Causal Direct Preference Optimization with Endogenous Prompts

Liz Lemma Future Detective

January 22, 2026

Abstract

Preference optimization methods such as Direct Preference Optimization (DPO) replace the RLHF pipeline with a simple classification objective derived from the Bradley–Terry model and a KL-regularized reward maximization target. However, by 2026 most preference data is observational: users both choose prompts and provide labels, inducing confounding through user objectives and contexts. Building on DPO's change-of-variables view (policy as an implicit reward model) and recent causal analyses of preference learning, we formalize preference optimization as a causal inference problem with endogenous prompt selection. We define a counterfactual welfare target corresponding to randomized (or policy-specified) prompt assignment and derive an importance-weighted DPO objective that is consistent for the same KL-regularized optimal policy one would obtain under randomized data collection. We provide identification conditions, M-estimation consistency, and finite-sample excess-risk bounds that expose the economic role of overlap (support) and the regularization parameter β. Empirically, we validate on confounded preference benchmarks (e.g., helpful vs harmless prompt-type confounding) and product-style telemetry simulations, showing that causal DPO corrects systematic failures of naive preference optimization under distribution shift. The results supply an econometric backbone for alignment training and specify what must be logged and audited for reliable generalization.

Table of Contents

3. 3. Model: agents (platform, users, labeler), endogenous prompt selection, assignment mechanism for response pairs, and target counterfactual regimes (randomized prompts / reweighted telemetry).

4. 4. Why naive preference optimization fails: illustrate confounding bias and limited overlap with simple examples; link to prompt-type confounding in HH-style datasets.

5. 5. Identification: conditions under which causal preference comparisons are identifiable from observational logs using propensity ratios (with and without observed user objective C).

6. 6. Causal DPO estimator: importance-weighted DPO loss; optional doubly-robust (DR) augmentation; implementation details and what must be logged.

7. 7. Theory: (i) population optimality characterization; (ii) consistency of weighted/DR DPO as M-estimation; (iii) finite-sample excess-risk bounds and dependence on overlap and β.

8. 8. Empirics: controlled confounding (HH helpful/harmless augmentation) + product telemetry simulation; ablations on overlap, propensity misspecification, and proxy C quality.

9. 9. Policy and practice implications: audit checklists for data collection; sensitivity analyses when C is latent; guidance for regulators and deployment teams.

10. 10. Limitations and extensions: unobserved confounding, latent Z discovery, strategic labeling, adaptive data collection; roadmap toward active overlap design and personalized alignment.

1 Introduction

By 2026, most frontier language models are trained and iteratively refined on large corpora of *observational* preference data: users supply prompts in the wild, the platform samples multiple candidate completions, and a human or model-based evaluator selects which completion is better. This data is attractive because it is abundant, continually refreshed, and reflects real usage. It is also structurally risky for alignment and governance, because the data-generating process is not a randomized experiment. Users are not passive "prompt emitters": they choose prompts strategically as a function of their goals, skills, risk tolerance, and prior interaction history. In short, the platform observes preferences *conditional on* an endogenous slice of the world.

This endogeneity matters because preference learning methods such as RLHF and DPO are often deployed as if they were estimating a stable, population-level reward signal. In practice, they estimate a reward signal *filtered through* the prompt distribution induced by the current user base, interface, and product incentives. When the same latent factor that drives prompt choice also drives what the evaluator prefers (e.g., expertise, intent, or safety sensitivity), the resulting learned policy can be systematically miscalibrated for the platform's intended deployment regime. The central issue is not merely "distribution shift" in the colloquial sense, but a specific kind of confounding: user type affects both what we ask the model (the prompts we log) and how we judge the model (the labels we collect).

We can build intuition with a simple story that recurs in real deployments. Suppose there are at least two user objectives: one group wants high-risk capability (e.g., security research, dual-use biology, or evasion techniques) and naturally writes probing prompts; another group wants benign assistance and tends to ask for safe summaries and everyday help. If the platform trains on pairwise preferences without accounting for how the prompt mix is selected, the learned policy may overfit to the prompt distribution of whichever group is overrepresented, more engaged, or more likely to trigger logging. Worse, if the preference signal itself differs by type—for instance, one group prefers direct answers while another prefers refusals or cautious framing—then fitting to observational preferences implicitly chooses a compromise that is optimal for the *observed* mixture rather than the *desired* mixture. The same phenomenon appears in more subtle ways: advanced users write prompts that elicit long-chain reasoning; novices write short prompts; enterprise deployments feature different compliance constraints than consumer chat; and red-team or audit traffic is intentionally adversarial. A naive training objective does not know which of these regimes we intend to optimize for.

From a safety perspective, the failure mode is particularly sharp. If the platform relies on observational preference data, it can inadvertently learn policies that appear aligned on the most common prompts while regressing on rare but high-stakes regions. In the extreme, the model can become well-calibrated on "easy" prompts that users willingly submit, and poorly calibrated on prompts that users avoid because the current model already responds badly (a form of self-selection). This creates a feedback loop: the logged dataset reflects what the model is already good at, which can entrench blind spots. Additionally, observational selection can obscure minority preferences and governance constraints: if vulnerable populations or cautious users disengage, their objectives disappear from the data, even if they are central to the platform's welfare mandate. The result is not only misoptimization but also diminished legitimacy of the training process, because it becomes difficult to argue that the policy is optimized for a transparent, defensible target population.

The technical point we develop in this paper is that DPO, despite avoiding explicit reward model fitting, is still a form of likelihood-based estimation of preference probabilities. As such, it inherits the same causal identification requirements as any estimator trained on confounded observational data. In population terms, unweighted DPO converges to the optimizer of a welfare objective under the observational joint distribution of prompts and user types. If the platform's *target* objective differs—for example, because governance requires evaluating across a standardized prompt suite, or because we wish to represent a different user mix than the one that happened to generate the logs—then naive DPO can be biased relative to the desired counterfactual optimum. Importantly, this bias persists even with infinite data and perfect optimization: it is not an optimization bug, but an estimand mismatch induced by endogeneity.

Our contributions are therefore as much about *measurement and logging* as about optimization. First, we formalize a causal data-generating process for preference labels in which a latent user type drives both prompt choice and evaluative judgments, and we define a target regime that represents the platform's intended welfare objective. Second, we show how importance weighting—using logged propensities of the platform's response-pair assignment mechanism and a specified reweighting from observational to target prompt-type distributions—restores identification of target expectations under standard unconfoundedness and overlap conditions. This yields a weighted DPO objective whose population minimizer corresponds to the KL-regularized optimal policy for the target regime (up to the usual reward equivalence class). Third, we highlight the statistical costs: weak overlap inflates importance weights, increasing variance and effective sample complexity, which in turn interacts with the KL temperature parameter and practical optimization constraints. This tradeoff is not incidental; it is the formal expression of a safety-relevant tension between aggressive optimization on sparse regions and conservative regularization to a trusted reference policy.

A key takeaway is that "DPO without a reward model" does not eliminate the need for causal design. If we want training to be auditable and aligned with a declared welfare objective, we need to treat the logged preference pipeline as an experiment with documented assignment probabilities. Concretely, platforms should (i) log the exact mechanism used to generate response pairs (including any mixture policies, temperature settings, and filters), (ii) preserve sufficient metadata or proxies to model how prompts correlate with user objectives, and (iii) intentionally create overlap by injecting randomized or standardized prompt traffic into the logging stream. These practices are not merely engineering hygiene: they are prerequisites for credible counterfactual claims about what policy would maximize welfare under a specified governance regime. Without them, even an otherwise rigorous training procedure can produce a policy that is "aligned" only with a nontransparent, shifting observational mixture.

Finally, we emphasize limitations and open problems that motivate the rest of the paper. The assumptions required for reweighting—notably overlap and correct propensity specification—can fail in realistic settings where some prompt-type combinations are rare or suppressed by safety filters. Moreover, user types are often latent and only weakly proxied by observable metadata, complicating estimation of target reweighting factors. These issues suggest a governance-alignment interface: the platform can trade off product freedom against evaluability by designing the logging policy and prompt randomization scheme. Our aim is to make this tradeoff explicit. The formalism that follows is not an abstraction for its own sake; it is a way to surface which commitments (to logging, to assignment transparency, to target populations) are required for preference-based training to support robust safety claims.

2 Background

We briefly review three ingredients that we will later combine: (i) the view of Direct Preference Optimization (DPO) as maximum-likelihood estimation of pairwise preferences, (ii) the equivalence between DPO and KL-regularized RLHF (and the resulting closed-form optimum in the nonparametric limit), and (iii) a causal/potential-outcomes framing for preference labels that clarifies what is and is not identified from logged comparisons.

KL-regularized RLHF and the exponential-tilt optimum. A standard abstraction of RLHF is that, for each prompt x, a completion y induces a latent reward $r^*(x, y)$, and we seek a policy $\pi(\cdot \mid x)$ that maximizes expected reward while staying close to a trusted reference policy $\pi_{\text{ref}}(\cdot \mid x)$. This is commonly written as a KL-regularized objective

$$J(\pi) \;=\; \mathbb{E}_x \mathbb{E}_{y \sim \pi(\cdot|x)}[r^*(x, y)] \;-\; \beta\, \mathbb{E}_x \mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x))\,, \qquad (1)$$

where $\beta > 0$ controls the conservatism of the update. In the nonparametric setting (i.e., optimizing over all distributions $\pi(\cdot \mid x)$ with support contained in that of π_{ref}), the pointwise optimizer has a closed form:

$$\pi^*(y \mid x) \;\propto\; \pi_{\mathrm{ref}}(y \mid x) \exp\Big(\tfrac{1}{\beta} r^*(x,y)\Big)\,. \tag{2}$$

This "exponential tilting" expression is useful for two reasons. First, it makes explicit that the KL term is not merely a regularizer but a *choice of geometry* that induces a softmax-like mapping from rewards to policies. Second, it implies that any procedure that implicitly estimates the relative reward differences $r^*(x,y) - r^*(x,y')$ (rather than absolute reward levels) can in principle recover the optimal policy up to a normalizing constant. This connects directly to pairwise preference learning.

Bradley–Terry (BTL) likelihood for pairwise preferences. DPO and related methods are typically trained on data consisting of a prompt x, two candidate completions (y, y'), and a binary label $L \in \{0,1\}$ indicating which completion is preferred. A widely used statistical model for such comparisons is the Bradley–Terry–Luce family. In its simplest form, there exists a latent utility (reward) function r^* such that

$$\mathbb{P}(L = 1 \mid x, y, y') \;=\; \sigma\big(r^*(x,y) - r^*(x,y')\big)\,, \tag{3}$$

where $\sigma(t) = 1/(1 + e^{-t})$ is the logistic sigmoid, and $L = 1$ indicates that y is preferred to y'. The BTL model is attractive because it is (a) invariant to adding a constant to all utilities for a fixed (x), and (b) yields a convex-in-probabilities logistic likelihood for each comparison. Importantly, (3) is naturally compatible with the KL-regularized optimum (2): if we can infer (or fit) the reward differences that explain observed preferences, we can recover a policy via exponential tilting relative to a reference.

Reward equivalence classes and what is identifiable. Preference data identifies *comparative* judgments, not absolute levels. Concretely, under (3), the conditional probability of a label depends on $r^*(x,y) - r^*(x,y')$; therefore, for any function $f(x)$ we have observational equivalence

$$r^*(x,y) \;\sim\; r^*(x,y) + f(x) \qquad \text{since} \qquad \big(r^*(x,y) + f(x)\big) - \big(r^*(x,y') + f(x)\big) = r^*(x,y) - r^*(x,y'). \tag{4}$$

This "reward equivalence class" is not a technical nuisance; it is precisely what makes the KL-regularized policy characterization well-defined. In (2), adding $f(x)$ multiplies the unnormalized density by $\exp(f(x)/\beta)$, which cancels in the per-x normalization. Thus, both the BTL likelihood and the KL-regularized optimum identify the policy only up to an additive baseline in rewards, and this is the correct invariance for preference-based training.

DPO as a reparameterized BTL model. DPO can be understood as directly parameterizing the (log) reward in terms of the policy π_θ relative to the reference π_{ref}. The key reparameterization is

$$r_\theta(x, y) \;=\; \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \;+\; \beta\, b(x), \tag{5}$$

where $b(x)$ is an arbitrary baseline capturing the reward equivalence class. Plugging (5) into (3), the baseline cancels and we obtain a model-implied preference probability

$$\mathbb{P}_\theta(L = 1 \mid x, y, y') \;=\; \sigma\Big(\beta\Big[\log \tfrac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \log \tfrac{\pi_\theta(y'|x)}{\pi_{\text{ref}}(y'|x)}\Big]\Big)\,. \tag{6}$$

Maximizing the corresponding conditional log-likelihood over θ yields the familiar DPO logistic loss. From this perspective, DPO is not "reward-free": it simply performs implicit reward learning in the specific coordinate system induced by the KL regularizer and the reference policy. When the model is well-specified and data are generated from a BTL process, DPO is a maximum-likelihood estimator for these induced preference probabilities.

Why this recovers KL-regularized RLHF. The connection to (2) is now immediate. If the true preference process is governed by some latent r^*, then in the nonparametric limit (allowing π to range over all distributions) the maximum-likelihood solution corresponds to matching the BTL logits, i.e., matching $r^*(x, y) - r^*(x, y')$ with $\beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} - \beta \log \frac{\pi(y'|x)}{\pi_{\text{ref}}(y'|x)}$, which implies

$$\log \frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \;=\; \tfrac{1}{\beta} r^*(x, y) \;+\; \text{const}(x), \tag{7}$$

and hence recovers exactly the exponential-tilt optimum (2). In other words, under idealized assumptions, DPO is a statistically convenient route to the same KL-regularized welfare optimum one would obtain by explicitly fitting a reward model and performing RL with a KL penalty.

A causal (potential-outcomes) view of preference labels. The preceding discussion is purely statistical: it posits a conditional distribution for labels given (x, y, y'). However, our motivating concern is that preference data are generated by a *pipeline* with selection effects. A clean way to express what is "fixed by nature" versus "chosen by the platform" is to introduce potential outcomes for labels. For each user/evaluator context (including any latent objective variables) and each triple (x, y, y'), define a potential label $L(x, y, y') \in \{0, 1\}$ representing the comparison outcome that *would* be observed if the platform presented that pair under that prompt. The BTL assumption is then a causal statement about these potential outcomes:

$$\mathbb{P}\big(L(x, y, y') = 1 \,\big|\, \text{context}\big) \;=\; \sigma\big(r^*(x, y) - r^*(x, y')\big)\,. \tag{8}$$

This framing matters because, in logged systems, we only observe L for the particular (y, y') that the platform chose to show. Consequently, the validity of likelihood-based training (including DPO) depends not just on the correctness of the BTL model but also on whether the assignment of response pairs renders the observed labels representative of the relevant potential outcomes. Informally, we need the displayed pair to behave "as if randomized" conditional on the variables we condition on. Later, when we specify an observational data-generating process and a counterfactual target regime, this potential-outcomes perspective will let us state unconfoundedness and overlap assumptions precisely, and to distinguish an estimand mismatch (optimizing the wrong population/regime) from mere finite-sample noise.

In summary, DPO can be viewed as maximum-likelihood estimation for a BTL preference model expressed in the coordinates of a KL-regularized policy improvement step. The crucial invariances (additive reward baselines) align with the KL-optimal policy characterization, and a causal view of preference labels clarifies which parts of the pipeline must be logged or controlled to justify counterfactual welfare claims.

3 Model: agents, endogenous prompts, and counterfactual regimes

We now specify a minimal model of the preference-data pipeline that is rich enough to capture the confounding failure modes that arise in practice when prompts are user-chosen rather than randomized. The core idea is to distinguish (i) what is endogenous to the deployment environment (users selecting prompts, and heterogeneous objectives) from (ii) what is controlled by the platform (which response pairs are shown for labeling, and how training/evaluation distributions are defined). This separation will let us state precisely which quantities are identified from logs, and which require either experimental intervention or explicit reweighting.

Agents and latent objectives. There are three roles. Users arrive with an objective or "type" $C \in \mathcal{C}$, which we treat as a latent variable capturing stable preference-relevant features (e.g., expertise, risk tolerance, domain, or intent). In deployed systems, C is rarely observed directly; instead one may have a proxy $\hat{C}$ (metadata, user segment, locale, or inferred intent). A platform (the trainer) chooses a reference policy $\pi_{\text{ref}}(y \mid x)$, a KL temperature $\beta > 0$, and a mechanism for generating and logging comparison data. Finally, a labeler/evaluator produces a binary preference label $L \in \{0, 1\}$ when shown a prompt x and two completions (y, y'). We interpret the labeler as implementing the Bradley–Terry choice rule at the causal level, i.e.,

$$\mathbb{P}(L = 1 \mid x, y, y', c) \; = \; \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big) \, , \tag{9}$$

where $r^*(x, y, c)$ is the latent utility induced for type c by completion y under prompt x. This formulation makes explicit that preferences are type-conditional; aggregation across users will generally depend on the type mix.

Endogenous prompt selection as the source of confounding. We model prompt choice as an action taken by the user (possibly adaptively over time). In the static abstraction used for our analysis, we write

$$C \sim p_O(c), \qquad X \sim p_O(x \mid c), \tag{10}$$

where p_O denotes the *observational* distribution induced by deployment. The key point is that $p_O(x \mid c)$ may be highly non-uniform: different user types ask systematically different questions, and they do so in ways that are correlated with how they would evaluate responses. This is precisely the classical confounding structure: C affects both the "treatment assignment" (which prompts are observed) and the outcome distribution (which completions are preferred via r^*). In addition, $p_O(x \mid c)$ may itself depend on the platform's historical behavior (e.g., users learn which prompts elicit helpful answers), yielding feedback loops; our static p_O should be read as a snapshot of that equilibrium. From an alignment perspective, this is not merely a statistical nuance: it means that training on logged prompts can overweight the objectives of heavy users, power users, or particular segments, even if the platform intends to optimize welfare for a different population.

Platform-controlled assignment of response pairs. Conditional on the observed prompt (and any conditioning variables the platform uses), the platform selects a pair of candidate completions to be compared. We represent this via an assignment mechanism

$$(Y, Y') \sim g_O(\cdot, \cdot \mid X, C), \tag{11}$$

where g_O is the *observational* comparison generator. In standard RLHF pipelines, g_O is induced by sampling from one or more model policies (e.g., current policy versus reference, or two samples from the same policy at different temperatures), potentially filtered by heuristics. Two aspects are safety-critical. First, g_O must be *logged* (or otherwise reconstructible) to support any counterfactual claims; in particular, we need the propensity $g_O(y, y' \mid x, c)$ or an estimator thereof. Second, g_O should maintain *overlap*: if some completions are essentially never shown for certain prompts or types, then preference information about those regions is not learnable from the data without additional exploration, and training may silently extrapolate in unsafe ways.

After the pair is generated, the labeler produces

$$L \sim \mathrm{Bernoulli}\big(\sigma(r^*(X, Y, C) - r^*(X, Y', C))\big)\,, \tag{12}$$

and the platform logs (at minimum) (X, Y, Y', L) along with the propensities of the mechanism that generated (Y, Y'). When C is unobserved, the log contains only a proxy $\hat{C}$ or nothing at all; we treat this as a central limitation rather than an afterthought, because it determines whether reweighting to a desired objective mix is feasible.

Observational versus target (counterfactual) regimes. The platform's ultimate goal is not necessarily to optimize average utility under $p_O(x, c)$, but under a *target* regime $p_T(x, c)$ that reflects a policy choice about which users and which prompts should matter. We consider targets that break the X–C confounding by design, for example:

1. *Randomized prompts:* hold the type distribution fixed while randomizing prompts within type, e.g., $p_T(x, c) = p_T(c)p_T(x \mid c)$ where $p_T(x \mid c)$ is specified by an experimental prompt set or a curriculum;
2. *Reweighted telemetry:* define p_T by reweighting observed traffic to match a governance-mandated mix (e.g., downweight power users, upweight underrepresented domains, or equalize across locales).

We also allow the target to use a different response-pair generator $g_T(y, y' \mid x, c)$ (e.g., if evaluation compares different model snapshots than those used during data collection), though in many deployments $g_T = g_O$.

The welfare objective we will later analyze is the KL-regularized value under the target regime,

$$J(\pi) \;=\; \mathbb{E}_{(x,c)\sim p_T}\mathbb{E}_{y\sim\pi(\cdot|x)}[r^*(x, y, c)] \;-\; \beta\,\mathbb{E}_{(x,c)\sim p_T}\mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))\,. \tag{13}$$

This makes the normative choice explicit: the platform is optimizing *target-population welfare*, not necessarily the welfare of the users who happened to generate the bulk of the logs.

Change of measure and the role of logged propensities. To connect observational data to target objectives, we will use importance weights that convert expectations under the observational regime into expectations under the target regime. At the level of joint tuples (x, c, y, y'), the Radon–Nikodym derivative takes the form

$$w(x, y, y', c) \;=\; \frac{p_T(x, c)}{p_O(x, c)} \cdot \frac{g_T(y, y' \mid x, c)}{g_O(y, y' \mid x, c)}. \tag{14}$$

Equation (14) is not merely a technical artifact; it encodes a concrete operational requirement. If the platform does not log (or cannot reconstruct) the assignment probabilities g_O, then it cannot, in general, certify that its preference-training procedure is optimizing (13) rather than some uncontrolled mixture. This links directly to governance and verification: auditing training data for propensity logging and overlap is a prerequisite for making credible statements about counterfactual alignment objectives.

Assumptions, limitations, and safety-relevant failure modes. Our subsequent results rely on two substantive conditions. First, conditional on (X, C), the displayed pair (Y, Y') must be "as-if randomized" with respect to the labeler's potential outcomes; otherwise, the platform can induce selection on unobservables by preferentially showing easy-to-judge or policy-favorable pairs. Second, there must be overlap: the weights (14) should not explode. In practice, overlap is threatened by aggressive filtering, by highly specialized prompts that only appear for narrow user segments, and by generator policies that collapse onto a small set of outputs.

Two additional caveats are worth flagging. If C is latent and only weakly proxied, then even perfect propensity logging does not identify $p_O(c \mid x)$, and reweighting to a desired type mix becomes ill-posed; this is an *information problem*, not a modeling bug. Moreover, users may strategically adapt prompts to elicit certain behaviors (including unsafe ones), meaning that shifting to a target prompt distribution can change not only what we evaluate but also what users learn to ask. These are precisely the situations where "naive" preference optimization can look successful on logged data while degrading real-world safety under deployment shifts.

4 Why naive preference optimization fails

Naive preference optimization methods (including unweighted DPO-style objectives) can look compelling because they are statistically efficient and operationally simple: we collect preference comparisons on whatever prompts users happen to generate, fit a policy to predict those preferences, and deploy the resulting model. The failure mode is that the procedure is implicitly optimizing welfare under the *observational* mixture of user types and prompts, not under the *target* regime the platform actually cares about (e.g., a governance-mandated population, a stress-test distribution, or a counterfactual in which prompts are randomized within type). When prompt choice is endogenous, this discrepancy is not a second-order nuisance; it can qualitatively change which behaviors are reinforced.

Confounding bias: the observational winner need not be the target winner. The core issue is that preference data identify $\mathbb{P}_O(L = 1 \mid x, y, y')$, which is an average over the (possibly highly skewed) type distribution $p_O(c \mid x)$. But the platform's normative objective is typically an expectation under $p_T(c \mid x)$ (or $p_T(x, c)$ more generally). Unless these conditional mixes coincide, optimizing against observational preferences can select a different policy than the target welfare maximizer.

A minimal example makes the point. Consider two user types $c \in \{c_1, c_2\}$, a single prompt x, and two candidate completions y_A, y_B. Suppose types disagree:

$$r^*(x, y_A, c_1) > r^*(x, y_B, c_1), \qquad r^*(x, y_A, c_2) < r^*(x, y_B, c_2).$$

If the observational traffic is dominated by c_1 (say $p_O(c_1 \mid x) = 0.9$) while the target population is balanced (say $p_T(c_1 \mid x) = 0.5$), then the observational Bradley–Terry probability $\mathbb{P}_O(L = 1 \mid x, y_A, y_B)$ can be close to 1 even though the target-averaged preference is near $1/2$ or even favors y_B. In this setting, unweighted DPO is behaving exactly as designed—it is fitting the observational preference distribution—but it is misaligned with the platform's intended welfare criterion. In safety terms, this is the mechanism by which heavy-user segments (or high-volume domains) can dominate training, even if the platform intends to optimize for a broader or different population.

Endogenous prompts amplify the problem by changing the comparison set. The previous example held x fixed. The more realistic failure is that C changes *which prompts are asked*, so the platform sees different regions of $\mathcal{X}$ for different types. Consider two prompts x_1, x_2 and two types c_1, c_2. Suppose c_1 mostly asks x_1 and c_2 mostly asks x_2: $p_O(x_1 \mid c_1) \approx 1$, $p_O(x_2 \mid c_2) \approx 1$. Now suppose there are two stylistic behaviors: y_{bold} (direct, high-capability, potentially risky) and y_{caut} (cautious, more refusal-prone). Assume c_1 wants boldness on x_1 (e.g., technical users), while c_2 wants caution on x_2 (e.g., safety-sensitive users):

$$r^*(x_1, y_{\text{bold}}, c_1) > r^*(x_1, y_{\text{caut}}, c_1), \qquad r^*(x_2, y_{\text{bold}}, c_2) < r^*(x_2, y_{\text{caut}}, c_2).$$

If the target regime breaks the prompt–type linkage (e.g., by randomized prompts within type, or by reweighting to equalize domains), then the policy must trade off these behaviors under a *different* joint distribution over (x, c) than the one implied by logs. Naively optimizing on observational data effectively solves a different KL-regularized objective $J_O(\pi)$ that substitutes p_O for p_T. This can induce systematic over-optimization for whichever prompt–type pairs are overrepresented in p_O, producing a model that is "aligned to the logs" but misaligned to the counterfactual evaluation.

Selection effects in pair generation can create spurious safety. Even holding (x, c) fixed, we must distinguish the distribution of candidate pairs (y, y') shown to the labeler from the distribution of outputs the deployed policy will generate. In many pipelines, $g_O(y, y' \mid x, c)$ is not "two independent samples from the current model" but a complicated product of sampling, filtering, deduplication, and safety heuristics. This can create a subtle problem: the platform might preferentially show comparisons that are easy to judge, or that avoid extreme failures, thereby collecting labels that make the system appear safe while leaving key regions unobserved. Formally, if g_O has low support on dangerous but plausible completions, then preferences provide little information about how to rank those completions, and optimization can drift there at deployment when the policy distribution changes.

Limited overlap: when weights (or uncertainty) blow up, learning becomes extrapolation. Overlap is the condition that the observational process provides non-negligible probability of seeing the comparisons needed to evaluate the target objective. In our setting, overlap must hold both for prompts/types (the support of $p_T(x, c)$ must be covered by $p_O(x, c)$) and for pair generation (the support of $g_T(y, y' \mid x, c)$ must be covered by $g_O(y, y' \mid x, c)$). When overlap fails, there is no amount of cleverness in the loss function that can recover counterfactual preferences without additional assumptions or new data.

A simple manifestation is rare-prompt fragility. Suppose a safety-critical prompt family x_{rare} (e.g., uncommon but high-impact biosecurity queries) has tiny probability under p_O, while the platform's evaluation regime intentionally upweights it under p_T. Then the importance ratio $\frac{p_T(x_{\text{rare}},c)}{p_O(x_{\text{rare}},c)}$ becomes large, and any estimator that tries to correct the mismatch inherits high variance. Practically, the learned policy will be determined by a small number of comparisons, often dominated by idiosyncratic labels or by systematic artifacts of the generator policy used to produce candidates on that rare prompt set.

How this shows up in HH-style datasets. These issues are not hypothetical; they are structurally encouraged by widely used preference datasets that mix heterogeneous sources of prompts and heterogeneous evaluation criteria. In HH-style data collection, prompts often come from a combination of "helpfulness" prompts (benign user requests), "harmlessness" prompts (adversarial or policy-violating queries), and occasionally synthetic or red-team sources. The latent "type" variable in our model can be read as the intent/domain of the prompter and the corresponding normative standard applied by labelers (e.g., maximize helpfulness subject to a safety policy). If harmful prompts are disproportionately generated by a red-team process while benign prompts are generated by regular users, then C and X are statistically entangled: the dataset effectively observes different regions of $\mathcal{X}$ under different implicit objectives. Training a single policy on the pooled dataset without explicitly representing or reweighting this structure can yield the familiar pathologies: over-refusal on benign prompts (because the harmful region is overrepresented or more strongly labeled) or under-refusal in safety-critical corners (because the model never saw informative comparisons there due to filtering and lack of overlap).

Safety implications and the transition to identification. From a safety perspective, the combination of confounding and limited overlap produces a hazardous pattern: apparent progress on in-distribution preference metrics can coincide with degraded performance under counterfactual regimes (new user mixes, different prompt curricula, or stress tests), and the degradation is hard to diagnose post hoc because the logs do not identify what would have happened under alternative prompt/type mixtures. This is why we treat propensity logging and explicit target-regime specification as first-class requirements. In the next section, we formalize when and how causal preference quantities are identifiable from observational logs via propensity ratios, and what changes when the user objective C is observed versus latent.

5 Identification: when causal preference comparisons are recoverable from logs

Before we choose an estimator, we need to know what the logged data *can* identify about the counterfactual regime we actually care about. The key object is not a pointwise "true reward" $r^*(x, y, c)$ (which is only defined up to additive $f(x, c)$), but rather the *causal* preference probabilities and the target-regime expectations they induce.

Set-up as a change of measure. Let the observational logging process induce a joint distribution over (X, C, Y, Y', L) of the form

$$p_O(x, c, y, y', \ell) = p_O(x, c)\, g_O(y, y' \mid x, c)\, \mathrm{Bernoulli}\big(\ell;\ \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big)\big)\ ,$$

and define a target regime

$$p_T(x, c, y, y', \ell) = p_T(x, c)\, g_T(y, y' \mid x, c)\, \mathrm{Bernoulli}\big(\ell;\ \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big)\big)\ .$$

The substantive modeling assumptions live in two places: (i) the Bradley–Terry structure for preferences given (x, c), and (ii) a conditional ignorability statement for pair assignment (our unconfoundedness assumption), which ensures that the observed label L is the correct draw from the causal preference distribution for the displayed pair (y, y') conditional on (x, c). Under these assumptions, the only difference between O and T is the distribution over contexts (x, c) and the assignment mechanism over pairs $g(\cdot \mid x, c)$.

Identification with observed C: importance weights are sufficient. When the user objective/type C is observed (or equivalently, when we have a proxy that is sufficient for the relevant effect modification), identification reduces to standard importance reweighting. Define the Radon–Nikodym derivative between regimes:

$$w(x, y, y', c) \ := \ \frac{p_T(x, c)}{p_O(x, c)} \cdot \frac{g_T(y, y' \mid x, c)}{g_O(y, y' \mid x, c)}.$$

Under overlap (i.e., the denominator is bounded away from zero on the support of the numerator), we obtain, for any integrable measurable f,

$$\mathbb{E}_T\big[f(X, C, Y, Y', L)\big] = \mathbb{E}_O\big[w(X, Y, Y', C)\, f(X, C, Y, Y', L)\big].$$

This is the basic identification statement we will rely on: if we can compute (or consistently estimate) w, then we can rewrite any target-regime population objective as an observational expectation with weights.

Two special cases are worth highlighting because they correspond to common deployment choices.

Targeting only prompts/types. If the platform keeps the same pair mechanism in training and evaluation (so $g_T = g_O$), then the weight simplifies to

$$w(x, y, y', c) = \frac{p_T(x, c)}{p_O(x, c)}.$$

This is the regime in which we are correcting purely for endogenous prompt selection and type mixture shift, while treating the candidate-generation pipeline as fixed.

Targeting a different candidate generator. If the platform intends to evaluate welfare under a different pair mechanism (e.g., different sampling temperature, different filtering, or a different policy used to propose candidates), then $g_T \neq g_O$ and we must also reweight by g_T/g_O. Concretely, this is where propensity logging becomes a hard requirement: for each logged comparison we need the probability with which the specific ordered pair (y, y') was produced under the logging mechanism. Without these propensities, the ratio g_T/g_O is not identifiable, and the reweighting argument breaks.

What exactly is identified? With observed C, we can identify target expectations of any function of (X, C, Y, Y', L), including:

$$\mathbb{P}_T(L = 1 \mid x, y, y', c), \qquad \mathbb{E}_T[L \mid x, y, y', c],$$

and, by aggregation,

$$\mathbb{P}_T(L = 1 \mid x, y, y') = \sum_c p_T(c \mid x)\, \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big).$$

Note the last expression is a *mixture of sigmoids.* This is not merely a technicality: it means that even if we ultimately care about a single shared policy, the target preference distribution depends on the target conditional mix $p_T(c \mid x)$ and cannot in general be represented as $\sigma(\bar{r}(x,y) - \bar{r}(x,y'))$ for a single scalar reward $\bar{r}$ unless we introduce additional restrictions. Identification, however, is about recovering the target comparisons we need for training and evaluation, not about recovering a globally consistent scalar reward.

When C is *not* observed: what breaks, and what can still be done. If C is latent, then the reweighting identity above cannot be applied as written because the weight depends on c and the label mechanism depends on c. In particular, observing only (x, y, y', ℓ) identifies

$$\mathbb{P}_O(L = 1 \mid x, y, y') = \sum_c p_O(c \mid x)\, \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big),$$

but the target quantity replaces $p_O(c \mid x)$ with $p_T(c \mid x)$. Without additional information, multiple latent decompositions can produce the same observational mixture while implying different target mixtures; this is a genuine non-identification result driven by unobserved effect modification.

There are, however, three practically relevant paths to recover identification.

(1) Measure a sufficient proxy $\hat{C}$ (or richer metadata Z). If we can log user metadata Z such that the heterogeneity relevant to preferences is captured by Z, then we can replace C by Z in the weighting scheme. Formally, we need a condition of the form

$$\mathbb{P}(L = 1 \mid x, y, y', c) = \mathbb{P}(L = 1 \mid x, y, y', z) \quad \text{whenever } z \text{ is generated from } c,$$

or, more weakly, that the conditional distribution of potential outcomes is independent of the assignment given (X, Z) and that Z suffices for transporting preferences from O to T. This is the deployment-motivated reason to treat segmentation variables (locale, product surface, account age, safety setting, domain tags, etc.) as first-class citizens in the training logs: they are not only useful features, but also *identification variables*.

(2) Impose a "no effect modification" restriction. If we are willing to assume that type only affects prompt frequency but not the preference comparison itself, i.e.

$$r^*(x, y, c) - r^*(x, y', c) = \Delta r^*(x; y, y') \quad \text{for all } c,$$

then C drops out of the Bradley–Terry probability and identification becomes possible without observing C. This assumption is often false in the motivating safety cases (different user objectives genuinely disagree), but it can be a reasonable approximation within carefully controlled slices (e.g., within a single product surface with stable norms).

(3) Model C and accept stronger assumptions (and corresponding fragility). One can posit a parametric latent-variable model for $p(c \mid x)$ and $r^*(x, y, c)$, estimate it from observational data (possibly using repeated measurements, instruments, or multi-environment variation), and then transport to p_T. This can work, but it moves the burden from propensity logging to model identifiability, and it introduces a new failure mode: if the latent model is misspecified, the transported preferences can be systematically wrong in precisely the regions we most care about (rare, high-stakes prompts).

Safety and governance interpretation. Identification clarifies what the platform must commit to *before* optimization. If we want counterfactual guarantees about welfare under a specified p_T and g_T, then (i) p_T must be explicitly defined, (ii) overlap must be engineered via data collection (often via randomization or deliberate coverage), and (iii) propensities and sufficient heterogeneity variables must be logged. Otherwise, training reduces to extrapolation: we may still produce a model, but we cannot defend claims about its behavior under the target regime using the logs alone.

6 Estimation: a causal DPO objective from logged comparisons

Given the change-of-measure identity in Section 5, the estimator design problem becomes concrete: we want an empirical objective whose *population* minimizer coincides with the target-regime KL-regularized optimum, while remaining implementable with logged comparisons.

From pairwise labels to DPO training tuples. Each logged example consists of $(x_i, c_i, y_i, y_i', \ell_i)$, where $\ell_i \in \{0, 1\}$ indicates whether the labeler preferred y_i to y_i'. We rewrite this as a *winner/loser* pair $(y_{w,i}, y_{\ell,i})$ by setting

$$(y_{w,i}, y_{\ell,i}) := \begin{cases} (y_i, y_i') & \text{if } \ell_i = 1, \\ (y_i', y_i) & \text{if } \ell_i = 0. \end{cases}$$

This bookkeeping step matters operationally because most DPO implementations assume an ordered (preferred, dispreferred) pair and optimize a single logistic term per comparison.

Importance-weighted DPO loss. Fix a reference policy π_{ref} (as in standard DPO) and choose a temperature/regularization parameter $\beta > 0$. For a candidate policy π_θ, define the usual DPO logit difference

$$\Delta_\theta(x; y_w, y_\ell) := \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_\ell \mid x)}{\pi_{\text{ref}}(y_\ell \mid x)}.$$

The *causal* modification is to weight each comparison by the importance ratio mapping the observational regime to the target regime. Concretely, for observed c (or a sufficient proxy), we use

$$w(x, c, y_w, y_\ell) := \frac{p_T(x, c)}{p_O(x, c)} \cdot \frac{g_T(y_w, y_\ell \mid x, c)}{g_O(y_w, y_\ell \mid x, c)}.$$

The empirical objective is then

$$\widehat{\mathcal{L}}_{\mathrm{wDPO}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \hat{w}_i \cdot \Big(-\log \sigma\big(\beta\, \Delta_\theta(x_i; y_{w,i}, y_{\ell,i})\big)\Big), \qquad \hat{w}_i \approx w(x_i, c_i, y_{w,i}, y_{\ell,i}).$$

Operationally, this is a minimal change to existing DPO code: we multiply each per-example loss by $\hat{w}_i$ (or equivalently, resample examples proportional to $\hat{w}_i$). The conceptual change is larger: without $\hat{w}_i$, we fit the observational preference distribution; with $\hat{w}_i$, we fit the *target* preference distribution implied by the counterfactual deployment regime.

Practical estimation and stabilization of weights. In the cleanest deployments, w is known by design: the platform randomizes prompts or reweights traffic, and the pair assignment mechanism g_O is instrumented to return exact propensities. In most realistic settings, parts of w must be estimated, and this introduces two familiar failure modes: (i) *variance blow-up* from heavy-tailed weights (weak overlap), and (ii) *bias* from misspecified propensity models.

We therefore treat weight stabilization as part of the estimator specification. Common choices include:

- **Normalization:** use self-normalized weights $\tilde{w}_i = \hat{w}_i / \big(\frac{1}{n} \sum_j \hat{w}_j\big)$, which preserves the target objective asymptotically but can reduce numerical scale issues.

- **Clipping/truncation:** replace $\hat{w}_i$ by $\min\{\hat{w}_i, W_{\max}\}$ for a chosen cap $W_{\max}$. This introduces bias but can be an explicit safety–robustness trade: we refuse to let rare, poorly-supported regions dominate gradients.

- **Stratification:** define p_T and the weighting scheme at the level of slices (product surface, locale, safety setting, etc.) rather than at the granularity of individual prompts, thereby trading correction fidelity for overlap.

From a safety perspective, it is often preferable to make these compromises explicit (and auditable) rather than implicitly relying on whatever prompt mix happened to be logged.

Optional doubly-robust (DR) augmentation. Importance weighting is unbiased under correct propensities, but it can be statistically brittle when $\mathbb{E}[w^2]$ is large or when g_O is only approximately known. A standard remedy in causal policy learning is to combine weights with an outcome model, yielding a *doubly-robust* objective: consistency holds if either the propensity model or the outcome model is correct.

In our setting, the "outcome" is the pairwise preference probability. Let

$$q^*(x, c, y, y') := \mathbb{P}(L = 1 \mid x, c, y, y')$$

and fit a model $\hat{q}(x, c, y, y')$ (e.g., a calibrated classifier over concatenated (x, y, y'), or a smaller preference model trained with cross-fitting). A DR-style causal DPO objective can be written schematically as

$$\widehat{\mathcal{L}}_{\mathrm{DR}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\Big[\hat{w}_i \cdot \ell_{\mathrm{DPO}}(\theta; x_i, y_{w,i}, y_{\ell,i}) + \big(\hat{\kappa}_i\big) \cdot \phi\big(\hat{q}, \theta; x_i, c_i, y_i, y_i'\big)\Big],$$

where $\ell_{\mathrm{DPO}}(\cdot) = -\log\sigma(\beta\Delta_\theta(\cdot))$, ϕ is an influence-function-like correction term, and $\hat{\kappa}_i$ is constructed so that the second term has mean zero when $\hat{w}$ is correct and cancels first-order errors when it is not. We do not need the exact algebra of ϕ to implement the core idea: we are adding a correction that uses $\hat{q}$ to reduce sensitivity to propensity error and to reduce variance in high-weight regions. In practice, DR objectives typically require careful cross-fitting (separate folds for fitting $\hat{w}, \hat{q}$ versus optimizing θ) to avoid overfitting-induced bias.

What must be logged (non-negotiables). The estimator is only as credible as the logging. To make the weighting argument operational, we need logs that allow us to reconstruct (or consistently estimate) every factor in w on the support of p_T. At minimum, we need:

- **The comparison itself:** x, both full responses y, y', and the preference label L (plus tie/abstain metadata if applicable).
- **Heterogeneity variables:** the user type c when available, or a proxy/metadata vector z that we are willing to treat as sufficient for transporting preferences (and that is available both in training logs and in the target definition).
- **Pair assignment propensities:** the probability under the logging mechanism of producing the *ordered* pair (y, y') given (x, c) (or (x, z)). This generally requires instrumenting the candidate generation pipeline, including any mixture over proposal policies, any filtering, and any stochastic decoding parameters.

- **Target-definition bookkeeping:** enough information to compute $p_T(x, c)/p_O(x, c)$ for the target regime we claim to optimize for (e.g., randomization probabilities, slice weights, or an explicit reweighting plan).

A recurring deployment pitfall is to log only which *policy* generated a response (or only the decoding parameters) without logging the *resulting probability of the realized sequence.* If we cannot compute (or bound) $g_O(y, y' \mid x, c)$, then the causal correction becomes an untestable modeling assumption rather than an identified estimator.

Implementation notes for modern training stacks. Weighted DPO integrates cleanly with minibatch SGD: each example carries a scalar multiplier $\hat{w}_i$, and gradients scale linearly. Two additional engineering details matter in practice. First, we must ensure $\pi_{\text{ref}}(y \mid x) > 0$ for all realized sequences, which is typically satisfied by using a dense neural LM with the same tokenizer/vocabulary and avoiding hard truncation rules that can assign exactly zero probability. Second, for numerical stability, we typically compute $\log \pi_\theta(y \mid x)$ as the sum of token log-probabilities under teacher forcing, matching standard DPO implementations.

Finally, we emphasize a governance-relevant point: the target regime (p_T, g_T) is part of the training specification, not an afterthought. Once weights are introduced, changing the target definition changes the estimator. This is precisely the intended behavior—it forces us to make explicit which counterfactual population and which generation mechanism our optimization is claiming to serve.

7 Theory: optimality, consistency, and finite-sample behavior

This section states the main theoretical guarantees that justify the estimator in Section 6. The through-line is that DPO is best understood as (regularized) maximum likelihood for a Bradley–Terry–Luce (BTL) comparison model after a particular reparameterization; once we explicitly change measure from the observational regime to the target regime, the familiar optimality and learning-theoretic conclusions go through, with the expected importance-weight penalties.

(i) Population optimality and the exponential-tilt form. Recall the target welfare objective

$$J(\pi) = \mathbb{E}_{(x,c)\sim p_T} \mathbb{E}_{y\sim\pi(\cdot|x)}[r^*(x, y, c)] - \beta\, \mathbb{E}_{(x,c)\sim p_T} \mathrm{KL}\big(\pi(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\big).$$

Under mild regularity (measurability and $\pi_{\text{ref}}(y \mid x) > 0$ on the relevant support), the pointwise optimization over $\pi(\cdot \mid x)$ yields the standard Gibbs/exponential-tilting characterization: for p_T-almost every x,

$$\pi^*(y \mid x) \propto \pi_{\text{ref}}(y \mid x) \exp\Big(\tfrac{1}{\beta}\, \bar{r}^*(x, y)\Big)\,, \qquad \bar{r}^*(x, y) := \mathbb{E}_{c \sim p_T(c \mid x)}[r^*(x, y, c)].$$

Two details matter for interpretation. First, only the *type-averaged* reward $\bar{r}^*$ appears when deploying a single shared policy: the platform cannot condition on c at generation time unless it explicitly builds a conditional policy $\pi(y \mid x, c)$. Second, β is not merely an optimization temperature; it is the explicit knob controlling how strongly we are willing to deviate from π_{ref} in pursuit of higher reward, and thus how much we are willing to amplify any estimation error in $\bar{r}^*$. From a safety standpoint, this is the core tradeoff revealed by the formalism: larger β is a commitment to conservatism, which can be desirable under distribution shift, logging gaps, or preference-model misspecification.

DPO as likelihood matching under the target regime. The BTL assumption posits

$$\mathbb{P}_T(L = 1 \mid x, c, y, y') = \sigma\big(r^*(x, y, c) - r^*(x, y', c)\big).$$

DPO replaces the unknown reward difference with a log-density ratio parameterization relative to π_{ref}, effectively fitting a logistic model whose "score" is $\beta\, \Delta_\theta(x; y_w, y_\ell)$. In population, minimizing the (target) logistic loss corresponds to matching the target comparison probabilities, and the induced optimal policy coincides (up to the standard additive-in-x, c reward equivalence class) with the maximizer of $J(\pi)$. The importance-weighted objective from Section 6 is exactly the device that converts an observational likelihood into a target likelihood.

(ii) Consistency of weighted DPO as an M-estimator. Let $\mathcal{L}_{\text{wDPO}}(\theta)$ denote the population weighted loss under the observational sampling measure,

$$\mathcal{L}_{\text{wDPO}}(\theta) = \mathbb{E}_O\big[w(X, C, Y_w, Y_\ell) \cdot \big(-\log \sigma(\beta\, \Delta_\theta(X; Y_w, Y_\ell))\big)\big]\,.$$

By the change-of-measure identity, $\mathcal{L}_{\text{wDPO}}(\theta)$ can be rewritten as an expectation under the *target* joint distribution over (X, C, Y_w, Y_ℓ) induced by (p_T, g_T). Consequently, if the model class is well specified (there exists θ^* such that $\pi_{\theta^*} = \pi^*$ almost everywhere) and standard empirical-process conditions hold, then the empirical minimizer $\hat{\theta} \in \arg\min_\theta \widehat{\mathcal{L}}_{\text{wDPO}}(\theta)$ is a consistent M-estimator:

$$\pi_{\hat{\theta}} \xrightarrow[n \to \infty]{\mathbb{P}} \pi^*,$$

with convergence understood in the usual sense induced by the loss (e.g., convergence of $\Delta_{\hat{\theta}}$ in $L_2(p_T)$, which is sufficient to identify the policy up to null sets).

The assumptions doing real work are exactly the causal ones we would expect: (a) overlap controls the weight magnitude; (b) correct propensities ensure the weights implement the intended target regime; and (c) sufficient heterogeneity features (true c or a valid proxy z) ensure that transporting from p_O to p_T does not silently change the label distribution in ways not accounted for by w. When these assumptions fail, the objective can still be optimized, but it no longer has a causal interpretation.

Consistency under optional doubly-robust augmentation. Weighted objectives are statistically fragile when $\mathbb{E}[w^2]$ is large or when propensities are estimated with error. A doubly-robust (DR) construction adds an explicit model $\hat{q}(x, c, y, y') \approx \mathbb{P}(L = 1 \mid x, c, y, y')$ and a correction term chosen to yield Neyman-orthogonality: first-order errors in $\hat{w}$ do not translate into first-order bias in the estimating equation. Under cross-fitting and mild regularity, DR consistency holds if either (i) the propensity model is consistent or (ii) the outcome model is consistent. This is not a free lunch—DR can raise implementation complexity and introduce its own failure modes (calibration errors in $\hat{q}$, leakage without proper sample splitting)—but it offers a principled "second chance" when exact logging is infeasible.

(iii) Finite-sample excess-risk and the role of overlap and β. In finite samples, the object we can control is the *excess weighted logistic risk* relative to the best-in-class parameter:

$$\mathcal{L}_{\mathrm{wDPO}}(\pi_{\hat{\theta}}) - \mathcal{L}_{\mathrm{wDPO}}(\pi^*).$$

Under bounded weights $w \leq W$ (or after explicit clipping) and a standard complexity control on the score class $\{\log \pi_\theta(\cdot \mid x)\}$ (e.g., via Rademacher complexity $\mathfrak{R}_n$), one obtains high-probability bounds of the form

$$\mathcal{L}_{\mathrm{wDPO}}(\pi_{\hat{\theta}}) - \mathcal{L}_{\mathrm{wDPO}}(\pi^*) \;\lesssim\; W\,\mathfrak{R}_n + W\sqrt{\tfrac{\log(1/\delta)}{n}}.$$

The dependence on overlap enters through W: if g_O or $p_O(c \mid x)$ can be arbitrarily small on the target support, then w becomes heavy-tailed and no meaningful finite-sample guarantee is available without clipping. This is not an artifact of the analysis; it is the classical statistical price of counterfactual evaluation and off-policy learning.

The parameter β enters in two coupled ways. First, it changes the sensitivity of the loss: the logistic term $-\log \sigma(\beta\,\Delta)$ becomes steeper as β increases in the logit scale, affecting Lipschitz constants and thus the prefactors in generalization bounds. Second (and more importantly for deployment), β governs the curvature of the welfare objective: smaller β permits sharper tilting away from π_{ref}, which can convert modest estimation error in Δ_θ or in w into large changes in the learned policy. In regimes with weak overlap or noisy propensities, larger β can therefore be interpreted as a robustness parameter: we accept slower improvement in $J(\pi)$ in exchange for reduced amplification of statistical and causal uncertainty.

Limitations and open problems. These guarantees are conditional on the causal transport assumptions: if c is latent and the proxy z is insufficient, then even perfectly computed weights need not identify the target preference distribution. Moreover, the analysis treats p_T as fixed and exogenous, while real platforms may induce feedback between deployed policies, prompt choice, and user composition. Extending causal DPO to such strategic or dynamical regimes (while keeping logging requirements auditable) remains an open and governance-relevant problem: without this, "optimizing for the target" may itself change the target.

8 Empirics: controlled confounding and telemetry-style simulation

We empirically stress-test the causal claims in two complementary regimes. First, we construct a *controlled confounding* benchmark on top of publicly-available helpful/harmless (HH) preference data, where we can dial the strength of X–C dependence and still retain a notion of ground-truth target welfare. Second, we run a *product-telemetry simulation* meant to mirror what a platform can realistically do with logged propensities: evaluate and train under one logging regime while caring about a counterfactual target regime that reweights the user/prompt mix and (optionally) the response-pair assignment.

Controlled confounding via HH augmentation. The key difficulty in evaluating causal DPO methods is that in real datasets the user type C is rarely observed and the latent reward r^* is unknown. To isolate the causal mechanism without losing contact with realistic language-model artifacts, we start from a dataset of prompts x paired with two candidate responses (y, y') and a preference label ℓ. We then introduce a *synthetic type* $c \in \{0, 1\}$ and a type-conditioned reward $r^*(x, y, c)$ by augmenting the base signal with a type-specific component derived from HH attributes. Concretely, we build two scalar scores for each completion, one intended to proxy "helpfulness" and another intended to proxy "harmlessness" (e.g., via an auxiliary classifier or rubric-based model). We define

$$r^*(x, y, c) = s_{\text{base}}(x, y) + \alpha_c \, s_{\text{HH}}(x, y),$$

where s_{HH} emphasizes helpfulness for $c = 0$ and harmlessness for $c = 1$, and α_c controls how strongly types disagree. Labels are then (re)sampled from the BTL model $\ell \sim \mathrm{Bernoulli}(\sigma(r^*(x, y, c) - r^*(x, y', c)))$. This construction gives us a known, controllable data-generating process while preserving realistic response distributions and prompt content.

To create confounding, we do *endogenous prompt selection*: prompts are partitioned into coarse categories (e.g., "benign assistance," "policy-sensitive," "self-harm," "medical") and we sample $x \sim p_O(x \mid c)$ with type-specific mixtures. By increasing the separation between $p_O(x \mid c = 0)$ and $p_O(x \mid c = 1)$ we increase confounding strength (e.g., measured by $I(C; X)$). The target regime $p_T(x, c)$ then either (i) randomizes prompts within type, or (ii) enforces a product distribution $p_T(c)p_T(x)$ meant to represent a fairness or coverage desideratum. Because r^* is known in this benchmark, we can evaluate the learned policy under the *true* target welfare

$$J(\pi) = \mathbb{E}_{(x,c)\sim p_T}\mathbb{E}_{y\sim\pi(\cdot|x)}[r^*(x, y, c)] - \beta\, \mathbb{E}_{(x,c)\sim p_T}\mathrm{KL}(\pi\|\pi_{\mathrm{ref}}),$$

using Monte Carlo rollouts from $\pi(\cdot \mid x)$ together with the synthetic reward.

Across confounding strengths, we compare (a) naive/unweighted DPO fit on (x, y, y', ℓ), (b) importance-weighted DPO using the true $w(x, y, y', c)$, and (c) optional doubly-robust variants when we intentionally corrupt propensities (described below). The qualitative pattern is stable: naive DPO reliably tracks the observational objective and can systematically mis-rank responses under the target mixture when type disagreement is strong; weighted DPO largely eliminates this bias when overlap holds, recovering policies whose target welfare is close to the oracle optimum within function-class limits. As expected from the theory, the welfare gap between naive and weighted methods grows with both $I(C; X)$ and the magnitude of α_c, i.e., when (i) prompts are more type-segregated and (ii) types disagree more about what constitutes a "good" completion.

Telemetry-style simulation with logged propensities. To mirror deployment, we next consider a setting where prompts and metadata arrive from a logging system, and the platform controls (and logs) the response-pair assignment mechanism $g_O(y, y' \mid x, \hat{c})$. We simulate a production workflow in which the logged dataset is collected under one prompt composition and one pairing strategy, while the platform wishes to optimize under a different target regime (e.g., reweighting toward safety-critical prompts, or representing an anticipated user-mix shift). In this simulation, C may be partially observed (metadata $\hat{c}$) and used for weighting, but the policy itself remains unconditional at generation time.

Operationally, we implement target reweighting by specifying $p_T(x, \hat{c})$ (often via stratified resampling or post-stratification on metadata buckets) and optionally specifying a counterfactual g_T that differs from g_O (e.g., if the evaluation regime compares π_θ to a different baseline than the logging regime did). We then train with weights

$$w(x, y, y', \hat{c}) = \frac{p_T(x, \hat{c})}{p_O(x, \hat{c})} \cdot \frac{g_T(y, y' \mid x, \hat{c})}{g_O(y, y' \mid x, \hat{c})},$$

using the logged propensities for g_O. Because true r^* is not available in this telemetry simulation, we evaluate with a held-out preference model (or a held-out set of human labels where available), reporting both target-weighted preference accuracy and proxy welfare estimates. The main empirical question here is not whether we can perfectly recover a true reward, but whether causal reweighting prevents systematic errors that arise from training on the wrong mixture.

Ablations: overlap and effective sample size. We explicitly ablate overlap by modifying g_O and/or $p_O(x, \hat{c})$ so that some regions of the target support are rarely logged. Practically, we do this by (i) narrowing the response-pair generator so that certain styles or lengths are under-sampled, and (ii) increasing the mismatch between p_O and p_T so that some $(x, \hat{c})$ cells receive little mass under logging. We track not only raw weight maxima but also the *effective sample size* $n_{\text{eff}} = (\sum_i w_i)^2 / (\sum_i w_i^2)$, which is a simple diagnostic for how brittle importance weighting becomes. The results match the predicted failure mode: as overlap weakens, variance dominates, n_{eff} collapses, and performance becomes sensitive to clipping. Weight clipping improves stability but introduces bias, yielding the expected bias–variance frontier rather than a dominated solution. This is also the regime where larger β empirically behaves as a robustness knob: policies remain closer to π_{ref} and degrade more gracefully when weights are heavy-tailed.

Ablations: propensity misspecification. We next study misspecified propensities by perturbing g_O or p_O before computing weights, e.g., multiplicative noise on w, mis-bucketing metadata, or omitting relevant covariates from the propensity model. As soon as the weight model fails to condition on variables that jointly affect $(X, \hat{C})$ and the label distribution, the weighted estimator inherits bias that can be comparable to (or worse than) naive DPO, especially when the perturbations are systematic rather than mean-zero. In these cases, DR-style augmentation partially mitigates error when the outcome model is reasonably calibrated, but the gains are not automatic: poor calibration in $\hat{q}(x, \hat{c}, y, y')$ can introduce its own bias, emphasizing the need for careful validation and cross-fitting when DR is used.

Ablations: proxy quality for latent type. Finally, we test how sensitive causal transport is to the quality of the type proxy. Starting from a "best available" metadata proxy $\hat{c}$, we progressively coarsen it (merge buckets), inject noise, or drop it entirely. Predictably, when $\hat{c}$ becomes uninformative, weighting reduces to reweighting on x alone and cannot correct type-driven preference shifts; the learned policy then reverts toward the observational optimum in exactly those slices where types disagree. This ablation is the sharpest empirical reminder of the core identification constraint: if the relevant drivers of preference heterogeneity are latent and not captured by logged covariates, no amount of reweighting can recover the target label distribution, and "causal DPO" becomes an aspirational interpretation rather than a guarantee.

Taken together, these empirical exercises aim less at producing a single headline number and more at mapping the *operational* boundary of the theory: causal DPO works well when the platform can (i) log or estimate propensities accurately, (ii) maintain overlap with the intended target support, and (iii) measure the preference-relevant heterogeneity needed for transport. When any of these pillars fails, the method still trains a model—but it ceases to be a credible answer to the counterfactual question the platform actually cares about.

9 Policy and practice implications: from causal DPO to deployable procedures

Our formalism is deliberately "training-loop native": it says that if we want welfare under a *counterfactual* prompt–type regime $p_T(x, c)$ (and possibly a counterfactual pairing regime g_T), then we should treat propensities and overlap as first-class objects rather than as incidental logging details. The practical implication is that a platform cannot credibly claim "we optimized for safety/helpfulness under population T" unless it can (i) specify T concretely, and (ii) show that the collected data supports transport from the observational regime O to T without uncontrolled variance or hidden heterogeneity. Below we translate this into audit checklists, sensitivity analyses, and governance hooks that deployment teams and regulators can actually operationalize.

Audit checklist for data collection and logging. Causal DPO is only as good as the platform's ability to compute (or approximate) the Radon–Nikodym derivative w. This induces a minimal "data-sheet" for preference logs that goes beyond storing (x, y, y', ℓ). At a minimum, we want:

- **Propensities for response-pair assignment.** Log $g_O(y, y' \mid x, \hat{c})$ (or a sufficient description to reconstruct it exactly) for each tuple, including any temperature, rejection sampling, filters, or mixture components used in pair generation. If the system used multiple generators, log the mixture identity and mixture probability per sample.

- **Prompt selection provenance.** Record enough metadata to estimate $p_O(x, \hat{c})$ and justify a transport map to $p_T(x, \hat{c})$. In product systems this typically means: surface/channel (search, chat, agent), locale, policy-mode flags, and safety-routing decisions that affect which prompts enter labeling.
- **Type-relevant covariates.** Since true C is usually latent, we need a proxy $\hat{c}$ that captures preference-relevant heterogeneity (e.g., user intent class, risk tier, domain bucket). The audit question is not "is $\hat{c}$ predictive of C?" but "does conditioning on $(x, \hat{c})$ plausibly block the major pathways by which preferences vary?"
- **Support and overlap diagnostics at collection time.** Ensure that for every $(x, \hat{c})$ cell in the intended target support, the logging pipeline produces non-negligible mass. Concretely, we recommend tracking per-cell counts and an *online* estimate of effective sample size $n_{\text{eff}} = (\sum_i w_i)^2 / (\sum_i w_i^2)$ under the current candidate target weights.

These requirements are not merely bureaucratic. They correspond exactly to the assumptions used to justify reweighting: without logged propensities, the estimator is not identifiable; without type-relevant covariates, the transport claim is not even well-posed.

Design guidance: prefer "cheap randomization" to heroic reweighting. When teams have any ability to intervene in data collection, the highest-leverage move is to reduce the variance of w by design. Two robust patterns are (i) *stratified collection* over $(x, \hat{c})$ cells that are important for p_T, and (ii) *pairing randomization* within each stratum so that g_O is simple and bounded away from zero. In our setting this often looks like: allocate labeling budget to a pre-specified mixture of prompt buckets, and within each bucket sample response pairs using a known mixture of policies with explicit mixture weights. The technical point is that importance weighting is a last resort: it corrects bias but amplifies noise. From a safety standpoint, reducing the need for large weights is equivalent to reducing the chance that a small number of rare, high-weight samples dominate the gradient.

Sensitivity analysis when C is latent. Most deployments will not satisfy the idealized setting where C is observed. We therefore recommend treating causal DPO claims as graded rather than binary: the question becomes how sensitive the learned policy and estimated welfare are to *plausible* violations of the conditional unconfoundedness assumption when conditioning only on $(X, \hat{C})$. Practically, we can do three complementary analyses.

- **Proxy stress tests.** Train and evaluate under progressively coarsened versions of $\hat{c}$ (merging buckets, dropping fields), and check whether the implied target-welfare ranking of policies is stable. Instability is evidence that unmeasured heterogeneity is driving the result.

- **Weight-robustness frontiers.** Report performance as a function of weight clipping thresholds and of β. Since clipping trades variance for bias and β controls how aggressively we move away from π_{ref}, plotting welfare (or preference accuracy) versus these knobs gives a concrete robustness curve rather than a single fragile point estimate.

- **Latent-confounding bounds.** Introduce an explicit sensitivity parameter Γ that upper-bounds how much an unobserved binary confounder could tilt the odds of $\ell = 1$ within a $(x, \hat{c}, y, y')$ cell (an analogue of Rosenbaum-style bounds). Even if the bound is coarse, it forces teams to state what magnitude of hidden heterogeneity would overturn their conclusion about which policy is preferred under p_T.

The meta-principle is that when C is latent, we should avoid treating reweighting as a magical deconfounder. Instead, we should publish *robustness envelopes*: ranges of conclusions under a family of plausible confounding models.

Actionable guidance for regulators and internal governance. A regulator (or an internal model-risk committee) should not need to inspect gradient code to evaluate whether a "causal alignment" claim is meaningful. We propose a lightweight set of artifacts that make the counterfactual objective auditable:

- **A declared target regime.** Document $p_T(x, \hat{c})$ and its motivation (e.g., anticipated user-mix shift, safety-critical over-sampling, fairness constraints). If p_T changes over time, version it.

- **A propensity and overlap report.** Provide summary statistics of w: quantiles, $\max w$, $\mathbb{E}[w]$, $\mathbb{E}[w^2]$, and n_{eff} for the chosen p_T. These are direct proxies for estimator variance and for the credibility of transport.

- **A counterfactual evaluation protocol.** Pre-specify how target-weighted evaluation is computed (including any clipping), which slices are monitored (e.g., safety-critical domains), and what constitutes a deployment-blocking regression.

- **Change management hooks.** Require that changes to logging, filters, or pair generation that affect g_O trigger a re-computation of propensities and overlap diagnostics. Silent changes to g_O are, in our model, silent changes to the estimand.

This is also where we see a concrete safety tradeoff: stricter governance around p_T, g_O, and overlap can slow iteration, but it sharply reduces the probability that a system is optimized for an unintended subpopulation due to confounded telemetry.

Deployment-team heuristics: when to trust the method and when to fall back. Finally, we want a simple operational decision rule. In our experience, causal reweighting is most trustworthy when (i) n_{eff} is not dramatically smaller than n under the chosen p_T, (ii) the top-weight mass is not dominated by a tiny number of prompts, and (iii) conclusions are stable across reasonable clipping/β choices. Conversely, if overlap is weak or if $\hat{c}$ is demonstrably uninformative, the safest posture is to treat weighted DPO as a *diagnostic* rather than as a guarantee: it can highlight where the logging regime is misaligned with the intended deployment regime, but it should not be the sole basis for high-stakes claims about counterfactual safety performance.

10 Limitations and extensions: unobserved confounding, strategic feedback, and a path to active overlap design

Our causal-DPO framing makes explicit what must be true for reweighting to transport preference data from the logged regime to a target regime. That explicitness is also where the main limitations become visible: the assumptions that make w meaningful (and finite) are strong, and the training loop can itself create new dependencies that break them. In this section we sketch the main failure modes and a research roadmap that, in our view, moves the method from a "correctness on paper" statement toward deployable alignment procedures.

Unobserved confounding is not a small technicality. The central vulnerability is that the preference model depends on a latent objective C, but in most deployments we only condition on proxies $\hat{C}$ (or on prompt metadata) rather than on the true C. If there remains a variable U that both (i) affects which prompts or which response pairs are labeled and (ii) affects label outcomes beyond what is explained by $(X, \hat{C})$, then importance weighting can be directionally misleading: we are correcting the wrong selection mechanism. This is not merely "variance inflation"; it is estimand drift. In particular, even if we can compute $\frac{p_T(x,\hat{c})}{p_O(x,\hat{c})}$ exactly, the transported quantity is $\mathbb{E}[r^*(X,Y,C) \mid X, \hat{C}]$ averaged under p_T, which need not coincide with $\mathbb{E}[r^*(X,Y,C) \mid X, C]$ averaged under p_T. The resulting policy can optimize for an unintended mixture of user goals—a safety-relevant failure when rare but high-risk objectives are systematically under-captured by $\hat{C}$.

Methodologically, this pushes us toward *partial identification* rather than point identification. One concrete direction is to integrate sensitivity models (e.g., odds-ratio bounded unobserved confounding with parameter Γ) directly into training: instead of producing a single $\hat{\pi}$, we would produce a family $\{\hat{\pi}_\Gamma\}$ or a worst-case robust policy $\hat{\pi}_{\text{rob}}$ that maximizes a lower bound on $J(\pi)$ over all confounding structures consistent with the bound. The open problem is to do this while preserving the attractive computational properties of DPO-style objectives, and without producing a policy that is so conservative that it simply collapses back to π_{ref}.

Latent Z discovery: learning the right conditioning set. A natural extension is to replace the ad hoc proxy $\hat{C}$ with a learned latent variable Z that captures preference-relevant heterogeneity (intent, risk tolerance, domain norms) in a way that is stable across collection channels. Conceptually, we want Z such that conditioning on (X, Z) renders the labeling process approximately unconfounded and the reward model approximately invariant across regimes. Practically, this becomes a joint representation-learning and policy-learning problem: we infer Z from available telemetry and text, and train using weights $w(x, z, \cdot)$.

This direction is promising but brittle. First, Z can absorb spurious correlates of labeling artifacts (annotator idiosyncrasies, UI differences) rather than underlying user objectives. Second, the objective "make preferences predictable" is not the same as "block backdoor paths"; naive representation learning can increase confounding by creating a collider-like summary. Third, personalization pressure can cause Z to encode sensitive attributes. For safety and governance, Z-discovery needs constraints: invariance tests across collection pipelines, privacy-preserving training, and explicit auditing of which features Z uses. A constructive near-term target is *multi-environment* logging (different surfaces, different prompt routers) and learning a Z that equalizes preference prediction error across environments; this is not a proof of causal sufficiency, but it is a measurable step toward transportability.

Strategic labeling and preference manipulation. Our model treats the labeler as sampling L from a Bradley–Terry distribution with latent reward differences. Real labeling pipelines violate this in at least three ways. (i) *Strategic annotators*: when incentives, fatigue, or policy pressure exist, the label distribution can shift in response to the platform's current model, rubric, or measurement regime. (ii) *Model-assisted evaluation*: when an evaluator model is used as a proxy labeler, its errors can be systematically exploited by the policy being trained, creating a Goodhart loop. (iii) *Adversarial inputs*: some users (or external actors) may craft prompts to elicit outputs that are likely to be labeled as "good" under the rubric but are unsafe in deployment contexts.

These phenomena suggest two extensions. First, we should treat labeling as a mechanism with its own causal graph and potential strategic behavior, not as exogenous noise. A minimal step is to add annotator identity and context as conditioning variables and to monitor nonstationarity of $\mathbb{P}(L = 1 \mid x, y, y', \cdot)$ over time. Second, we should incorporate robust aggregation and adversarial evaluation into the target objective: instead of a single r^*, we may need a set of plausible reward functions (capturing rubric ambiguity and evaluator disagreement) and optimize for a conservative criterion (e.g., quantile welfare or worst-case over reward sets) on safety-critical slices. This connects directly to verification: if the target is "never produce disallowed content," then preference optimization should be constrained by a separately audited policy-compliance classifier, rather than relying on preferences to implicitly enforce constraints.

Adaptive data collection breaks i.i.d. and changes the estimand unless we model it. Modern training loops are adaptive: we deploy a candidate policy, collect new prompts conditioned on that policy's behavior, choose which examples to label, and update again. This adaptivity changes both p_O and g_O in a history-dependent way; naive importance weighting that treats logged data as i.i.d. from a fixed p_O can be invalid. The right abstraction is sequential decision-making: data collection is a policy over queries, and preference labels are outcomes. In that setting, the relevant tools look more like off-policy evaluation and doubly-robust estimation in contextual bandits than like one-shot covariate shift.

From an alignment perspective, the key issue is *exploration for overlap.* If we only label what the current system is likely to produce, then the support of g_O shrinks around current behavior and the effective ϵ deteriorates over time. This creates a failure mode where the training loop becomes increasingly confident while becoming less identified. A safety-aligned adaptive pipeline should therefore include explicit exploration constraints: guarantees that each critical $(x, \hat{c})$ stratum receives labeling mass and that each response-generation mode remains sampled with nontrivial probability, even when short-term metrics suggest otherwise.

Roadmap: from passive reweighting to active overlap design and personalized alignment. The long-run goal is to make overlap a *design parameter* rather than a post hoc diagnostic. Concretely, we can view the platform as choosing the logging design (p_O, g_O) under a labeling budget constraint to minimize the expected generalization error for the target regime. This suggests optimization problems of the form: allocate labeling across strata to minimize $\mathbb{E}[w^2]$ subject to covering safety-critical regions, or to maximize n_{eff} subject to fairness and privacy constraints. Technically, this is a form of experimental design for preference learning with KL-regularized policy updates.

Finally, we anticipate growing pressure toward *personalized alignment*: learning $\pi(y \mid x, c)$ (or $\pi(y \mid x, z)$) rather than a single shared policy optimized for $\bar{r}^*$. Personalization can reduce conflicting gradients across types, but it introduces new attack surfaces (users gaming c), new governance questions (what types are permitted, how they are inferred), and new safety constraints (ensuring that personalization cannot relax safety policies). A principled path is to treat personalization as *bounded*: optimize within a family of type-conditional policies while enforcing global constraints and monitoring for distributional shifts in inferred c. The open problem is to combine such constraints with causal transport and adaptive collection in a way that remains auditable—so that "aligned for whom, under what regime" remains a question we can answer with logged evidence rather than with aspiration.