Extended Abstract

Motivation. Outfit construction is sequential by nature: each chosen piece changes what should come next, and aesthetic quality depends on how the pieces work together rather than on any single item in isolation. Most recommendation systems score items individually or generate an entire slate in one shot. Even when they do not, they typically optimize a single objective signal. Taste does not work that way. What reads as a strong outfit under one aesthetic may read as boring under another, and neither interpretation is necessarily more correct. Our main question is whether a richer, multi-dimensional aesthetic reward leads to better sequential policies than CTL's native binary compatibility signal, and whether improvements on a learned reward proxy survive evaluation by a direct VLM judge.

Method. We formulate outfit construction as a finite-horizon MDP over CLIP embeddings. A small transformer policy selects one item at a time from a candidate pool. Rewards come from a four-axis aesthetic rubric (chromaticity, stylistic synergy, visual harmony, and overall impression) scored by Gemini 2.5 Flash, calibrated via z-scores, and distilled into an MLP reward model trained on 1,000 constructed outfits. We compare three RLOO conditions: a multi-dimensional aesthetic reward, the CTL native binary compatibility signal, and a persona-conditioned extension. We also include a tree-of-styles inference variant as an exploratory extension.

Implementation. The data backbone is the Fashion-2 subset of Complete the Look (CTL). We built a 5,800-item CLIP ViT-B/32 candidate pool and constructed outfits through nearest-neighbor expansion around CTL-annotated seed products. Reward scoring uses Gemini 2.5 Flash, while Gemini 2.5 Pro serves as a strong one-shot baseline. The full pipeline, including dataset construction, VLM scoring, reward-model training, supervised fine-tuning, RLOO training, evaluation, persona-recognizability analysis, and reward-hacking diagnostics, is orchestrated on Modal with WandB logging.

Results. The core RL question is whether a multi-dimensional aesthetic reward beats binary compatibility, and on the learned reward proxy the answer is yes: `rloo_multi_dim` reaches a best mean reward of 0.190 compared to 0.031 for `rloo_binary`, a roughly $6\times$ improvement. The RL pipeline clearly learns to optimize the richer reward signal. On the direct Gemini 2.5 Flash judge, however, that advantage largely disappears. A paired comparison that favors `rloo_multi_dim` 76% of the time under the learned reward falls to 48% under the direct judge. A simple greedy nearest-neighbor heuristic is also unexpectedly strong, performing competitively with, and on shared scenes sometimes exceeding, the Gemini 2.5 Pro one-shot baseline. Post-hoc analyses suggest that candidate-pool design may be a major contributor to this discrepancy. When human-paired CTL items are admitted into the candidate pool, RL recovers far more CTL-paired items than greedy selection. This suggests that the heuristic performs well when the pool is constructed to be highly CLIP-similar to the seed, while RL benefits when the pool contains more diverse cross-category alternatives. The persona part is still exploratory. A follow-up contrastive version makes the outfits differ more across personas, but a VLM judge still does not clearly read those personas back from the final outfits.

Discussion. The main contribution of this work is methodological. We build a complete sequential aesthetic-RL pipeline and show that optimizing a learned reward proxy is not the same as improving outfit quality under a direct judge. The proxy-judge gap is especially interesting because the reward model narrowly missed its pre-registered quality gate on two rubric dimensions. Small imperfections in the reward model appear to become amplified during optimization, producing exactly the kind of failure mode discussed in the reward-hacking literature. Our candidate-pool analysis further suggests that reward design and search-space design are tightly linked in this setting and should not be studied independently.

Conclusion. Multi-dimensional aesthetic rewards clearly outperform binary compatibility on the learned reward proxy, but that advantage does not survive direct VLM evaluation. Even so, the result is informative. We end up with a validated sequential aesthetic-RL pipeline, a concrete example of proxy over-optimization in a subjective visual domain, and evidence that candidate-pool design may be an important bottleneck in aesthetic RL.

Sequential Outfit Curation with Multi-Dimensional Aesthetic Rewards

Esidore Eneinyang
Department of Computer Science
Stanford University
esi@stanford.edu

Chloe Murdoch
Department of Computer Science
Stanford University
cmurdoch25@stanford.edu

Nicole Cortes
Department of Computer Science
Stanford University
nicortes@stanford.edu

Abstract

We build and evaluate a sequential reinforcement learning pipeline for outfit completion: given a seed product, the policy assembles four more items from a candidate pool to produce a coherent outfit. The reward comes from a four-axis aesthetic rubric scored by Gemini 2.5 Flash, calibrated via z-scores, and distilled into a small MLP reward model. We compare three RLOO conditions on top of a CLIP-based transformer policy: a multi-dimensional aesthetic reward, CTL's native binary compatibility signal, and a persona-conditioned extension. The core RL question is whether a multi-dimensional aesthetic reward beats binary compatibility, and on the learned proxy the answer is yes: the multi-dimensional condition reaches a best mean reward of 0.190 versus 0.031. On direct VLM evaluation, however, that advantage mostly disappears: a comparison that favors the multi-dimensional policy 76% of the time under the learned reward falls to 48% under the direct judge. A simple greedy nearest-neighbor baseline is also unusually strong in the primary 50-NN setting. Post-hoc analyses suggest candidate-pool design may be a major reason why. When human-paired CTL items are admitted into the candidate pool, RL substantially outperforms greedy selection. The persona part is still exploratory: a follow-up contrastive version makes the outfits differ more across personas, but a VLM judge still does not clearly read those personas back from the final outfits. We take the main contribution of the paper to be methodological: getting better at a learned aesthetic proxy is not the same as getting better outfits from a direct judge, and reward design in this setting cannot really be separated from search-space design.

1 Introduction

The central question of this project is whether sequential reinforcement learning, supervised by a multi-dimensional aesthetic reward, can build coherent outfits one piece at a time. Outfit construction is sequential by nature: each choice changes what should come next, and quality depends on how the pieces work together rather than on any single item alone. Most existing recommenders either score items in isolation or generate a whole slate in one shot, and even then they usually optimize engagement signals like CTR rather than outfit-level coherence (Afsar et al., 2023; Ie et al., 2019; Hu et al., 2018). We borrow AesRec's multi-dimensional aesthetic scoring methodology (Ye et al., 2026) to go beyond CTL's (Kang et al., 2019) native binary compatibility signal, and we ask whether that richer supervision actually leads to better outfits on a held-out judge.

Stanford CS224R 2026 Final Report

In the milestone, we framed this project more directly as a comparison between sequential RL and one-shot LLM outfit selection. After scaling the full pipeline, the more informative question turned out to be whether richer aesthetic rewards improved sequential policies in a way that survived evaluation by a direct judge.

We built a full pipeline for this question: a 4-axis Gemini-derived aesthetic rubric calibrated against three human raters (mean Pearson 0.55), an MLP reward head distilled from 1,000 VLM-scored outfits, a transformer policy over CLIP item embeddings supervised on CTL multi-product scenes, and three RLOO training conditions (multi-dim, binary, and a persona-conditioned extension). We evaluate the resulting policies on 100 held-out paired test scenes with bootstrap CIs, against a deterministic greedy nearest-neighbor baseline and a Gemini 2.5 Pro one-shot LLM baseline.

On the learned reward proxy, the multi-dimensional condition reaches a $6\times$ best mean reward over the binary condition (Section 5.1). But that is not the full story. When we score the resulting outfits with the underlying VLM judge the proxy was distilled from, the advantage gets much smaller and is no longer statistically convincing (Section 5.1). On that direct judge, the strongest method is greedy nearest-neighbor in CLIP space, which performs competitively with the Pro one-shot LLM (Section 5.2). We use the rest of the paper to unpack that mismatch, show where the proxy and judge separate, and present post-hoc analyses suggesting that candidate-pool design may be a primary contributor to the discrepancy (Section 5.5). We also explore a persona-conditioning extension (Bourdieu, 1984); because those results remain near chance, we treat that part as exploratory rather than central.

Our main contributions are threefold: (1) we build a complete sequential outfit-construction RL pipeline with a VLM-derived multi-axis reward; (2) we show that improvements on a learned aesthetic proxy do not necessarily transfer to a direct judge; and (3) we present post-hoc analyses suggesting that candidate-pool design may be a primary contributor to the proxy-judge discrepancy.

Greedy nearest-neighbor as the heuristic baseline. Throughout the paper we compare the learned policies against *greedy nearest-neighbor* (greedy NN), a deterministic heuristic that at each step picks the candidate whose CLIP embedding is closest to the running mean of the items chosen so far. Greedy NN is not RL: it has no policy network, no reward, and no learning. We still treat it as a serious baseline because in a tightly clustered candidate pool, "keep picking the item that looks most like what you already have" is already a strong rule.

2 Related Work

Deep RL for recommendation. Afsar et al. Afsar et al. (2023) survey the field and show a common pattern: most systems recommend single items and optimize engagement signals like CTR or dwell time, not set-level aesthetic coherence. SLATEQ Ie et al. (2019) moves closer to slate recommendation by decomposing slate value into per-item Q-values, but the slate still arrives in one shot and the reward is still engagement-driven. Closer to our setup, Hu et al. Hu et al. (2018) cast e-commerce ranking as a sequential MDP; even there, the supervisory signal is implicit user feedback rather than aesthetic judgment.

Preference-based learning. The toolkit for training policies against pairwise preference rewards, RLOO and IPO included, comes largely from RLHF-style work Christiano et al. (2017); Rafailov et al. (2023); Ouyang et al. (2022). These methods almost always target single-output settings (a response, a summary, a caption), rather than the multi-step construction of a set.

Reward model misspecification. A well-known failure mode of preference-based RL is that the policy can optimize a learned reward proxy in ways that do not transfer to the underlying preference or judge (Gao et al., 2023; Casper et al., 2023). Most demonstrations of this phenomenon live in text-completion settings; we observe a clean instance of it in subjective aesthetic ranking, where the proxy is a VLM-distilled MLP and the underlying judge is the same VLM family.

Complete the Look. Kang et al. Kang et al. (2019) introduce CTL, a corpus of scene-product pairs that supports compatibility modeling in a contextual visual setting. Its native supervisory signal is binary: scene-item pairs are labeled compatible or not, with no notion of degree along any stylistic

axis. We adopt CTL as our data backbone and treat its binary label as the floor against which richer rewards should be tested.

Multi-dimensional aesthetic scoring (AesRec). Pushing directly against that binary framing, Ye et al. Ye et al. (2026) propose AesRec, which defines an eight-axis aesthetic framework, pairs it with a VLM-prompted scoring pipeline calibrated via z-scores, and folds the resulting signal into a joint compatibility-plus-aesthetic loss. Our setup borrows their scoring methodology and utilizes four of the eight axes; we lift the signal out of supervised scoring and into sequential RL on top of CTL.

The gap. What no existing line of work tests directly is whether sequential RL construction with a multi-dimensional aesthetic reward still looks better when judged by the underlying VLM rather than the learned reward proxy, and how candidate-pool design changes that comparison. CTL benchmarks evaluate scoring rather than generation. AesRec defines richer rewards but stops short of sequential RL. The slate-recommendation literature picks every item at once and does not deal with proxy-versus-judge transfer. Our contribution is to test that exact gap, including the negative results.

3 Method

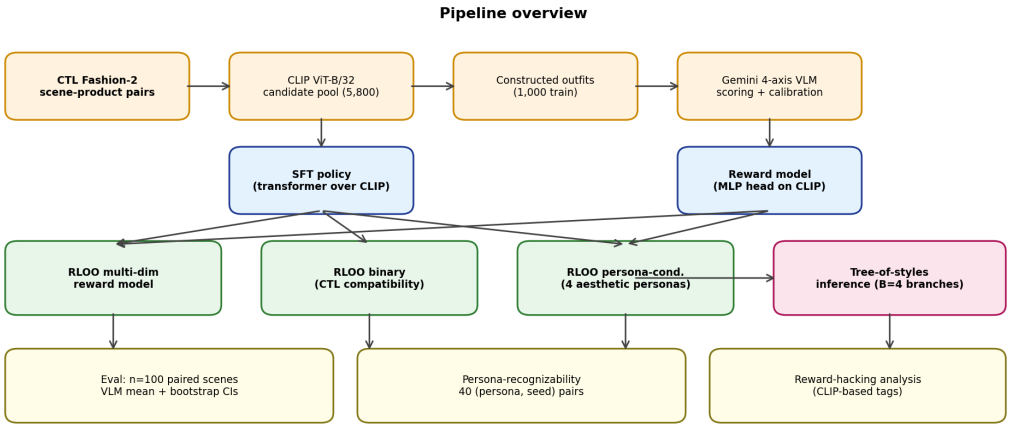

Figure 1: Pipeline overview. CTL Fashion-2 scene-product pairs feed both the constructed-outfit set (used for reward training) and the sequential policy demonstrations (used for SFT). A four-axis VLM rubric is z-score calibrated and distilled into an MLP reward head. RLOO fine-tunes the transformer policy under one of three reward conditions (multi-dim, binary, persona). Tree-of-styles inference expands a beam of persona-targeted branches at decode time.

3.1 Formulation

We frame outfit construction as a finite-horizon MDP. Each episode begins with one seed product drawn from a CTL Fashion-2 scene; the state at step t is the partial outfit $S_t = \{i_1, \ldots, i_{t-1}\}$ together with a candidate pool of the 50 nearest neighbors of the seed in CLIP space. Each action selects the next item, and an episode terminates when the outfit reaches five items (the seed plus four). The terminal reward is a calibrated multi-dimensional aesthetic score $r_\phi(S_T)$, or in the persona-conditioned setting $r_\phi(S_T, p)$ for a target persona p.

3.2 Data

We pulled the Fashion-2 subset of the Complete the Look (CTL) corpus, which gives scene-product pairs with bounding boxes rather than ready-made outfits. To turn this into something an RL agent can train on, we downloaded a working slice of product images, embedded each one with CLIP ViT-B/32 (frozen) to produce a 5,800-item candidate pool, and constructed 1,000 training outfits by taking a CTL-annotated product as the seed and expanding it with four nearest neighbors in CLIP space. The construction is deterministic given a seed, so the same held-out test scenes can be reused across every method we compare, which gives us paired comparisons without extra work.

3.3 Reward Model

We borrow AesRec's scoring methodology without using its data. Four aesthetic axes are defined: *chromaticity* (color harmony), *stylistic synergy* (do the pieces belong together), *visual harmony* (overall coherence), and *overall impression* (gestalt). Axis scores are collected by prompting Gemini 2.5 Flash with a fixed rubric and JSON-output instruction, then z-score-calibrated to debias per-axis scoring distributions. The reward model itself is a small MLP that takes in a mean-pooled CLIP embedding over the items in the outfit and outputs a prediction for the calibrated four-axis vector. A separate persona-conditioned head accepts a persona one-hot embedding alongside the pooled outfit embedding and predicts persona-specific scores.

We pre-registered a held-out gate: per-axis Pearson correlation > 0.40 on a 20% validation split before any RL training proceeds. The persona-conditioned reward model passed its gate (mean Pearson 0.484, range 0.41 to 0.62). The unconditioned reward model failed the gate by 0.03 on two axes (visual_harmony 0.373, overall_impression 0.359). We proceeded with that model under explicit caveat, and in retrospect that shortfall is central to the proxy-judge mismatch reported in Section 5.

3.4 Policy

The policy is a 2-layer, 4-head transformer encoder over CLIP item embeddings, hidden dimension 256, with an optional persona token prepended when conditioning is enabled. At each step, the encoder consumes the sequence of (persona token, seed, picks so far), projects the final hidden state to a query, and dot-products against the candidate pool's CLIP embeddings to produce a distribution over candidates. Already-chosen candidates are masked out.

We supervise the policy on CTL multi-product scenes as expert demonstrations. For each scene with at least two products, we treat one as the seed and the rest as the demonstration trajectory; the policy learns next-item prediction over the seed's 50-nearest-neighbor candidate pool. SFT runs for 10 epochs at learning rate 3×10^{-4}.

3.5 RLOO Training

We fine-tune the SFT policy with RLOO under three reward conditions:

- `rloo_multi_dim`: the unconditioned multi-axis reward model (mean across the four axes).
- `rloo_binary`: the CTL native binary compatibility signal, computed as the fraction of chosen items that co-occur with the seed in any CTL scene.
- `rloo_persona`: the persona-conditioned reward model, with a uniformly sampled persona per rollout at training time.

Per RLOO update, we sample $K = 4$ rollouts per scene, compute leave-one-out advantages $A_i = R_i - \frac{1}{K-1} \sum_{j \neq i} R_j$, and update with REINFORCE plus a KL penalty to the frozen SFT reference policy ($\beta = 0.1$). We use Adam at 1×10^{-5}, gradient clipping at 1.0, and run 200 RLOO iterations per condition with 16 scenes per iteration. Each run takes roughly 30 minutes on a Modal A10.

3.6 Tree-of-Styles Inference

At inference time, the persona-conditioned policy can be queried with any persona one-hot. *Tree-of-styles* runs $B = 4$ branches in parallel, each conditioned on a different style persona (minimalist, streetwear, preppy, Y2K). A diversity penalty is applied across siblings: at each step, a candidate's logit is reduced by $\lambda_{\text{div}} = 1.5$ in every branch where it is not currently top-1 in that branch's own distribution. This is intended to push branches toward distinct items rather than letting them simply converge on a single "consensus best." For the win-rate matrix in Section 5 we show the highest-log-probability branch as the tree-of-styles output and the full grid of four branches per seed is reserved for qualitative figures.

4 Experimental Setup

4.1 Test Scenes

We hold out 100 test scenes from CTL Fashion-2, selected via a deterministic sampler with seed 224. Test scenes are omitted from the SFT training set and from the constructed-outfit set used to train the reward model. The same 100 scenes are used for every method, so that all comparisons are paired. For the Pro one-shot LLM steel-man we use a 25-scene subset due to API budget constraints, and that subset is the first 25 of the same 100 scenes, so paired comparisons against the steel-man are still well-defined on those 25 scenes.

4.2 Methods Compared

Eight methods plus the Pro steel-man:

- `random`: floor baseline (4 random items from the candidate pool).
- `topk_retrieval`: top-4 CLIP-NN of the seed.
- `greedy_nn`: greedy expansion against the running outfit centroid.
- `oneshot_llm`: Gemini 2.5 Pro selects 4 items from the candidate pool in one shot; 2 trials per scene at temperature 0.8, best-of-2 by self-rated overall impression.
- `sft_policy`: SFT-only policy, greedy decode.
- `rloo_multi_dim`, `rloo_binary`, `rloo_persona`: RLOO policies under their respective rewards, greedy decode.
- `tree_of_styles_persona`: tree-of-styles inference on top of the persona-conditioned policy.

4.3 Metrics

VLM mean. Each method's output outfit (seed plus four chosen items) is scored by Gemini 2.5 Flash with the same four-axis rubric used for reward training and the per-outfit mean is reported. This is the underlying judge.

Reward-model score. For comparison purposes, we report the trained MLP reward model's per-outfit prediction. This is the proxy.

Win-rate matrix. For each ordered pair (A, B) of methods, we compute $P(\text{VLM}_A > \text{VLM}_B)$ on the 100 paired scenes (25 for the steel-man).

Bootstrap CI. 10,000 paired bootstrap resamples (seed 42) yield 95% percentile intervals.

Persona-recognizability. For each of 4 personas, we sample 10 seed scenes, run the persona-conditioned policy targeting that persona, and ask Gemini 2.5 Flash to classify the resulting outfit's persona from the four options. Top-1 accuracy is reported overall and per-persona.

Reward-hacking. We tag each of the 800 generated outfits with named failure modes using cheap CLIP-based heuristics: *color_locking* (pairwise cosine of chosen embeddings > 0.85), *category_pileup* (pairwise cosine in $(0.78, 0.85]$), *seed_drift* (seed-to-chosen cosine < 0.25), *mode_collapse* (any chosen item appearing in more than 25% of all rollouts), and *safe_default* (mid-cosine plus low VLM mean).

5 Results

The results follow one main story. First, the RL pipeline does learn to optimize the richer reward signal (Section 5.1). Second, that gain does not survive direct VLM evaluation, and a simple greedy nearest-neighbor baseline is unusually strong in the primary 50-NN setting (Sections 5.1 and 5.2). We then add two secondary analyses: persona conditioning as an exploratory extension (Section 5.4) and a post-hoc candidate-pool diagnostic suggesting that the primary evaluation pool may hide many cross-category alternatives (Section 5.5). We close with a reward-hacking catalog that helps explain why proxy-side gains do not automatically carry over to the judge (Section 5.6).

5.1 Headline: Multi-Dim vs Binary on the Proxy vs the Judge

The proposal's central claim is that multi-dimensional aesthetic rewards beat binary compatibility under sequential RL. To test that, we trained three RLOO conditions for 200 iterations each and tracked per-iteration mean reward on the learned proxy. Figure 2 shows the first part clearly: `rloo_multi_dim` and `rloo_persona` both pull away from `rloo_binary`, with best mean rewards of 0.190 and 0.228 against 0.031. On the proxy, the richer reward is doing what it is supposed to do.

Figure 2: RLOO training curves on the learned reward proxy, 200 iterations per condition. Faint lines are per-iteration mean reward, bold lines are running best. Multi-dim and persona-conditioned both reach 6 to 7× the binary signal on the proxy, showing that the RL pipeline can optimize the richer learned reward.

The harder question is what happens when we hand those same outfits to the underlying VLM judge instead of the MLP reward head. Table 1 reports the same multi-dim vs binary comparison on both metrics.

Table 1: The headline ablation, evaluated on the reward proxy versus the underlying VLM judge. The proxy gap is real and meaningful. It does not survive translation to the judge.

Comparison	Metric	Mean diff	Paired win-rate
`rloo_multi_dim` vs `rloo_binary`	RM-score (proxy)	+0.216	0.76
`rloo_multi_dim` vs `rloo_binary`	VLM mean (judge)	+0.163	0.48

On held-out scenes scored by the same MLP head, the multi-dim vs binary paired win-rate is 0.76. When the same outfits are scored by the underlying VLM judge instead, that drops to 0.48, which is basically chance. The positive mean difference but sub-0.5 judge-side win-rate suggests a small number of larger multi-dim wins mixed with many narrow losses, so the judge-side advantage is not robust.

Figure 3 makes the gap visible at the per-outfit level. Each point is one of the 825 method-scene outfits, plotted with its reward-model score on the x-axis and its VLM mean on the y-axis; the large outlined markers are per-method centroids. If the proxy and the judge mostly agreed, the points would line up more tightly and the method centroids would keep the same ranking on both axes. They do not. The per-outfit Pearson is only 0.34, the trained-policy centroids spread out much more on the proxy axis than on the judge axis, and `greedy_nn` sits high on the judge axis despite a modest proxy score. In other words, the policies got better at the score they were trained on without clearly getting better at the score we cared about most.

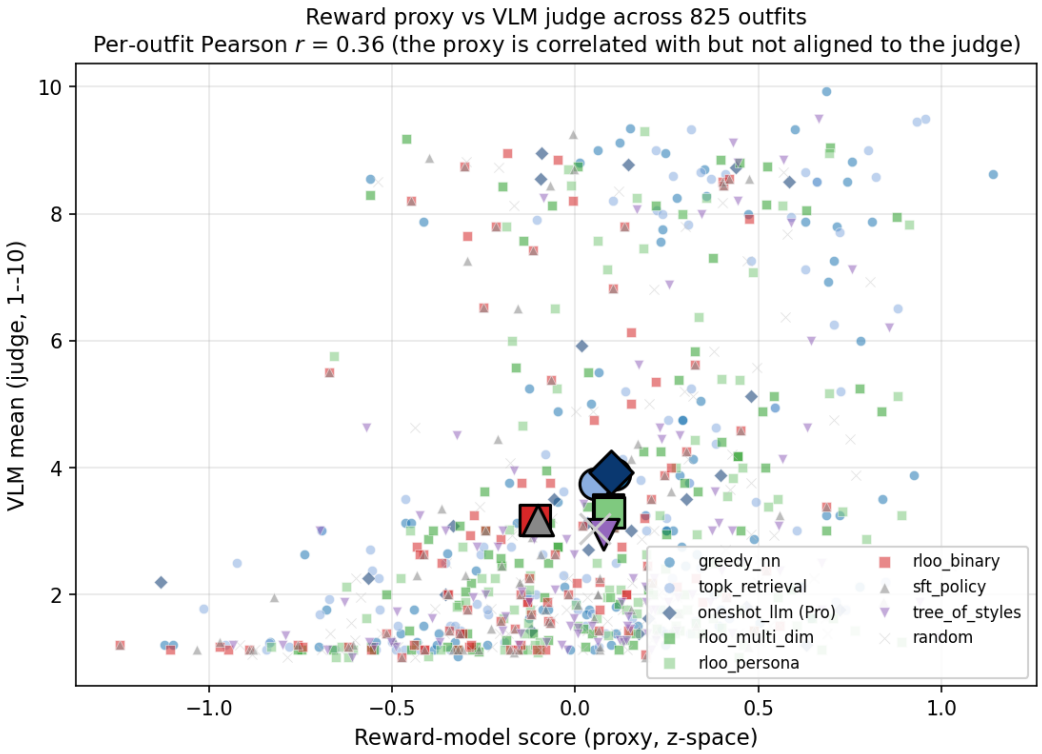

Figure 3: Reward-model proxy (x-axis) vs VLM-mean judge (y-axis) across all 825 method-scene outfits. Small transparent markers are individual outfits; large outlined markers are per-method centroids. The per-outfit Pearson is only 0.34, and the per-method centroids are stretched on the proxy axis but compressed on the judge axis, so a large proxy gap translates into a small judge gap. The misspecification finding in one figure.

5.2 Full Method Comparison on the VLM Judge

Stepping back from the single ablation, Figure 4 and Table 2 report per-method VLM mean with 95% bootstrap CIs across all eight methods plus the Pro steel-man. Among the methods evaluated on all 100 scenes, greedy nearest-neighbor comes out on top, with the Pro one-shot steel-man close to it on the shared 25 scenes. The RL methods cluster together in the middle, all within ± 0.05 of each other. Tree-of-styles falls below random, and `sft_policy` and `rloo_binary` sit near the bottom of the trained-policy group.

Figure 4: Per-method VLM mean on the 100 paired test scenes, with 95% bootstrap CIs. `oneshot_llm` reported on its 25-scene subset (Pro steel-man). The "top" group (`oneshot_llm`, `greedy_nn`, `topk_retrieval`) cleanly dominates the trained policies; the multi-dim vs binary headline ablation is not statistically distinguishable on the VLM judge.

Table 2: Per-method VLM mean (Gemini 2.5 Flash, 4-axis mean per outfit). $n = 100$ scenes for all methods except `oneshot_llm`, which is reported on its 25-scene subset. 95% CIs are paired bootstrap percentiles.

Method	n	VLM mean	95% CI
`oneshot_llm` (Pro steel-man)	25	**3.915**	$[2.906, 5.002]$
`greedy_nn`	100	**3.884**	$[3.344, 4.443]$
`topk_retrieval`	100	3.737	$[3.220, 4.270]$
`rloo_multi_dim`	100	3.330	$[2.853, 3.830]$
`rloo_persona`	100	3.284	$[2.825, 3.774]$
`sft_policy`	100	3.169	$[2.702, 3.666]$
`rloo_binary`	100	3.167	$[2.715, 3.641]$
`random`	100	3.034	$[2.613, 3.483]$
`tree_of_styles_persona`	100	2.938	$[2.525, 3.388]$

5.3 Steel-Man Comparison

The Pro one-shot LLM is the strongest baseline we had budget to run, and its pairwise behavior is useful context. On the shared 25 scenes, `oneshot_llm` beats `rloo_multi_dim` at 0.64, `rloo_binary` at 0.72, `tree_of_styles_persona` at 0.60, and `sft_policy` at 0.80, but it *loses* to `greedy_nn` at 0.36 (paired mean difference -0.74 in favor of greedy). One plausible explanation is simple: the 50-NN candidate pool is dense enough in CLIP space that a centroid heuristic is already very well matched to the available choices.

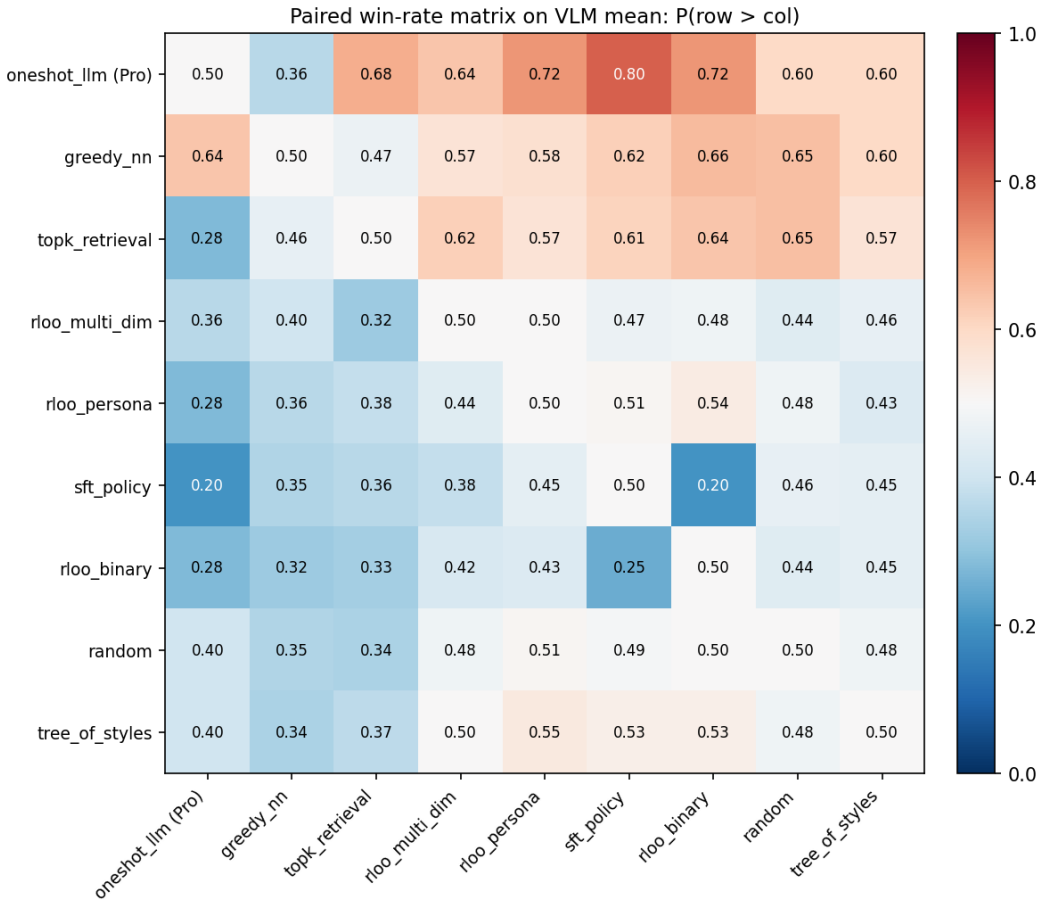

Figure 5: Paired win-rate matrix on VLM mean. Entry (i, j) is $P(\text{method}_i > \text{method}_j)$ on the shared test scenes. The diagonal is fixed at 0.5 by convention. The top rows (`oneshot_llm`, `greedy_nn`, `topk_retrieval`) highlight where the simple baselines dominate the trained policies.

5.4 Persona-Recognizability

The persona-conditioning evaluation is basically a null result. Across 40 (target persona, seed) pairs (10 seeds per persona, 4 personas), Gemini 2.5 Flash classifies the generated outfit's intended persona at 27.5% top-1, barely above the 25% chance floor for a 4-way classification. Per-persona accuracy is 20% (minimalist), 60% (streetwear), 20% (preppy), and 10% (Y2K), and as Figure 6 shows, the confusion matrix is close to flat, with the classifier predicting "streetwear" for most outfits no matter what persona the policy was conditioned on. The reward model did learn per-persona scoring at training time, but the policy that optimized it did not turn that into outfits a VLM judge could reliably read back as the intended persona.

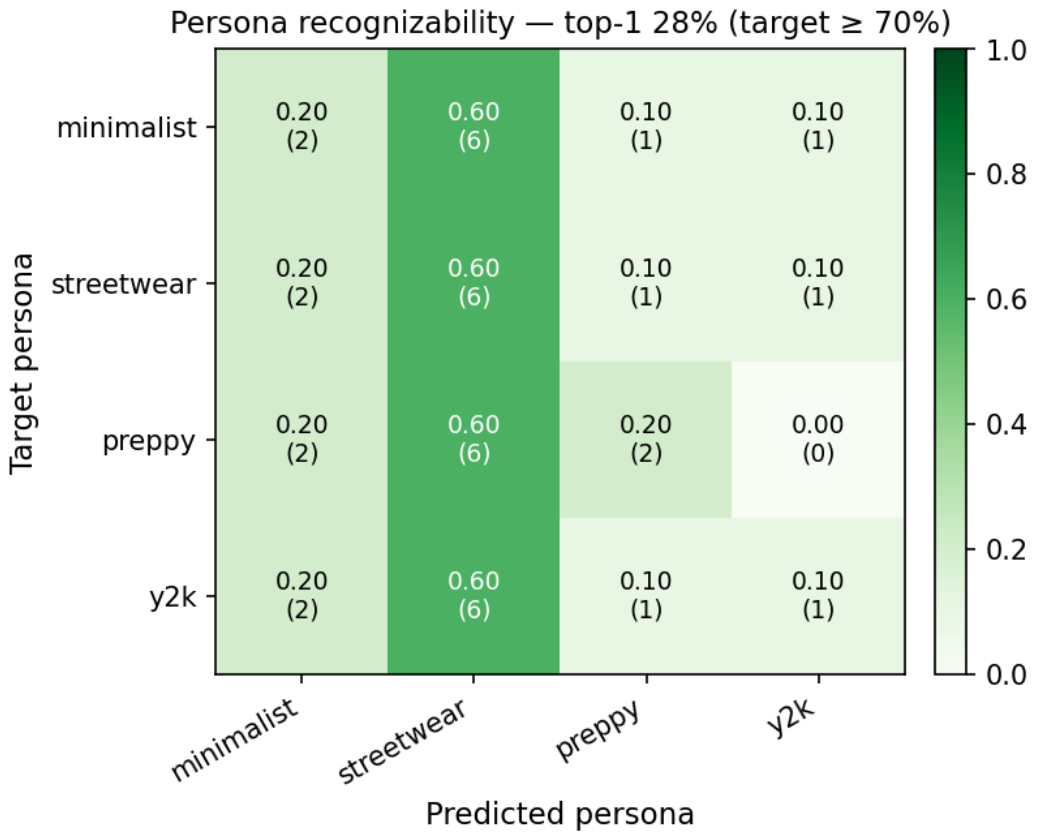

Figure 6: Persona-recognizability confusion matrix (40 pairs, Gemini 2.5 Flash classifier). The "streetwear" column dominates regardless of the target row; intended persona conditioning does not produce VLM-distinguishable aesthetic outfits at this scale.

At the item level, the original persona-conditioned policy is even sharper than the classifier number suggests, and that sharpness is what we used to diagnose where the signal failed (Section 6.3). Under greedy decode, for 4 of 5 randomly sampled seeds the original persona-conditioned policy picked the *exact same* four items across all four target personas, and for the fifth seed only one of the four items changed across personas. The persona token was used at training time (the RLOO reward climbed) but did not survive to the argmax at decode time. The follow-up experiment in Section 6.3 directly examines this collapse and provides a more nuanced explanation than the initial null suggested.

5.5 Post-hoc Candidate-Pool Diagnostic

The greedy-NN result in Section 5.2 prompted a follow-up diagnostic: was the candidate pool itself hiding the kinds of cross-category alternatives RL might otherwise use? This section is *post-hoc* rather than pre-registered, and we treat it as diagnostic evidence rather than as a replacement headline result. Two observations motivate it. First, the 50-NN candidate pool the policies see at test time contains, on average, only **12%** of the items that CTL's human annotators paired with the seed (that is, items that appear in any CTL scene alongside the seed). Widening the pool to the 200 nearest neighbors of the seed in CLIP space lifts this only to 21%. Almost four-fifths of the human-paired alternatives are unreachable to any method, RL or heuristic.

Second, when we construct a candidate pool that explicitly includes the seed's CTL-paired items (the seed's human-paired alternatives plus enough NN to fill to 50 items, applied at test time on the same 100 paired test seeds), the picture changes sharply. Figure 7 reports the mean number of CTL-paired items each method selects per outfit, out of four possible. The deterministic greedy nearest-neighbor heuristic picks essentially none of the planted CTL items, at 0.09 mean hits out of 4. Sequential RL trained on the multi-dimensional aesthetic reward picks 1.70 (paired win-rate 0.77

against greedy, 95% bootstrap CI on the difference $[+1.28, +1.95]$), and sequential RL trained on the binary CTL co-occurrence signal picks 1.83 (paired win-rate 0.86, 95% CI $[+1.42, +2.06]$). Because this test-time pool is different from the primary 50-NN evaluation pool, we interpret the result as a stress test of pool sensitivity rather than as directly comparable headline performance.

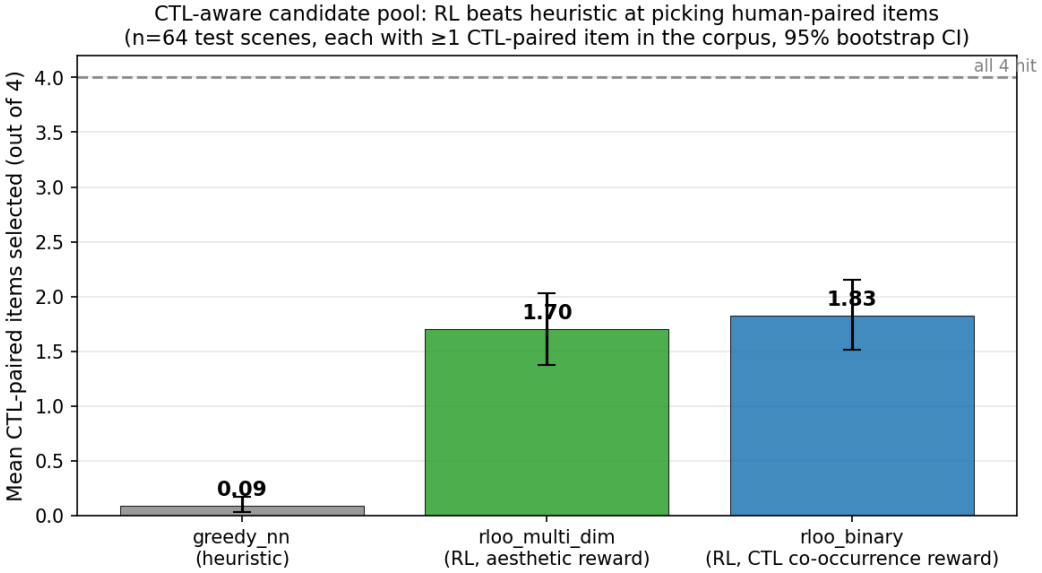

Figure 7: Post-hoc CTL-aware candidate-pool diagnostic. The candidate pool for each test seed includes the seed's CTL-paired items (from the full corpus) plus NN fill to 50 items. Sequential RL picks ~ 1.7 to 1.8 human-paired items per outfit (out of 4); the greedy nearest-neighbor heuristic picks ~ 0.09. We use this figure as evidence that pool composition can strongly affect which alternatives are even available to a method.

We interpret this diagnostic cautiously. In the primary 50-NN setting, the greedy heuristic benefits from a pool that is already a tight CLIP cluster around the seed, where staying near the centroid is a strong rule. When human-paired items are admitted into the pool, RL substantially outperforms greedy selection, indicating that search-space design interacts strongly with reward design. Taken together, these post-hoc analyses suggest candidate-pool design may be a primary contributor to the proxy-judge discrepancy.

5.6 Reward-Hacking Catalog

Within the constraints of the 50-NN candidate pool (mean pairwise CLIP cosine 0.88, high by construction), RL methods diversify meaningfully more than the greedy and top-k baselines. Table 3 reports the fraction of each method's 100 rollouts tagged with named failure modes. `greedy_nn` and `topk_retrieval` hit 98% and 95% color_locking respectively because they push the running outfit toward a tight cluster around the seed; the RL methods relax this to 66 to 79%, with the binary RLOO the most diverse. The catch is that this diversification does not translate into higher VLM scores; if anything, the most-diverse method (`rloo_binary`) and the most-diverse-by-construction method (`tree_of_styles_persona`) sit at the bottom of Table 2.

Table 3: Reward-hacking failure-mode rates per method (CLIP-based tags, $n = 100$ per method). Within the same NN-candidate pool, RL methods diversify 20 to 30 percentage points more than greedy NN, but the extra diversity does not buy higher VLM scores.

Method	color_locking	category_pileup
`greedy_nn`	98%	2%
`topk_retrieval`	95%	5%
`random`	83%	14%
`tree_of_styles_persona`	79%	19%
`rloo_persona`	78%	20%
`rloo_multi_dim`	76%	22%
`sft_policy`	69%	29%
`rloo_binary`	66%	30%

6 Discussion

6.1 Where the Heuristic Becomes Hard to Beat

Table 2 suggests that the candidate pool itself shapes the comparison. The 50-NN pool has mean pairwise CLIP cosine 0.88, so the heuristic of staying close to the seed's CLIP centroid is already strong by construction: almost everything it can choose from already looks similar to the seed and to the items already picked. Greedy nearest-neighbor takes advantage of exactly that structure, and the Pro one-shot LLM often behaves similarly on the same restricted pool. The RL policies, which are trying to optimize a richer signal, end up moving away from that centroid in ways the underlying VLM judge does not consistently reward. Our reading is therefore not "RL does not work for this task," but "RL did not beat the centroid heuristic under this particular candidate-pool design." The CTL-aware follow-up is consistent with that interpretation, though it does not prove it on its own.

6.2 Three Layered Findings

The results tell three related stories, and each one matters for future work.

(1) The RL pipeline works. The 6× gap between `rloo_multi_dim` at best mean reward 0.190 and `rloo_binary` at 0.031 is large and consistent across the 200 training iterations, with the persona-conditioned variant at 0.228. Training is stable across all three conditions. This tells us the policy really can optimize the richer learned reward.

(2) The proxy and the judge can disagree. The same multi-dim vs binary comparison goes from a 0.76 paired win-rate on the proxy to 0.48 on the underlying VLM judge that the proxy was distilled from. This is the main methodological finding. The reward gate missed by 0.03 on two of four axes, and that shortfall is one plausible reason a model can capture a lot of the VLM signal and still be over-optimized in ways the underlying judge does not reward. This is the kind of failure mode discussed in the reward-hacking literature (Gao et al., 2023; Casper et al., 2023), here showing up in a subjective visual setting rather than a text-generation one.

(3) The candidate pool likely shapes what RL can do. Greedy nearest-neighbor in CLIP space, a deterministic heuristic with no learned parameters, posts the highest VLM mean of any method we tested, narrowly tying the Pro one-shot LLM. The simplest reading is that when the candidate pool is a tight 50-NN cluster around the seed, the rule "stay close to the seed" is already hard to beat. The CTL-aware follow-up is consistent with this interpretation. The clearest next experiment is to rerun the pipeline against a more category-diverse candidate pool and see whether the RL methods improve relative to the heuristic there.

6.3 Persona Conditioning: An Exploratory Extension

Beyond the core multi-dim vs binary question, we explored persona conditioning as an extension motivated by the idea that taste is plural rather than singular (Bourdieu, 1984). We picked four aesthetic personas of our own design (*minimalist*, *streetwear*, *preppy*, *Y2K*), trained a persona-conditioned reward model and policy, and added a *tree-of-styles* inference mechanism that runs four

persona-targeted branches in parallel at decode time. The reward model itself learned well (mean Pearson 0.484 vs the VLM, all four axes above the 0.40 gate) and the policy trained to the highest best mean reward of any RLOO condition we ran (0.228). The VLM-based persona-recognizability evaluation, however, came back at 27.5% top-1 against a 25% chance floor. A follow-up diagnostic showed why: under greedy decode, four of five test seeds produced identical outputs across all four target personas, and the within-versus-cross persona CLIP centroid ratio under sampled decode was 1.005. The persona signal mattered during training, but it mostly disappeared by inference time.

We treated this as a training-time problem rather than a decode-time one and ran a focused fix. The new RLOO condition uses a persona-contrastive reward of the form $r_{\text{target}} - \frac{1}{n-1} \sum_{p \neq \text{target}} r_p$, which pushes the gradient to reward per-persona differentiation rather than just absolute persona-conditional score, together with a persona-broadcast architecture that adds the persona embedding to every transformer input token rather than just prepending it. The combined run reached best mean reward 0.301, the highest of any RLOO condition we trained, and four of five test seeds now produce four fully distinct outfits across the four personas under greedy decode, against zero in the original. This suggests that making the policy persona-responsive at the item-selection level is tractable with the architectural and reward changes we tried.

What did not change is the VLM-recognizability number. The persona-conditioned outputs from the contrastive policy come back at 23.75% top-1 on a larger 80-pair sample, not clearly different from the original 27.5%, and both are still near chance. The policy now picks different items per persona, but those differences are not yet the kind a VLM judge reads back as the intended persona. We read this as two separate problems: making a policy respond to persona at the item-selection level, and making those differences visibly legible to a downstream judge. The fix helps with the first problem, not the second.

Tree-of-styles inference, which adds a diversity penalty across branches at decode time, is the worst RL variant on the VLM judge (2.938 mean, below random at 3.034). The penalty pushes branches toward second-choice items in the policy's distribution, and those items happen to be the ones the policy did not pick first because they were less coherent. In a tightly-clustered candidate pool, off-policy diversity ends up trading quality for branch separation. We report this as a concrete pitfall future work on aesthetic beam search should be aware of.

6.4 Limitations and Directions for Future Work

We flag the alternative readings of our results so future work can pick up the most actionable threads.

Candidate pool design is the highest-value lever. The 50-NN pool has mean pairwise CLIP cosine 0.88, which means the heuristic of staying close to the seed is already strong by construction. A larger and more category-diverse pool would weaken that built-in advantage and should give RL more room to help. This is the experiment we would run next if we had another quarter, and it is probably also the one most likely to help visible persona expression.

Reward-model headroom. Our unconditioned reward model passed by a narrow margin on two axes and missed by 0.03 on the other two. Pushing the held-out Pearson higher would likely reduce the proxy-judge gap and make the multi-dim vs binary comparison cleaner on the underlying judge. The 0.40 gate we pre-registered was a reasonable starting point, but a stricter gate plus a per-axis correlation report would have given us a clearer signal about whether to proceed to RL.

One VLM family for both training and eval. The reward model is distilled from Gemini 2.5 Flash and the headline eval also uses Gemini 2.5 Flash, with the persona-recognizability classifier on the same model family. A genuinely independent judge (a different VLM family, or a properly-powered human study) would strengthen the misspecification claim because it would rule out the possibility that we are watching a single model disagree with a downstream-distilled version of itself rather than two genuinely different aesthetic opinions. We did try to use Qwen2.5-VL-7B as an open-weight backup judge, but on a 15-outfit calibration set spanning the Gemini score range Qwen-7B's scores correlated with Gemini's at only Pearson 0.18 (and a stricter anchoring prompt only moved this to 0.18), so at this scale open-weight VLMs are not drop-in substitutes on subjective aesthetic eval.

Reward gate failed by 0.03. The unconditioned reward model came in at visual_harmony 0.373 and overall_impression 0.359 against a pre-registered 0.40 threshold, and we proceeded with that caveat rather than stopping the project. In retrospect, that decision is part of what makes the misspecification finding visible. With a tighter reward model the proxy-judge gap might have been smaller. We do not think the gate miss invalidates the finding; if anything, it shows how a barely-good-enough reward can still be over-optimized in ways the underlying judge does not reward.

Single seed per RL run and Pro steel-man at reduced n. Compute and API budget forced us to run one RLOO seed per condition (rather than the two or three the eval plan called for) and to run the Pro one-shot steel-man on $n = 25$ scenes instead of the full 100. Bootstrap CIs on both comparisons are correspondingly wider than they would be at full scale. Multi-seed replication and a full 100-scene Pro run are the obvious next steps.

Narrow candidate pool. The 50-NN pool means the reachable outfit space is highly seed-dependent, with mean pairwise CLIP cosine 0.88 by construction. A larger and more category-diverse pool (for example, 200 candidates spanning multiple categories per seed rather than 50 NN of a single item) might unlock both better RL gradients and more visible persona expression, because the policy would have items available to express a persona that the current pool simply does not contain.

7 Conclusion

We built and validated a sequential RL pipeline for outfit completion: a 4-axis VLM-derived aesthetic reward calibrated against human raters at Pearson 0.55, an MLP reward head distilled from 1,000 scored outfits, a transformer policy over CLIP item embeddings supervised on Complete the Look demonstrations, and three RLOO training conditions. On the learned reward proxy, the core RL question the proposal posed has a clear answer: the multi-dimensional condition reaches a $6\times$ best mean reward over the binary one, and training is stable across all three conditions.

The main empirical finding comes after that proxy optimization step. On the underlying VLM judge, the multi-dim advantage becomes much smaller and is no longer clearly different, while a deterministic greedy nearest-neighbor heuristic is the strongest method in the primary 50-NN evaluation setting. So we frame the contribution of this paper less as "RL beats the baseline" and more as "aesthetic RL can optimize the learned proxy without clearly improving the direct judge." That matters because it gives a concrete example of reward misspecification in a subjective visual task.

Our post-hoc CTL-aware pool diagnostic suggests one plausible reason the primary setting is so hard for RL to improve upon: the tightly clustered 50-NN pool may exclude many of the cross-category alternatives that humans actually pair with the seed. When those human-paired items are admitted into the pool, RL substantially outperforms greedy selection, indicating that search-space design interacts strongly with reward design. Because that experiment changes the test pool, we still treat it as diagnostic rather than definitive. The clearest next step is to rerun the full pipeline under a more category-diverse candidate-pool design and test whether the proxy-side gains transfer more cleanly there.

8 Team Contributions

The original proposal split the work into three roles: data and reward modeling (Chloe), policy and RL (Esi), and baselines and evaluation (Nicole). In practice, the work evolved as the pipeline came online, and responsibilities were adjusted to match the needs of each stage of the project.

- **Esidore Eneinyang.** Led the policy and RL side of the project, including the transformer policy over CLIP item embeddings, supervised fine-tuning, RLOO training under the binary, multi-dimensional, and persona-conditioned settings, and tree-of-styles inference. Also contributed to the reward-modeling pipeline and the analysis tooling used for reward-hacking, persona-recognizability, and paired evaluation.
- **Nicole Cortes.** Led the data pipeline end-to-end, including CTL ingestion, construction of the 5,800-image product set, creation of the CLIP ViT-B/32 candidate pool, and assembly of the constructed-outfit dataset used for reward training. Also scaled the reward dataset from 100 outfits

at the milestone stage to 1,000 outfits for the final report and contributed to experimental analysis and interpretation.
- **Chloe Murdoch.** Led the baselines and evaluation side of the project, including implementation and rerunning of the baseline assembly methods under a shared comparison protocol, scaled evaluation on the 100-scene test set, and the qualitative galleries used to compare methods. Also contributed to reward-model validation and final analysis of the headline result.

All three of us contributed to the human-rating pass on the 30 validation outfits, interpretation of the main findings, framing of the final paper, and writing and revision of the report.

Changes from Proposal. The proposal positioned multi-dim aesthetic rewards as the primary claim, with persona-conditioning and tree-of-styles inference as supporting extensions. After running the full evaluation pipeline, the finding shifted from "multi-dim beats binary" (the proposal claim) to "multi-dim beats binary on the proxy but not on the underlying judge, and a simple greedy baseline is strongest in the primary evaluation regime" (the actual finding). The persona-conditioning and tree-of-styles results remained exploratory rather than becoming central evidence. We did not drop scope; the methods we set out to build are all built, validated, and reported on. What changed is the story they support.

9 AI Tools Disclosure

We used Claude and OpenAI Codex as implementation and writing aids during the project. These tools were used for some brainstorming on design questions, estimating Modal and Gemini credit costs before long-running jobs, boilerplate support (argument parsing, dataclasses, Modal wrappers), debugging assistance, and revision help on the final written report. The architecture choices for the policy and reward model, the RLOO and loss formulations, the persona rubric, the candidate-pool experiments, the evaluation protocol, the interpretation of the results, and the final research claims were decided by the team.

References

M. Mehdi Afsar, Trafford Crump, and Behrouz Far. 2023. Reinforcement learning based recommender systems: A survey. *Comput. Surveys* 55, 7 (2023), 1–38.

Pierre Bourdieu. 1984. *Distinction: A social critique of the judgement of taste*. Harvard University Press.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research* (2023).

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. *Proceedings of the International Conference on Machine Learning* (2023).

Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement learning to rank in e-commerce search engine: Formalization, analysis, and application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 368–377.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. 2019. SLATEQ: A tractable decomposition for reinforcement learning with recommendation sets. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.

Wang-Cheng Kang, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian McAuley. 2019. Complete the look: Scene-based complementary product recommendation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10532–10541.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* (2023).

Anonymous Ye et al. 2026. AesRec: Multi-dimensional aesthetic scoring for outfit recommendation. *(in submission)* (2026). Referenced per project proposal..

A Reward Model Gate Detail

The held-out per-axis Pearson correlations on the final 1,000-outfit reward dataset are reported in Table 4. The persona-conditioned variant passes; the unconditioned model fails on two axes.

Table 4: Held-out per-axis Pearson on the final 1,000-outfit dataset (20% val split, $n_{\text{val}} \approx 200$). Bold = below the pre-registered 0.40 gate.

Model	chromaticity	stylistic_synergy	visual_harmony	overall_impression
Unconditioned reward	0.500	0.430	**0.373**	**0.359**
Persona-conditioned reward	0.616	0.494	0.422	0.406

B Implementation Details

Compute. All training and evaluation ran on Modal A10 instances (1.10 USD/hr at time of writing) except the reward-hacking analysis, which is CPU-only. Total Modal spend across the project was approximately 35 USD of a 513 USD course credit allocation. Gemini API spend was 25 USD across two prepayment top-ups (15 USD then 10 USD), the final 10 USD covering the headline VLM-mean eval and the Pro steel-man at $n = 25$.

Open-weight VLM calibration. We attempted to use Qwen2.5-VL-7B-Instruct (self-hosted on Modal A10) as an open-weight backup VLM judge after exhausting our first Gemini top-up. On a 15-outfit calibration set spanning the Gemini score range (Gemini means from 1.00 to 9.85), Qwen-7B scores ranged only from 6.88 to 10.0 and correlated with Gemini at Pearson 0.18. A stricter anchoring prompt with explicit low-score examples improved this only marginally (Pearson 0.18 to 0.18). At 7B parameters, open-weight VLMs are not drop-in replacements for closed VLMs on subjective aesthetic eval, which is itself worth flagging for future work in this space.

Code. The full implementation lives in a private repository at `taste-rl`. Key modules: `src/scoring/` (rubric, personas, Gemini client, calibration), `src/reward_model/` (MLP architecture, training, gate check), `src/policy/` (transformer, SFT, RLOO, tree-of-styles inference, baseline adapter), `src/analysis/` (baseline experiment, persona eval, reward hacking, cache recovery utilities), and `modal_app.py` (Modal orchestration).