Constrained Generative Editing for Brand-Aware Creative Optimization

Maxwell Moroz

January 31, 2026

Abstract

Generative editing enables scalable production of brand-specific ad creatives by transforming existing media (e.g., replacing a product can in an image). However, two practical constraints make this difficult to deploy: (i) performance is user-dependent (different audiences respond differently to the same creative), and (ii) edits must satisfy acceptability requirements (brand safety, realism, and asset correctness), which restrict the set of feasible transformations. We model creative generation as a constrained sequential decision problem over (content, brand, edit) actions. We validate the approach in a simulator aligned with the real pipeline and show that naive non-contextual learning fails under heterogeneous users, while cohort-based stationarity restores learnability and approaches a constrained oracle. We then describe a real image pipeline that instantiates the same components using mask-based segmentation and brand-specific edits.

1 Introduction

Digital advertising systems typically optimize *selection* (which creative to show), while treating the creative itself as fixed. In many settings, however, the actionable degree of freedom is also *transformation*: given an image or video from a creator, an advertiser may wish to insert a brand-specific product (e.g., Coke vs. Pepsi) into an existing "slot" such as a can, bottle, or billboard.

This creates a coupled problem. First, the optimal brand/content pairing varies by audience: the same base content can benefit from different brand variants across cohorts. Second, not all transformations are permissible: an edited creative must satisfy acceptability constraints (e.g., logo legibility, realism, forbidden-scene rules), so the system must optimize performance *over a feasible set*.

We propose a two-layer pipeline: (i) a *content factory* that generates feasible brand variants via segmentation, editing, and acceptability filtering; and (ii) a *decision layer* that learns, from feedback, which feasible variants to serve to which audience segments.

Contributions. (1) We formalize brand-specific generative editing as a constrained sequential decision problem over creative variants. (2) We introduce a simulator that mirrors the real pipeline and use it to diagnose a key failure mode: non-contextual bandits collapse in heterogeneous user environments. (3) We show that cohort-based stationarity, combined with simple Thompson sampling on discrete (content, brand, edit) arms and aggregated feedback, yields consistent improvement and approaches a constrained oracle. (4) We outline a real image pipeline that instantiates the same interfaces using mask-based segmentation (e.g., SAM-style models), simple overlay edits, and automated acceptability checks.

2 Setup and problem formulation

2.1 Entities and inputs

A campaign begins with an advertiser brief specifying (a) a set of candidate brands, (b) allowable asset packs (logos/products), (c) targeting cohorts (audience segments), and (d) acceptability constraints.

The system operates on a pool of base media items (images in the initial implementation). For each base image, the pipeline attempts to identify an editable "slot" (e.g., a can region) and to generate a small number of brand-specific variants.

2.2 Actions: selecting and editing creatives

At decision time, the system chooses: (1) a base image from a candidate set, (2) a brand (e.g., Coke vs. Pepsi), and (3) an edit action (e.g., no-edit vs. overlay-edit). The edit action produces a candidate variant of the base image.

2.3 Acceptability as a hard feasibility constraint

Not all edits are allowed. Each proposed variant must pass an acceptability gate that captures realism and policy constraints. Concretely, acceptability can depend on (i) per-image "editability" (some images contain clear, editable slots; others do not), (ii) segmentation/mask quality, (iii) asset correctness (e.g., logo is legible), and (iv) artifact checks.

The decision layer only optimizes over variants that pass this gate, ensuring infeasible creatives are never served.

2.4 Outcome and feedback

When a feasible creative is served to an audience cohort, the system observes an outcome such as click/no-click. This feedback is used to update the serving policy.

A central challenge is that outcomes are heterogeneous: different cohorts have different response distributions for the same creative. If learning ignores this heterogeneity, the estimated value of each creative variant can be washed out, preventing effective exploration and exploitation.

2.5 Simulation-aligned evaluation

To validate the decision layer before deploying real edits, we construct a synthetic environment aligned with the real pipeline: (1) cohorts induce stable response patterns, (2) edits are sometimes feasible and sometimes rejected depending on per-item editability, and (3) brand preference exists independently of editing (so "which brand" is a meaningful decision).

We evaluate policies against baselines (random, no-edit-only) and a constrained oracle that knows the true environment. Our experiments show that cohort-based learning is necessary for non-contextual bandits to reliably approach oracle performance.

3 Mathematical setup

3.1 Objects and embeddings

Let $u \in \mathcal{U}$ denote a user and $c \in \mathcal{C}$ a cohort/segment. Let $v \in \mathcal{V}$ denote a base media item (image/video) and $b \in \mathcal{B}$ denote a brand (advertiser identity).

We assume learned representations (embeddings) $e_u \in R^d$ for users, $e_v \in R^d$ for media items, and $e_b \in R^d$ for brands.

For an edited item v', we write $e_{v'}$ for its embedding (e.g., from a CLIP/VideoCLIP-style encoder). We use this same representation for both user–content affinity and brand–content compatibility, matching the dependency "transform $\rightarrow$ embed $\rightarrow$ score" in our implementation.

3.2 Editing as an action

We model brand-specific editing as an explicit action. Let $a \in \mathcal{A}$ denote an edit action (e.g., no-edit, overlay/inpaint, diffusion edit). Given an original media item v, the transformed media item is

$$v' = T(v, b, a). \tag{1}$$

3.3 Acceptability (feasibility) constraint

Not all edits are allowed. We define an acceptability indicator

$$\mathrm{Acc}(v, b, a) \in \{0, 1\}, \tag{2}$$

which captures brand safety, realism, and asset correctness. The decision layer only considers feasible actions

$$\mathcal{F}(v, b) = \{a \in \mathcal{A} : \mathrm{Acc}(v, b, a) = 1\}. \tag{3}$$

3.4 CTR model

For a user (or cohort) and a feasible edited item v', we model click/no-click as

$$y \sim \mathrm{Bernoulli}(p), \tag{4}$$

with click probability

$$p = \sigma\big(\alpha + s_{uv'} + s_{v'b} + s_{ub}\big), \tag{5}$$

where $\sigma(x) = 1/(1 + e^{-x})$ and the score terms capture (i) user–content affinity, (ii) brand–content compatibility, and (iii) user–brand preference. A simple instantiation is

$$s_{uv'} = e_u^\top e_{v'}, \tag{6}$$

$$s_{v'b} = h_\theta(e_{v'}, e_b), \tag{7}$$

$$s_{ub} = g_\phi(e_u, e_b), \tag{8}$$

where h_θ and g_ϕ are learned compatibility models.

3.5 Constrained decision problem

At decision time, the system chooses a base item v and edit action a for a brand b to maximize expected CTR over a target population D (or cohort distribution), subject to acceptability. Since $\mathrm{Acc}(v, b, a)$ is a hard gate in our setup, the constraint is naturally expressed as set membership:

$$\max_{v,a} E_{u\sim D}\big[\mathrm{CTR}(u, T(v, b, a), b)\big] \tag{9}$$

$$\text{s.t. } a \in \mathcal{F}(v, b). \tag{10}$$

We discuss in Section 5 how this can be generalized to probabilistic acceptability when the gate is uncertain.

3.6 Serving and learning over discrete arms

In our implementation, we treat each feasible triple (v, b, a) as a discrete arm. For each cohort c, we maintain a Beta–Bernoulli posterior over the (unknown) click rate of each arm. If an arm has posterior parameters $(\alpha_{c,v,b,a}, \beta_{c,v,b,a})$ and we observe s clicks out of N impressions, we update

$$\alpha_{c,v,b,a} \leftarrow \alpha_{c,v,b,a} + s, \qquad \beta_{c,v,b,a} \leftarrow \beta_{c,v,b,a} + (N - s). \tag{11}$$

We use Thompson sampling within each cohort by sampling a click rate from each candidate arm's posterior and selecting the feasible arm with the largest sampled value.

Practical considerations. The content factory pre-computes edited variants and their embeddings, so online serving only scores and samples over a candidate set. When $|\mathcal{V}|\,|\mathcal{B}|\,|\mathcal{A}|$ is large, naive enumeration over all arms is not feasible; in practice, we restrict to a retrieved candidate set and/or maintain hierarchical indexing. Finally, while our cohort-based formulation assumes stationarity within cohorts, real systems exhibit drift; a standard mitigation is to use rolling windows or discounting, yielding piecewise-stationary updates.

4 Practical implementation: automated product placement in images

4.1 Overview

We implement the content factory for images as an automated product placement pipeline that generates and reviews brand-specific variants for a base image. Given an advertiser brief specifying a set of brands $\mathcal{B}$ and an object concept to edit (e.g., "drink can"), the pipeline produces (i) one or more slot masks, (ii) edited variants for each brand, and (iii) acceptability scores and reviewer decisions. The resulting approved variants define the feasible arm set used by the decision layer.

4.2 Slot detection and masking (SAM 3)

For each input image v, we run a promptable segmentation model to detect candidate regions corresponding to the editable object class. In our prototype, SAM 3 is prompted with a short open-vocabulary phrase (e.g., "can") and returns a set of instance masks, bounding boxes, and confidence scores. We select the top-scoring mask(s) and store them as binary masks m for downstream editing.

4.3 Variant generation from a mask

For each detected mask m and brand $b \in \mathcal{B}$, we generate an edited image

$$v'_b = T(v, b, a; m), \tag{12}$$

where the edit action a specifies the editing backend. In the simplest implementation, T is a deterministic overlay that composites a brand asset (logo/product render) into the masked region; this provides a controllable baseline. More advanced backends replace the masked region using a learned editor conditioned on the original image, the mask, and a brand reference (text or image), enabling realistic insertion while preserving lighting and scene context.

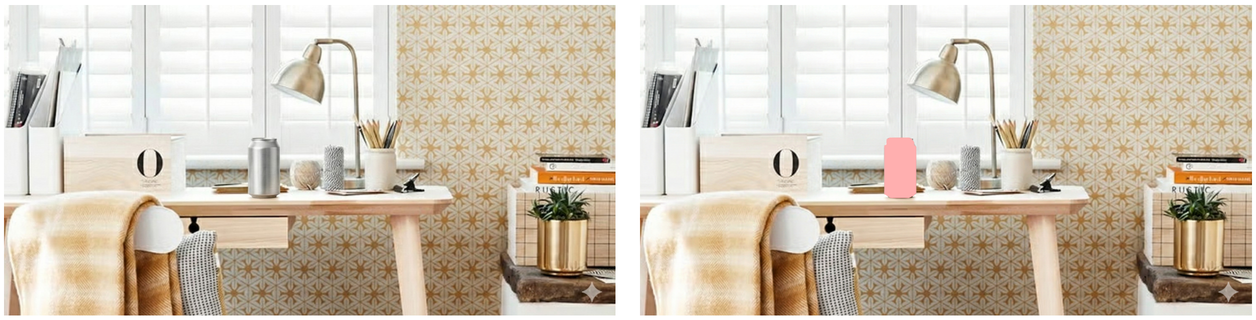

Figure 1: Example outputs after mask-based variant generation.

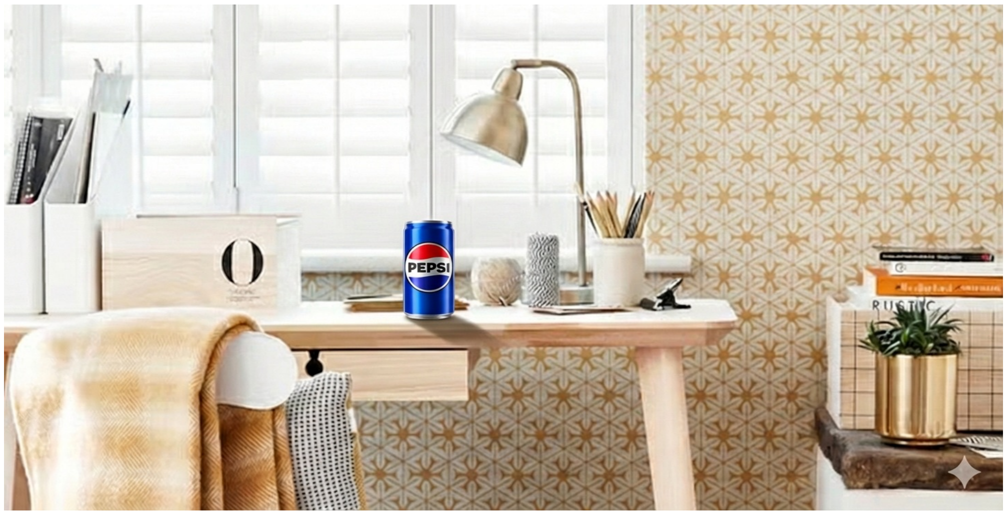

Figure 2: Example ad embedding.

4.4 Automated acceptability scoring

Each generated variant is assigned an acceptability score that approximates the hard feasibility gate used in our model. The score combines lightweight checks such as (i) similarity of the edited region to brand references, (ii) legibility/correctness of inserted marks, and (iii) boundary artifacts near the mask. Variants below thresholds are rejected; borderline variants are routed to manual review.

4.5 Human-in-the-loop review and export

We expose a review interface that displays the original image, the predicted mask, and the per-brand variants side-by-side. Reviewers approve/reject variants and optionally annotate failures. Approved variants are exported into a manifest containing (v, b, a) identifiers, mask paths, variant paths, and acceptability scores. This manifest provides the feasible arm set for the online serving policy.

5 Discussion

Our main formulation treats acceptability as a deterministic gate. In deployments, automated scoring introduces uncertainty; a natural extension is to model acceptability probabilistically and optimize under chance constraints or risk-sensitive objectives.