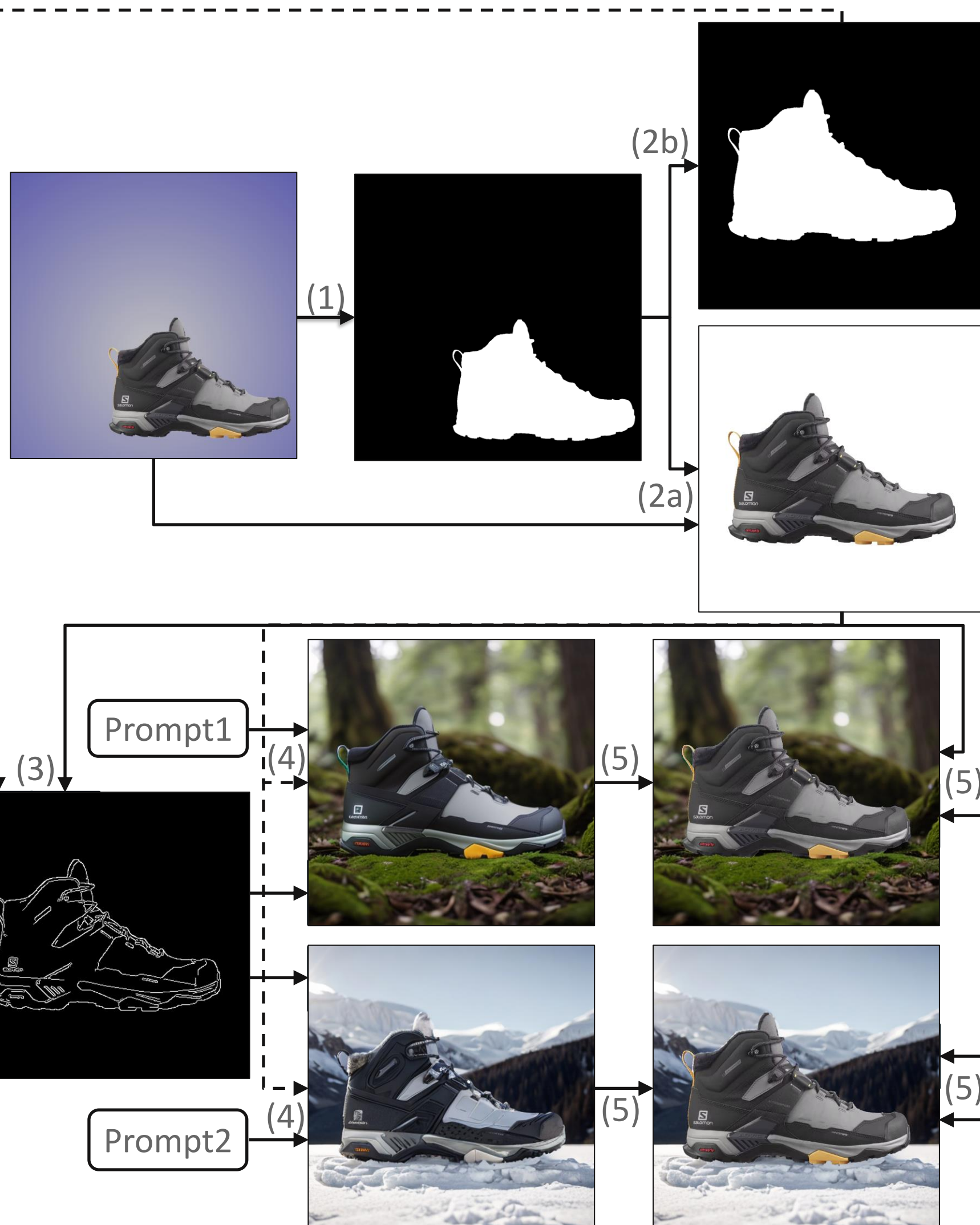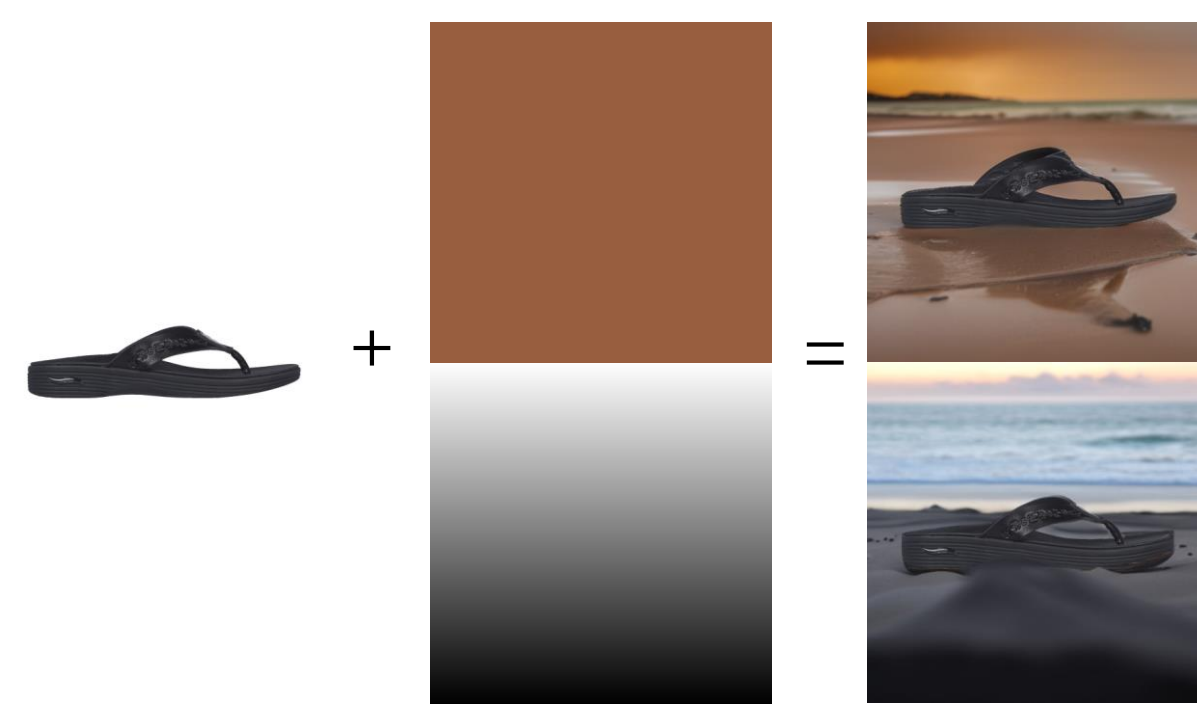
### Steps

(1) Object detection and **masking**.

(2) **Position & scale** the product (and mask) according to the placement. **Remove** original **background**, if applicable.

(3) **Edge detection**
[optional] Reinforce contours using the mask

(4) **Image generation** using the edges as constraints
[optional] Condition on the product to increase quality

(5) **Cut back** the original product onto the generated image.

## System architecture & Scaling
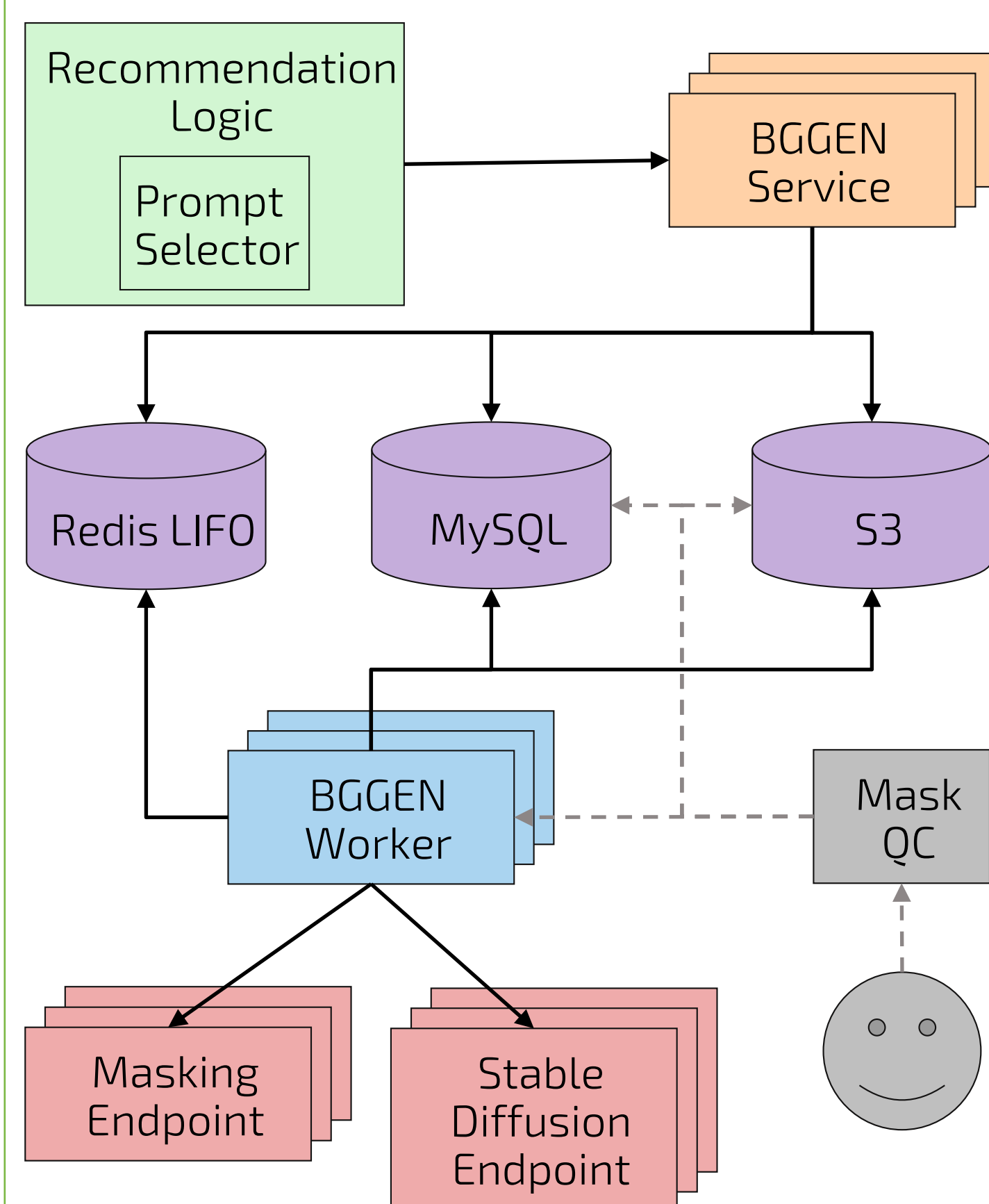
### Recommendation Logic

1. **Select item** for given placement (candidation, ranking)
2. **Select prompt** for item based on category
   (a) random select / (b) bandit / (c) contextual bandit

### BGGEN Service

**Find mask & generated image for bucket in MySQL**
1. **Not available:**
   - Create image generation task on Redis LIFO
2. **Available:**
   - Fetch image from S3
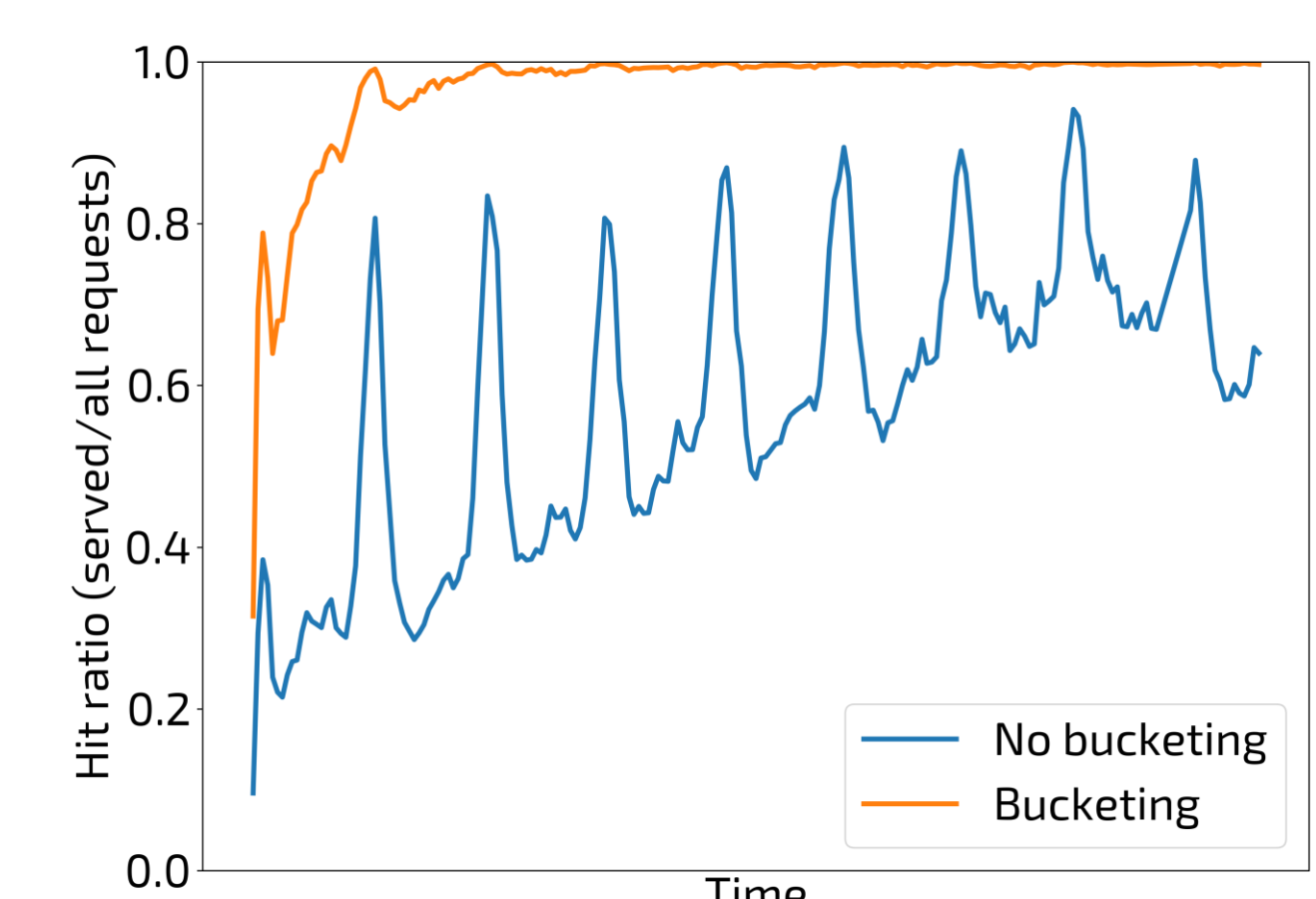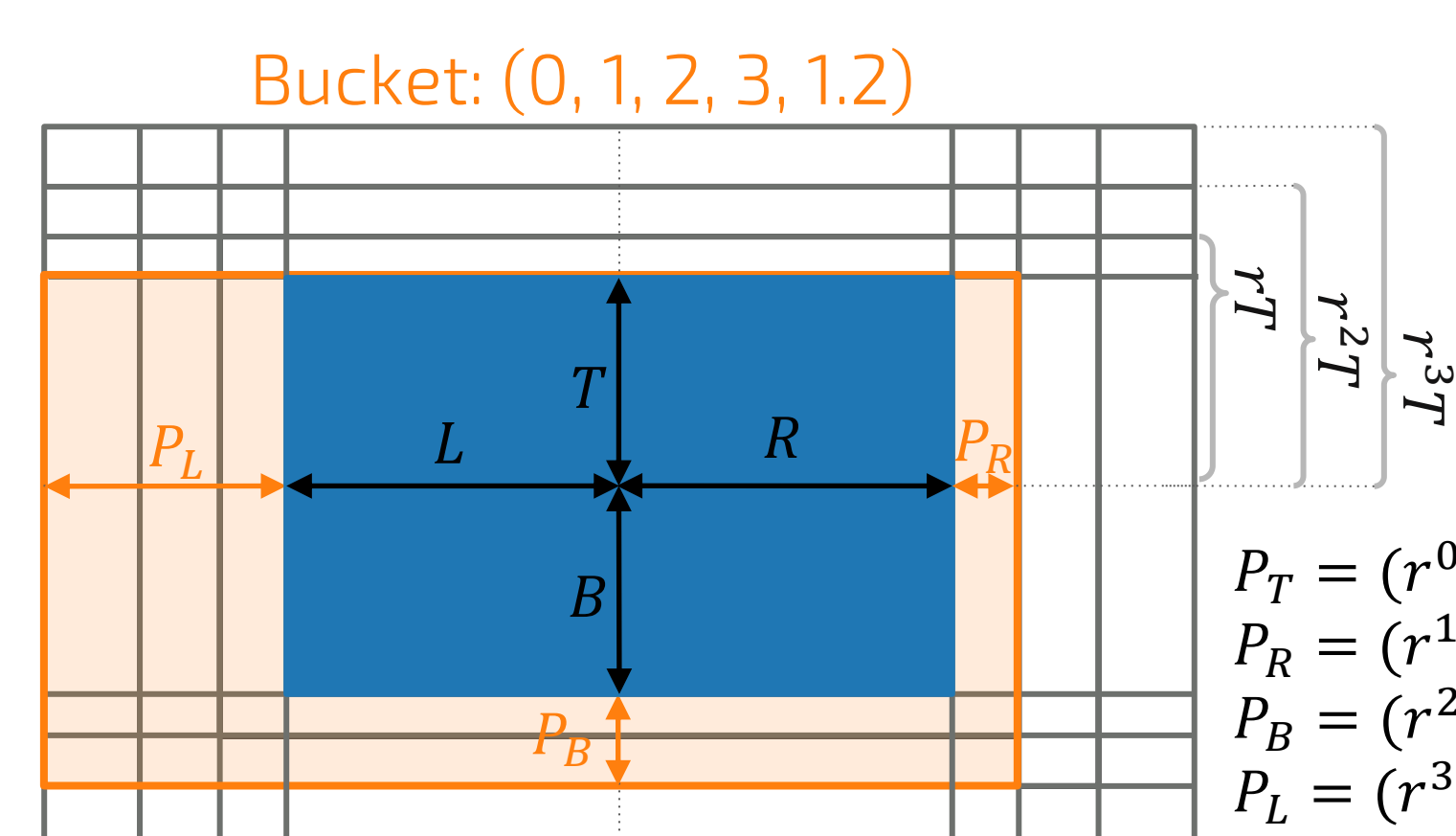   - Crop, resize

### BGGEN Worker

1. **Poll next task** from Redis LIFO
2. **Mask not available:**
   - **Mask computation** request to Masking endpoint
   - Store mask in MySQL & S3
3. Generated image not available:
   - **Image generation** request to SD endpoint
   - **Store generated image** & info in S3 & MySQL

### Bucketing

- Generate a **single image for similar aspect ratios** (for a given product)
- **Target image** (size: $W \times H$) is **cropped**, resized from the appropriate bucket
- Number of **potential requests** is **lowered by 2 magnitudes**

- **Bucket definition**
  - Base aspect ratio = product bounding box
  - Target resolution + fill ratio + positioning → padding $P_D$ for each direction
  - Bucket is identified by $(B_T, B_R, B_B, B_L, r)$
  - $B_D = \left\lceil \log_r \frac{P_D + D}{D} \right\rceil$, where direction $D \in \{T, R, B, L\}$, $r$ is the increment ratio
    - E.g. $r = 1.2$ → bin thresholds increase by 20% relative to the previous one

Bucket: (0, 1, 2, 3, 1.2)

$P_T = (r^0 - 1)T$
$P_R = (r^1 - 1)R$
$P_B = (r^2 - 1)B$
$P_L = (r^3 - 1)L$

Hit ratio (served/all requests) vs Time — No bucketing / Bucketing

## A/B tests & Results

### Setup

- Main KPI: **CTR uplift** over control
- Domain: apparel, clothing, fashion
- Item catalog sizes: 1K – 100K products
- Max. 3 **predefined prompts** per item category

### Observations

- Direct effect on CTR
- Indirect effect on conversions, CPA, etc.
- Positioning & scaling already helps (~5%)
- Background increases CTR by 5-40%
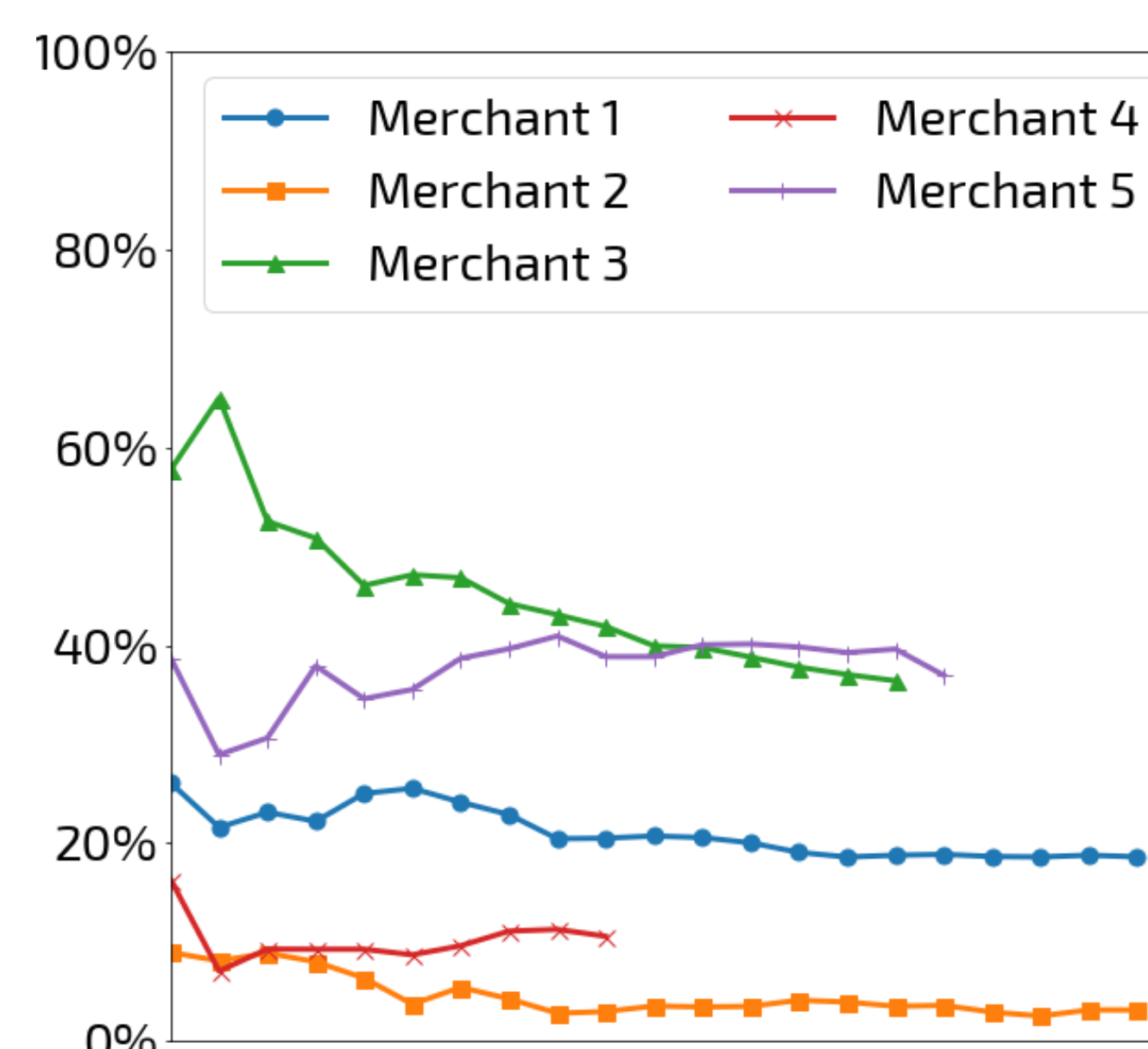- Small improvement by personalization

### Proof of Concept

- **A:** Generated background
- **B:** Positioned product
- **Control:** Product image

Generated background / Positioned product
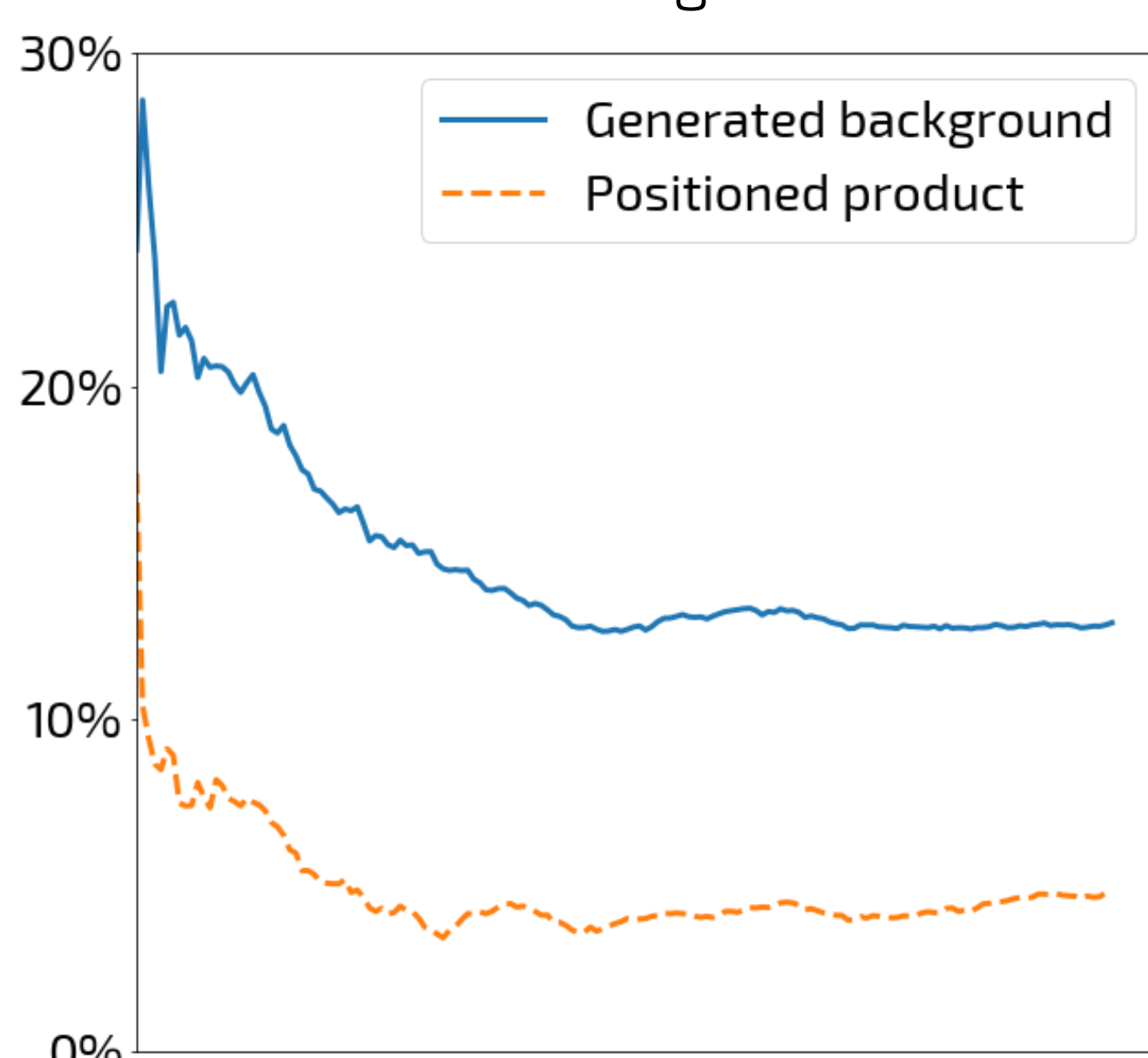
### Improved Pipeline

- **A:** Improved generated background*
- **Control:** Product image
*Uplift between old and new pipelines: ~2%

Merchant 1 / Merchant 2 / Merchant 3 / Merchant 4 / Merchant 5

### Personalization

- **A:** Personalized background
- **Control:** Randomly selected prompt

Personalized background