Personalized Alignment with Guarantees: Multi-Objective Constrained DPO under Endogenous Prompts and Fairness Caps

Liz Lemma Future Detective

January 22, 2026

Abstract

Preference optimization methods such as Direct Preference Optimization (DPO) provide a simple RL-free route to aligning language models by learning from pairwise comparisons. However, 2026-era alignment is dominated by heterogeneous demand: different user populations and regulatory environments impose distinct objectives (e.g., helpfulness vs harmlessness, adult vs minor settings, enterprise compliance). Simultaneously, preference data are observational and confounded because users choose prompts endogenously, creating objective–prompt correlations that break naive generalization. We develop a clean multi-objective framework for conditional alignment in which the deployed policy is $\pi_\theta(y \mid x, c)$, conditioned on an objective/type c, and is trained from type-conditional preference data using a constrained DPO objective with interpretable Lagrange multipliers enforcing groupwise harm caps. Building on DPO's change-of-variables view (policy as an implicit reward) and recent causal analyses of preference learning (confounding and overlap), we (i) characterize the closed-form welfare-optimal constrained policy as an exponential tilt of a type-conditional reference model, (ii) provide identification and consistency conditions for constraint-augmented DPO under overlap in a causal feature space, (iii) derive generalization bounds that separate within-type estimation error from cross-type overlap error, and (iv) propose multihead/adversarial representation learning as a practical method to reduce prompt-type confounding. Empirically, we outline evaluations on HH-style helpful/harmless data and multi-domain preference logs, including stress tests under prompt-type shifts and type misreporting. The result is a principled alternative to one-size-fits-all alignment with explicit welfare and safety guarantees relevant to regulation and product design.

Table of Contents

preference learning.

2. 2. Related work: DPO/RLHF regularization; causal preference learning (confounding/overlap, latent treatments); multi-objective reward modeling; fairness constraints in ML; personalization and hidden context in RLHF.

3. 3. Setup and primitives: user types/objectives c, endogenous prompt generation $p_c(x)$, responses, preferences via Bradley–Terry; harms and caps; reference policies; what is observable vs latent.

4. 4. The planner's problem: define the social objective (weighted subgroup welfare) with KL regularization and harm caps; discuss identifiability and reward equivalence classes.

5. 5. Closed-form characterization: derive $\pi^*(y \mid x, c)$ as exponential tilting of π_{ref} with Lagrange multipliers; discuss uniqueness up to prompt-only shifts and when numerical methods are needed.

6. 6. Constrained Multi-Objective DPO: define the empirical objective (type-conditional DPO with constraint penalties); show equivalence to MLE under a reparameterized Bradley–Terry model; propose group-specific β_c.

7. 7. Robustness under endogenous prompts: articulate overlap assumptions in a causal feature space $z = g(x, y)$; derive generalization bounds separating within-type estimation error and cross-type overlap error; discuss how confounding manifests as overlap failure.

8. 8. Representation learning to mitigate prompt-type confounding: multihead architecture (shared encoder, objective heads) and adversarial invariance to prompt-type; mapping to assumptions needed for the theory; when guarantees become heuristic.

9. 9. Pareto + fairness results: conditions under which conditional policies Pareto-dominate pooled policies subject to harm caps; interpretability of Lagrange multipliers as 'group conservatism prices'.

10. 10. Extensions: partial observability of c (proxy $\hat{c}$), misreporting incentives and equilibrium reporting; mixture-of-experts; time-varying objectives and online updates.

11. 11. Empirical plan: HH-style dataset with synthetic counterfactual objectives; multi-domain preference logs; stress tests (prompt-type shift, overlap degradation, length/format artifacts, misreporting); metrics (subgroup win rates, harm cap violations, KL drift).

12. 12. Discussion and policy implications: audit and logging requirements; when personalization is necessary; setting β_c and harm caps; limitations and open questions.

1 Introduction

Modern alignment practice increasingly resembles platform design: we deploy a single, general-purpose language model into an environment where many users arrive with heterogeneous objectives, and where the platform is accountable for both usefulness and safety. Even when a model is advertised as "general," usage patterns quickly segment: some users seek terse factual answers, others want brainstorming, others want code, tutoring, emotional support, or role-play. The central difficulty is that these objectives are not merely different loss functions evaluated on a fixed data distribution; they shape which prompts users choose to write, which in turn shapes what the model is trained and evaluated on. In other words, demand is heterogeneous and endogenous, and the resulting feedback loop interacts nontrivially with safety constraints.

A standard response in current RLHF/DPO pipelines is to pool preference data and train a single policy that performs well on average. This pooling choice is often implicit: preferences are aggregated over annotators, prompts, and contexts, and the objective is optimized under a global regularizer (typically a KL penalty to a reference policy) plus whatever safety filters or rejection rules are applied. Pooling is attractive because it avoids personalization infrastructure and because it appears to reduce variance. However, it also forces compromises whenever subpopulations disagree about which responses are better, and it can systematically misrepresent minority objectives when the training mixture reflects platform convenience rather than a normative allocation. Moreover, because prompt distributions differ across user groups, pooled training can inadvertently learn spurious correlations that hold in the training mixture but fail for any particular group at deployment.

We can see this tension in mundane examples. Suppose one user type values creative elaboration while another values brevity; a pooled policy may respond with medium-length answers that satisfy neither. More importantly for safety, suppose a subset of users seeks instruction-like content in high-risk domains (e.g., chemistry, cybersecurity), while other users never touch these domains. Pooled training data will correlate certain prompt features with high-risk content, and naive optimization can drive the policy into regimes where it either over-refuses benign requests for the cautious group or under-refuses risky requests for the vulnerable group. In both cases, the failure mode is amplified by endogenous prompt selection: users adapt their prompts to obtain desired behavior, so the prompt distribution is itself a moving target influenced by the policy.

Our aim in this work is to make this deployment reality explicit in the mathematical object we optimize. We model a finite set of user types/objectives $c \in \mathcal{C}$, each with its own prompt distribution $p_c(x)$ and latent utility $u_c(x, y)$ over model responses. This is deliberately a stylized representation: "type"

can stand in for a mixture of user intent, context, risk tolerance, and jurisdictional constraints. The key is that conditioning on c is not merely a convenience; it is a mechanism for avoiding compromises that are artifacts of pooling and for expressing normative decisions (via planner weights and constraints) in a way that can be audited. Conceptually, we can interpret the platform as choosing a training and deployment procedure that maps observed information to a type-conditional policy $\pi_\theta(y \mid x, c)$, potentially routed by an observed proxy $\hat{c}$.

We then connect this deployment picture to an optimization problem that is simultaneously familiar from RLHF and more transparent about safety trade-offs. Starting from a reference policy π_{ref} (e.g., an SFT model), we consider a KL-regularized welfare objective that trades off type-conditional expected utility against deviation from π_{ref}, with type-specific conservatism β_c. Crucially, we incorporate explicit harm metrics $h_m(x, y)$ and groupwise caps $\delta_{m,c}$ that can encode regulatory requirements, platform policy, or user-protection standards. This yields a constrained optimization problem whose Lagrangian reveals the operational meaning of safety constraints: each binding cap induces a multiplier $\lambda_{m,c}$ that prices the corresponding harm within the affected type, effectively subtracting a penalty from the utility signal.

This structure gives an immediate conceptual payoff: under standard regularity conditions, the welfare-optimal feasible policy has a closed-form "exponential tilting" solution,

$$\pi^*(y \mid x, c) \;\propto\; \pi_{\text{ref}}(y \mid x, c) \exp\left(\frac{u_c(x, y) - \sum_m \tilde{\lambda}_{m,c} h_m(x, y)}{\beta_c}\right),$$

which makes precise what practitioners often describe informally as "start from a safe baseline and nudge toward preferred behavior." Here, β_c governs how aggressively we move probability mass away from the baseline for type c, while $\tilde{\lambda}_{m,c}$ governs how strongly we avoid particular harms. This view also clarifies a common governance question: what does it mean to "tighten a policy"? In our model it corresponds to reducing $\delta_{m,c}$, which (when the constraint binds) increases $\lambda_{m,c}$ and thereby shifts the entire distribution toward responses with lower expected h_m.

A second payoff is methodological. Direct Preference Optimization can be interpreted as maximum-likelihood estimation in a Bradley–Terry preference model after a change of variables that identifies the reward only up to a function of the prompt. We leverage this interpretation to motivate a constrained, type-conditional DPO objective: we fit $\pi_\theta(\cdot \mid x, c)$ so that the induced log-ratio reward $\hat{r}_{\theta,c}(x, y) = \beta_c \log \frac{\pi_\theta(y|x,c)}{\pi_{\text{ref}}(y|x,c)}$ matches preference differences within each type, while adding penalties corresponding to harm caps. Under correct specification of c (or under small proxy error), this objective is consistent for the same family of constrained optima characterized above. Intuitively, DPO learns the utility differences that explain

within-type comparisons; the constraint terms then shift the learned reward by harm prices, producing precisely the exponential tilt implied by the Lagrangian.

The central obstacle in making these claims meaningful is generalization under endogenous prompts. If type c tends to ask only a narrow slice of prompts, the preference dataset for c will overrepresent that slice; optimizing a conditional policy can then overfit to correlates of utility that do not hold when the prompt distribution shifts (e.g., when users learn to "prompt around" refusals). To address this, we state an overlap/positivity condition in a causal feature space $z = g(x, y)$, together with a density-ratio overlap constant κ that quantifies how different types (or training and deployment distributions) can be over relevant features. This yields a decomposition of regret into within-type estimation error plus a shift term amplified by κ. Operationally, this highlights a safety-relevant failure mode: weak overlap does not merely increase variance; it makes confounding-induced generalization failures likely, because the model can exploit spurious prompt features that are predictive in-sample but unstable out-of-sample. It also suggests concrete mitigations: targeted data collection to improve overlap, representation learning that maps prompts and responses into more causally stable z, and evaluation protocols that explicitly measure performance under distribution shift rather than only on the observed mixture.

Personalization is often criticized as incompatible with safety, because it might allow a user to self-select into a less restricted model. Our framework separates two questions that are frequently conflated: (i) should the platform optimize different utility trade-offs for different objectives, and (ii) should it permit different safety constraints for different users? We allow both to be explicit design choices. When harm caps are fixed (possibly equal across types), conditioning on c can be Pareto-improving in welfare relative to any pooled policy whenever genuine preference heterogeneity exists. The reason is simple: a pooled policy must compromise on response pairs where types disagree, whereas a conditional policy can tilt probability mass in opposite directions for different types without harming other groups. This observation is not merely about "user satisfaction"; it matters for governance because it implies that imposing a single pooled objective may be an avoidable source of both dissatisfaction and pressure to weaken safety constraints.

Finally, we confront a practical limitation: type c may not be directly observed. In deployment, we might only have a noisy proxy $\hat{c}$ inferred from context, user settings, or interaction history, and misrouting can create both welfare loss and safety risk. We provide a robustness statement: if $\mathbb{P}(\hat{c} \neq c) \leq \epsilon$, then welfare loss relative to the ideal type-routed policy scales as $O(\epsilon)$ under bounded utilities and mild smoothness. This formalizes a common engineering intuition: imperfect personalization can still be beneficial if misclassification is rare and if safety margins are designed to be robust to occasional routing errors.

In sum, our contributions are to (a) formalize heterogeneous, endogenous demand in a KL-regularized, safety-constrained alignment objective; (b) connect constrained, type-conditional DPO to the resulting constrained optimum via a transparent Lagrangian interpretation; and (c) articulate the key statistical and governance conditions—overlap, proxy-type error, and binding safety caps—under which personalization is welfare-improving rather than merely a product feature. Throughout, the formalism is not an end in itself: it is a way to expose the safety trade-offs we are already making implicitly, and to make them legible enough to be audited, stress-tested under shift, and adjusted when policy or regulation changes.

2 Related work

Our framing sits at the intersection of RLHF-style alignment, causal preference learning under selection, and constrained optimization for safety and fairness. We briefly situate each ingredient and highlight where our emphasis differs: rather than treating the training distribution as fixed, we treat heterogeneous user demand as endogenous, and we use this to motivate type-conditional regularization and explicit groupwise harm constraints.

RLHF, KL regularization, and "stay close to a reference." Most deployed alignment pipelines combine (i) supervised fine-tuning (SFT) toward demonstrations, (ii) preference modeling or direct preference learning from pairwise comparisons, and (iii) a regularized policy-improvement step that controls deviation from a reference model **???**. The KL penalty (or equivalent trust-region constraint) plays a dual role: it stabilizes optimization and acts as a conservatism prior that limits unanticipated capability shifts and reward hacking **??**. From a maximum-entropy control perspective, KL-regularized expected reward leads to a Gibbs / exponential-tilting form of the optimal policy relative to the reference, a point that appears across entropy-regularized RL and control-as-inference literatures **??**. Our contribution is not to re-derive this classical structure, but to emphasize its *type-conditional* interpretation: different subpopulations may warrant different conservatism parameters and different harm prices, and making these explicit renders a set of otherwise implicit product and governance decisions auditable.

Direct Preference Optimization and likelihood-based preference learning. DPO and related methods replace explicit reward-model training plus RL with a direct optimization of the policy using a Bradley–Terry / logistic likelihood on pairwise preferences **??**. This line of work highlights an important identification fact: the learned "reward" is only determined up to an additive, prompt-dependent baseline, and it is the *difference* in

rewards that drives preference probabilities. That observation is often presented as a technical convenience (enabling stable training without a separate reward model), but it also clarifies what aspects of feedback are and are not recoverable from comparisons. We build on this likelihood view to motivate a constrained variant in which harm penalties enter as Lagrange terms, aligning the statistical target of DPO with the policy form implied by KL-regularized constrained welfare maximization. Related work on IPO, KTO, and other preference-based objectives can also be interpreted through this lens as making different modeling choices about noise, margins, or calibrations of the implicit reward **??**.

Confounding, endogenous data collection, and causal preference learning. A recurring challenge in preference learning is that labels are collected under a policy and a prompt distribution that are themselves shaped by deployment and experimentation. In interactive settings, selection effects arise because the platform chooses which outputs to show (and therefore which comparisons are observed), while users adapt their prompts in response to prior outcomes **??**. Even in static datasets, "prompt-type confounding" can occur when different user groups systematically produce different kinds of prompts and also have different preference functionals, so that correlations between prompt features and preferred responses do not transport across groups. Causal inference and domain adaptation literatures formalize these issues using overlap/positivity and density-ratio control, yielding bounds where generalization degrades with weak overlap **??**. Recent work on causal representation learning and invariant prediction similarly aims to map observations into feature spaces where relationships are more stable across environments **??**. Our use of a causal feature map $z = g(x, y)$ and an overlap constant κ is in this spirit: it is a compact way to express when preference estimates within a type can be trusted to generalize under shifts induced by adaptation, policy updates, or cross-type pooling.

Latent context, hidden treatments, and preference heterogeneity. Several strands of work emphasize that feedback often depends on unobserved context: user intent, expertise, risk tolerance, jurisdictional rules, or situational constraints that are not fully captured by the prompt text. In RL and bandits, this is modeled as partially observed state (POMDPs), latent user embeddings, or hidden confounders affecting both action and reward **??**. In RLHF specifically, annotator heterogeneity and prompt-dependent label noise can be viewed as latent variables, and multiple works study aggregation, mixture models, and disagreement-aware training **??**. Our focus is on the decision-theoretic implication: if heterogeneity is genuine (types rank some response pairs in opposite directions), then any single pooled policy must compromise on those pairs, whereas conditioning on (observed or

inferred) type can avoid the compromise. This connects to classic results in social choice and mechanism design about aggregation under conflicting preferences, but in our setting the key additional complication is that the distribution over prompts is endogenous and correlated with type.

Multi-objective reward modeling and constrained alignment. Alignment systems are routinely trained against multiple criteria: helpfulness, harmlessness, honesty, style, and domain-specific policy compliance. Practically, this is handled via reward shaping, scalarization, rejection sampling, and "constitution"-style rule hierarchies **??**. A large literature in multi-objective optimization studies scalarization, Pareto frontiers, and lexicographic constraints; in safe RL, constrained Markov decision processes and Lagrangian methods provide canonical tools for enforcing cost budgets **??**. Our formulation is closest in spirit to constrained optimization, but with two distinctions motivated by deployment: (i) constraints are *groupwise* (indexed by type), which reflects how many safety and compliance requirements are actually written (e.g., different rules for minors vs. adults, different jurisdictions), and (ii) we interpret the resulting multipliers as explicit "harm prices" that can be audited and adjusted, rather than as opaque coefficients in a monolithic reward.

Fairness constraints, group robustness, and normative weights. Imposing constraints or priorities that differ across groups has a long history in fair machine learning, including demographic parity, equalized odds, and related constraints **??**. Group distributionally robust optimization (Group DRO) similarly focuses on improving worst-group performance under mixture uncertainty **?**. In sequential settings, fairness constraints and safe exploration introduce additional complications because policies affect future data **?**. While our harm caps $\delta_{m,c}$ are not fairness constraints in the classical sense, they play a structurally similar role: they bound an externality (harm) for each subgroup, and they can bind differently depending on exposure and risk. Likewise, planner weights w_c make explicit that training mixtures encode normative choices; changing w_c is not merely reweighting data but changing the welfare objective. This separation between empirical mixture and normative weighting parallels discussions in fairness about the gap between observational prevalence and ethical priority.

Personalization, routing, and safety concerns. Finally, our type-conditional approach relates to personalization and conditional computation in large language models: mixture-of-experts routing, adapter selection, user embeddings, and context-conditioned policies **??**. In practice, product teams already expose user-selectable modes (e.g., "creative" vs. "precise"), and research prototypes explore preference profiles and user-specific alignment. A

common objection is that personalization can become a safety vulnerability if users can self-select into a less restrictive regime. Our framework clarifies the design degrees of freedom: one can allow type-dependent utility tradeoffs while holding harm caps fixed (or even tightened) across types, and one can quantify the effect of imperfect routing through proxy-type error ϵ. This connects to broader governance themes: auditing requires that the mapping from observed signals to types, and from types to constraints, be legible; verification requires measuring harms and constraint violations per group; and robustness requires safety margins that tolerate occasional misclassification without catastrophic externalities.

3 Setup and primitives

We model deployment as an interaction between heterogeneous users and a platform that trains and serves a language model under explicit safety constraints. The goal of this section is to make the data-generating process and the platform's degrees of freedom concrete: what varies across users, what the platform can condition on, what feedback is observed, and which quantities remain latent.

User objectives and types. We posit a finite set of user objectives or "types" $\mathcal{C} = \{1, \dots, K\}$. The index $c \in \mathcal{C}$ is not meant to be a demographic label per se, but a sufficient statistic for a preference functional over responses. Concretely, a type may represent an application domain (e.g., tutoring vs. coding assistance), a risk tolerance (e.g., medical triage vs. general wellness), a jurisdictional regime, or a product mode (e.g., "creative" vs. "precise"). Each type c induces a latent utility function

$$u_c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R},$$

where $x \in \mathcal{X}$ is a natural-language prompt and $y \in \mathcal{Y}$ is a natural-language response. We allow $u_c(x, y)$ to encode any desiderata—helpfulness, truthfulness, style—provided it yields a coherent ranking over candidate responses for a fixed prompt and type.

Endogenous prompt generation. A central feature of deployment is that prompts are not exogenous. Users adapt what they ask based on their objectives, prior experiences, and the platform's affordances. We capture this by letting prompts be drawn from a type-conditional distribution $x \sim p_c(x)$. We emphasize that p_c is an *equilibrium object*: it summarizes the composition of user demand after accounting for selection into the platform, prompt-engineering behavior, and any interface or policy changes that shape what users choose to request. In a static analysis, we treat p_c as fixed, but we interpret it as implicitly dependent on the deployed system and surrounding

context. This is precisely where "prompt-type confounding" enters: if p_c differs substantially across types, then naive pooling of data can conflate type differences in preferences with differences in the prompts those types tend to issue.

Policies, conditioning, and responses. Given a prompt x and type c, the deployed model samples a response from a type-conditional policy $\pi_\theta(\cdot \mid x, c)$. The parameter vector θ stands for both model weights and any conditional computation mechanisms (e.g., adapters, routing, system prompts) that implement type conditioning. We also fix a type-conditional reference policy $\pi_{\text{ref}}(\cdot \mid x, c)$, intended to capture baseline behavior prior to preference optimization (for instance, an SFT model or a previous checkpoint). The reference plays two roles: it defines a default distribution over responses for each (x, c), and it anchors the notion of "distance moved" by training.

Pairwise preference feedback and the Bradley–Terry model. Training feedback is modeled as pairwise comparisons. For a given prompt x (and, conceptually, a type c), the platform produces two candidate responses y and y', and an annotator or user provides a binary label $L \in \{0, 1\}$ indicating which response is preferred. We use a type-conditional Bradley–Terry (logistic) model for this comparison noise:

$$\mathbb{P}(L = 1 \mid x, y, y', c) = \sigma\big(u_c(x, y) - u_c(x, y')\big),$$

where $\sigma(t) = 1/(1 + e^{-t})$. This assumption is not that humans are logistic, but that comparison probabilities depend monotonically on a latent utility difference, with larger gaps producing more reliable labels. The key identification implication (which we leverage later) is that only *differences* in $u_c(x, \cdot)$ are learnable from comparisons; any additive baseline $f_c(x)$ cancels in $u_c(x, y) - u_c(x, y')$.

Harms as measurable costs and groupwise caps. Alongside user utility, we introduce harm metrics that represent externalities the platform wishes (or is required) to limit. Let $h_m(x, y) \geq 0$ denote harm metric m evaluated on the prompt–response pair (x, y). Examples include a policy-violation indicator, a toxicity score, an unsafe-instruction classifier output, or a domain-specific compliance flag. We allow h_m to be either directly measurable (e.g., via automated detectors) or approximated (e.g., via audits or human review), but we treat it as a function that can be aggregated in expectation under a policy.

Crucially, we index constraints by type: for each metric m and type c, we impose a cap $\delta_{m,c}$ on the expected harm experienced when serving that

type. Formally, if $X \sim p_c$ and $Y \sim \pi(\cdot \mid X, c)$, we consider constraints of the form

$$\mathbb{E}\big[h_m(X, Y) \mid c\big] \leq \delta_{m,c}.$$

This captures the deployment reality that safety and compliance requirements are often conditional (e.g., stricter rules for minors, different medical disclaimers for health contexts, different legal regimes). It also anticipates a governance desideratum: constraint satisfaction should be auditable *per group*, not merely on average.

Conservatism and type-specific regularization. We parameterize the platform's reluctance to deviate from π_{ref} using a type-specific temperature $\beta_c > 0$. Intuitively, β_c controls how aggressively we allow the policy to chase estimated utility improvements for type c versus staying near a known baseline. This accommodates heterogeneous risk: some contexts warrant conservative updates because the cost of distribution shift is high or the evaluation pipeline is less reliable. While β_c enters the planner's objective in the next section, it is helpful already at the primitive level: it is a knob the platform can set by product policy, regulatory guidance, or empirical calibration.

Proxy types and imperfect routing. In many settings the platform does not observe c directly. Instead, it observes a proxy $\hat{c}$ inferred from user-provided signals, account attributes, or a classifier on the prompt and metadata. We allow for misrouting: $\mathbb{P}(\hat{c} \neq c)$ may be nonzero, reflecting both classification error and strategic behavior. Our analysis therefore distinguishes the *latent* type c that determines utility and constraints from the *observed* proxy $\hat{c}$ used for conditioning. This distinction matters both for welfare (the model may optimize the wrong objective for a user) and for safety (the wrong constraint regime may be applied). We treat misrouting probability as a primitive error rate that can, in principle, be reduced via better inference or mitigated via safety margins.

Causal features and overlap. To reason about generalization under shifting prompts, we will sometimes map (x, y) into a representation $z = g(x, y)$, interpreted as a causal feature sufficient for utility. The associated overlap/positivity requirement is that, for relevant regions of $\mathcal{Z}$, each type's induced distribution $p_c(z)$ assigns nontrivial mass, so that within-type learning does not extrapolate entirely out of support. We record this here as a modeling component: the platform may choose architectures or representation-learning procedures that encourage such invariances, but the success of those choices is constrained by the underlying data-collection process.

What the platform observes (and what it does not). The platform's training data consists of tuples (x, y, y', L), together with either the true type c (in settings where type is explicitly labeled) or the proxy $\hat{c}$. The platform can also log model likelihoods under π_{ref} and π_θ, and it can estimate harms $h_m(x, y)$ via detectors or audits. By contrast, the true utilities $u_c(x, y)$ are latent, as are the prompt distributions p_c except through observed traffic. This asymmetry is the core technical challenge: the platform must infer how to improve policy from comparisons that identify only utility differences, under a training distribution that is itself shaped by user adaptation and by past policy choices.

Timing and the induced data-generating process. Putting the pieces together, a stylized interaction proceeds as follows. First, the platform (and possibly a regulator) selects the reference policy π_{ref}, the conservatism parameters $\{\beta_c\}$, and harm caps $\{\delta_{m,c}\}$. Next, a user of latent type c arrives and generates a prompt $x \sim p_c$. The platform then produces candidate responses (for instance, by sampling from current policies or via a proposal mechanism) and obtains a comparison label L according to the Bradley–Terry model. Finally, the platform updates θ using these logged comparisons, with access to the relevant type signal (true or proxied) and to harm measurements. Deployment repeats this loop, with the important caveat that policy updates can shift user behavior and therefore shift p_c over time.

This completes the primitives. In the next section we formalize the platform's objective as weighted welfare maximization with KL regularization and explicit harm caps, and we discuss the identification consequences of learning from comparisons in the presence of type conditioning and reward equivalence classes.

4 The planner's problem: welfare, regularization, and constraints

We now make explicit the platform's (or regulator's) normative criterion for choosing a deployed policy. The key modeling choice is to treat deployment as *type-conditional decision-making* under two kinds of limitations: (i) the platform does not directly observe utilities, only noisy pairwise preferences; and (ii) the platform is not free to optimize user utility arbitrarily because it must remain close to a vetted baseline and satisfy explicit harm caps. This section formalizes the corresponding objective and clarifies what is and is not identified from preference data.

Weighted subgroup welfare. For each type $c \in \mathcal{C}$, we evaluate a type-conditional policy $\pi(\cdot \mid x, c)$ by its expected utility on the induced (endoge-

nous) prompt distribution p_c. We write the within-type welfare as

$$W_c(\pi) \;:=\; \mathbb{E}_{x\sim p_c}\,\mathbb{E}_{y\sim\pi(\cdot|x,c)}\big[u_c(x,y)\big].$$

A global planner then aggregates subgroup welfares using nonnegative weights w_c with $\sum_c w_c = 1$:

$$W(\pi) \;:=\; \sum_{c\in\mathcal{C}} w_c\, W_c(\pi).$$

The weights w_c can be interpreted in several ways depending on the institutional setting: as population shares in the platform's user base, as contractual priorities across product modes, or as explicit normative weights imposed by governance. We keep w_c exogenous, but it is worth flagging an operational subtlety: because p_c is endogenous, a platform policy can change who shows up and what they ask for, thereby shifting realized welfare weights over time. Our analysis holds p_c fixed to isolate the within-period training problem; in practice, p_c should be viewed as a slowly moving equilibrium object that motivates monitoring and periodic retuning.

KL regularization as conservatism and anchoring. Pure welfare maximization is ill-posed in language generation because it encourages extreme distribution shifts whenever the estimated reward is imperfect. We capture the platform's desire to remain close to a reference behavior via a type-specific KL penalty. For each type,

$$\mathrm{KL}_c(\pi\|\pi_{\mathrm{ref}}) \;:=\; \mathbb{E}_{x\sim p_c}\,\mathrm{KL}(\pi(\cdot\mid x,c)\,\|\,\pi_{\mathrm{ref}}(\cdot\mid x,c))\,,$$

and we introduce a temperature $\beta_c > 0$ that scales the marginal cost of deviating from π_{ref} for type c. The resulting regularized objective is

$$\sum_{c\in\mathcal{C}} w_c\Big(W_c(\pi) \;-\; \beta_c\,\mathrm{KL}_c(\pi\|\pi_{\mathrm{ref}})\Big).$$

This term has three interpretations that will matter later. First, it is a robustness device: when preference data are noisy or sparse for a type, the optimizer defaults back toward π_{ref}. Second, it reflects product risk tolerance: we can set larger β_c in higher-stakes contexts to dampen reward chasing. Third, and crucially for identification, KL regularization provides an *anchor* that ties any learned "reward" to a concrete probability model via log-likelihood ratios against π_{ref}.

Harm metrics and groupwise caps. We incorporate safety and compliance by imposing type-indexed constraints on expected harms. For each harm metric m and type c, define

$$H_{m,c}(\pi) \;:=\; \mathbb{E}_{x\sim p_c}\,\mathbb{E}_{y\sim\pi(\cdot|x,c)}\big[h_m(x,y)\big],$$

and impose caps $H_{m,c}(\pi) \le \delta_{m,c}$. This formulation is deliberately "auditable" in the sense that it is an expectation over logged traffic for a given type, and it enables different standards across regimes (e.g., stricter caps for certain product modes). We emphasize, however, that this is an *in-distribution* constraint with respect to p_c; if the prompt distribution shifts, constraint satisfaction may not transfer unless harms are sufficiently stable functions of the causal features. This is one place where overlap and representation choices become safety-critical rather than merely statistical conveniences.

Putting welfare, conservatism, and safety together, the platform's constrained planning problem is

$$\max_{\pi} \quad \sum_{c \in \mathcal{C}} w_c \Big(W_c(\pi) - \beta_c \, \mathrm{KL}_c(\pi \| \pi_{\mathrm{ref}}) \Big) \tag{1}$$

$$\text{s.t.} \quad H_{m,c}(\pi) \;\le\; \delta_{m,c} \qquad \forall (m,c). \tag{2}$$

We interpret π as ranging over all type-conditional stochastic policies with the appropriate measurability properties. Implicitly, feasibility requires that for each type there exist responses in the support of $\pi_{\mathrm{ref}}(\cdot \mid x, c)$ that satisfy the caps on average; if not, the constraints specify an unattainable safety regime and the dual variables (introduced next section) would diverge. In practice, this is exactly why platforms often combine policy optimization with dataset curation and pretraining/SFT adjustments: feasibility can be a property of the baseline model, not merely of the optimizer.

The Lagrangian viewpoint and "priced" harms. Although we postpone the closed-form solution to the next section, it is helpful to preview the structure of the constrained optimization. Introducing Lagrange multipliers $\lambda_{m,c} \ge 0$, we obtain the Lagrangian

$$\mathcal{L}(\pi, \lambda) \;=\; \sum_{c} w_c \Big(W_c(\pi) - \beta_c \, \mathrm{KL}_c(\pi \| \pi_{\mathrm{ref}}) \Big) \;-\; \sum_{m,c} \lambda_{m,c} \big(H_{m,c}(\pi) - \delta_{m,c} \big).$$

This makes the economic meaning of the harm caps explicit: at the optimum, $\lambda_{m,c}$ acts like a *shadow price* on harm metric m for type c. When a constraint is slack, its multiplier is zero and that harm is effectively ignored by the optimizer; when a constraint binds, the multiplier is positive and the policy behaves as if it were optimizing a modified utility $u_c - \sum_m (\lambda_{m,c}/w_c)\, h_m$. This priced-harm perspective is important for governance, because it suggests a path toward verification: one can audit not only whether caps are met, but also whether the implied harm prices are stable and interpretable across time.

Identifiability: utilities versus policies. The planner's problem is written in terms of latent utilities $u_c(x, y)$, but the platform never observes u_c di-

rectly. Under the Bradley–Terry assumption, pairwise preference data identify only *differences* in utility: for fixed (x, c), adding an arbitrary prompt-only baseline $f_c(x)$ leaves all comparison probabilities invariant because

$$\big(u_c(x, y) + f_c(x)\big) - \big(u_c(x, y') + f_c(x)\big) \;=\; u_c(x, y) - u_c(x, y').$$

Thus, even with infinite data and correct type labels, u_c is identifiable only up to an equivalence class

$$u_c(x, y) \;\sim\; u_c(x, y) + f_c(x).$$

This is not merely a technical nuisance: it means any training objective that treats u_c as a cardinal reward is underspecified unless it is formulated to be invariant to these shifts.

Fortunately, the planner's optimization (1)–(2) is naturally compatible with this invariance. In the KL-regularized setting, the optimal policy depends on $u_c(x, \cdot)$ only through relative values across y for a given (x, c). Adding $f_c(x)$ multiplies the exponential weights for *all* responses by the same factor and is absorbed into the normalization constant. In other words, while u_c is not uniquely defined, the induced optimizer $\pi^*(\cdot \mid x, c)$ can still be well-defined as a function of the equivalence class, once we fix π_{ref} and β_c.

Anchoring rewards to log-ratios. This invariance also motivates the reward parameterization used by direct preference optimization methods. For any candidate policy π_θ, define the type-conditional *implicit reward*

$$\hat{r}_{\theta,c}(x, y) \;:=\; \beta_c \log \frac{\pi_\theta(y \mid x, c)}{\pi_{\text{ref}}(y \mid x, c)}.$$

Because $\pi_\theta(\cdot \mid x, c)$ must normalize over y, this reward is itself only identified up to a prompt-only shift: multiplying all $\pi_\theta(y \mid x, c)$ by the same factor is impossible, and correspondingly $\hat{r}_{\theta,c}$ is defined only up to adding $f_c(x)$. This is exactly the same equivalence class induced by pairwise comparisons. The important takeaway is that the combination of (i) pairwise feedback and (ii) KL anchoring to a reference distribution yields a coherent notion of what can be learned: not absolute utilities, but the reward *differences* that determine how probability mass should move relative to π_{ref}.

We will use these observations in the next section to characterize the optimizer of the planner's problem explicitly. The resulting form makes clear when type conditioning yields Pareto improvements, how harm caps translate into priced penalties, and why numerical training objectives like constrained DPO can be interpreted as approximating the same underlying regularized and constrained optimum.

5 Closed-form characterization: exponential tilting of the reference

The constrained planner problem in (1)–(2) has a useful structural property: because we regularize by KL to a fixed reference, the optimal policy can be written in closed form as an *exponential tilt* of π_{ref}. This characterization is more than algebraic convenience. It tells us (i) how preference information should reweight probability mass relative to a baseline, (ii) how harm caps enter as explicit "prices" on unsafe attributes, and (iii) what exactly we should expect numerical training procedures (such as DPO variants) to approximate.

A mild absolute-continuity requirement. KL regularization implicitly restricts attention to policies that are absolutely continuous with respect to the reference: if $\pi_{\text{ref}}(y \mid x, c) = 0$ then assigning $\pi(y \mid x, c) > 0$ yields infinite KL. Operationally, this means the optimizer can only reweight behaviors that the baseline already assigns nonzero probability. This is both a safety feature (we cannot "invent" entirely new modes of behavior outside the vetted support) and a limitation (if the baseline never produces a needed safe response, no amount of preference optimization can recover it without changing π_{ref} or the model class).

Lagrangian decomposition and pointwise optimality. Fix multipliers $\lambda = \{\lambda_{m,c}\}_{m,c}$ with $\lambda_{m,c} \geq 0$. Consider the Lagrangian from the previous section,

$$\mathcal{L}(\pi, \lambda) = \sum_c w_c \Big(W_c(\pi) - \beta_c \, \mathrm{KL}_c(\pi \| \pi_{\text{ref}}) \Big) - \sum_{m,c} \lambda_{m,c} \big(H_{m,c}(\pi) - \delta_{m,c} \big).$$

Because both the welfare term and the harm term are expectations under $\pi(\cdot \mid x, c)$, and because the KL term is an expectation of a pointwise KL, maximizing $\mathcal{L}(\pi, \lambda)$ over π decomposes over (x, c): for each type and prompt, we solve a regularized expected-reward maximization with an adjusted reward

$$\tilde{u}_{c,\lambda}(x, y) \;:=\; u_c(x, y) \;-\; \sum_m \tilde{\lambda}_{m,c} \, h_m(x, y), \qquad \tilde{\lambda}_{m,c} := \frac{\lambda_{m,c}}{w_c}.$$

The appearance of $\tilde{\lambda}_{m,c} = \lambda_{m,c}/w_c$ is a first indication that governance choices (the welfare weights w_c) and safety choices (the caps that determine $\lambda_{m,c}$) interact: holding $\lambda_{m,c}$ fixed, increasing w_c effectively reduces the per-type harm price and permits a more aggressive tilt toward utility for that type.

For a fixed (x, c), maximizing the Lagrangian over distributions $\pi(\cdot \mid x, c)$ subject to normalization yields the standard maximum-entropy/KL-regularized solution. Introducing a prompt-specific normalization multiplier $\alpha_c(x)$ to enforce $\sum_y \pi(y \mid x, c) = 1$ and taking first-order conditions gives

$$\pi^*(y \mid x, c) \;\propto\; \pi_{\text{ref}}(y \mid x, c) \exp\left(\frac{1}{\beta_c}\, \tilde{u}_{c,\lambda}(x, y)\right).$$

Writing the normalization constant explicitly, we obtain the partition function

$$Z_c(x) := \sum_{y \in \mathcal{Y}} \pi_{\text{ref}}(y \mid x, c) \exp\left(\frac{1}{\beta_c}\left(u_c(x, y) - \sum_m \tilde{\lambda}_{m,c} h_m(x, y)\right)\right),$$

so the optimal policy for fixed λ is

$$\pi^*(y \mid x, c) = \frac{1}{Z_c(x)}\, \pi_{\text{ref}}(y \mid x, c) \exp\left(\frac{1}{\beta_c}\left(u_c(x, y) - \sum_m \tilde{\lambda}_{m,c} h_m(x, y)\right)\right). \tag{3}$$

This is the precise sense in which the deployed model should be "the reference model, reweighted by (utility minus priced harm), tempered by β_c."

KKT conditions and the meaning of the multipliers. To recover the *constrained* solution to (1)–(2), we must choose multipliers λ so that the resulting π^* satisfies the constraints and complementary slackness. Under standard regularity (convexity in π, feasibility, and an appropriate constraint qualification), strong duality holds and there exists λ^* such that:

$$H_{m,c}(\pi^*) \le \delta_{m,c}, \qquad \lambda^*_{m,c} \ge 0, \qquad \lambda^*_{m,c}\big(H_{m,c}(\pi^*) - \delta_{m,c}\big) = 0 \quad \forall (m, c).$$

Complementary slackness makes the "price" interpretation crisp: if a harm metric m is comfortably below its cap for type c, then $\lambda^*_{m,c} = 0$ and the metric exerts no force on the optimum; if it binds, $\lambda^*_{m,c} > 0$ and the policy behaves as though it were optimizing an augmented utility that subtracts $\tilde{\lambda}^*_{m,c} h_m$.

Two safety-relevant failure modes are also visible. First, if the constraint set is infeasible given the support of π_{ref}, then no finite λ can satisfy the KKT system; the formal signal is that the dual problem drives some multipliers to $+\infty$. Second, if harms are measured with systematic error (e.g., h_m is a noisy classifier score), the policy may satisfy the *measured* caps while violating the intended underlying property; in this view, the multipliers become prices on proxy metrics, not on true harms.

Uniqueness: policies are pinned down even when utilities are not. Equation (3) also clarifies the identifiability discussion from the previous section. Because $u_c(x, y)$ is only defined up to adding a prompt-only function $f_c(x)$, one might worry that π^* is not well-defined. In fact it is: adding $f_c(x)$ to $u_c(x, y)$ multiplies the numerator in (3) by $\exp(f_c(x)/\beta_c)$ for *every* y, which cancels in $Z_c(x)$. Thus, while the cardinal utility is not identified, the induced *relative* reweighting of responses—and therefore the optimal policy—is invariant to these shifts.

For fixed multipliers λ and temperatures β_c, the mapping $(u_c, h_m, \pi_{\text{ref}}) \mapsto \pi^*$ is pointwise unique on the support of π_{ref}. Non-uniqueness can still arise in degenerate cases (e.g., if π_{ref} assigns zero mass to entire regions of $\mathcal{Y}$ or if multiple responses are exactly tied under $\tilde{u}_{c,\lambda}$), but these are typically knife-edge; in practice, the stochasticity of π_{ref} and the strict convexity of KL regularization make the solution effectively unique.

When do we still need numerical methods? The closed form (3) should be read as a characterization of the *target distribution*, not as an algorithm for producing it in modern language models. There are three distinct reasons numerical optimization remains necessary.

First, even if we could represent π^* exactly, we generally cannot compute $Z_c(x)$ by summing over $\mathcal{Y}$; $\mathcal{Y}$ is combinatorially large. This is the same computational barrier that prevents exact maximum-entropy inference in sequence models. Training therefore proceeds by parameterizing π_θ and using stochastic gradients to move its log-probabilities in the direction suggested by (3) without explicitly normalizing over all sequences.

Second, the multipliers $\lambda_{m,c}$ are typically not known ex ante. Even with perfect knowledge of u_c, we would still need to solve for λ so that the *expectation* constraints $H_{m,c}(\pi) \leq \delta_{m,c}$ hold under the induced policy and prompt distribution. This is a dual optimization problem that is usually handled by primal–dual methods (e.g., dual ascent on λ combined with policy updates), and it inherits all the familiar practical issues: step-size sensitivity, nonstationarity when p_c drifts, and the need for safety margins when h_m is estimated.

Third, and most importantly for deployment, we rarely optimize over the space of all stochastic policies. We optimize over a restricted family $\{\pi_\theta\}$ determined by architecture, conditioning interface, and training procedure. If π^* is not representable in this family (or if conditioning on c is imperfect and we must route by $\hat{c}$), then (3) remains the normative benchmark, but the realized optimum is only approximate and may require explicit regularizers, constraint penalties, or data collection interventions to close the gap.

In the next section we turn this characterization into a concrete learning objective based on pairwise preferences: we show how direct preference optimization can be interpreted as fitting precisely the log-ratio structure

suggested by (3), and how adding constraint penalties corresponds to learning the priced-harm version of utility implicit in the KKT system.

6 Constrained multi-objective DPO: likelihood fitting with safety prices

The exponential-tilt form in (3) tells us what the *target* conditional policy should look like once we have an appropriate notion of adjusted utility. The remaining question is how to estimate and realize that target from the supervision we actually collect in deployment: pairwise preference labels over model-generated responses, together with measurable (or approximable) harm signals. In this section we formalize a constrained, type-conditional variant of direct preference optimization (DPO) as a joint procedure for (i) fitting the log-ratio structure implied by (3) and (ii) enforcing groupwise harm caps via Lagrange penalties.

Preference data and the type-conditional Bradley–Terry likelihood. For each type $c \in \mathcal{C}$, suppose the platform collects tuples

$$(x_i, y_{i,w}, y_{i,\ell}, c_i) \sim \text{deployment process}, \qquad L_i \in \{0, 1\},$$

where $L_i = 1$ indicates that, conditional on (x_i, c_i), the labeler (or user) prefers $y_{i,w}$ to $y_{i,\ell}$. Under the maintained behavioral model,

$$\mathbb{P}(L = 1 \mid x, y, y', c) = \sigma\big(u_c(x, y) - u_c(x, y')\big),$$

maximum likelihood estimation would, in principle, fit a family of scores $s_c(x, y)$ whose differences match the utility differences. However, two features of our setting make direct utility estimation both unnecessary and potentially ill-posed. First, only *differences* matter, so u_c is identified only up to an additive $f_c(x)$; second, we ultimately need a *policy* $\pi_\theta(\cdot \mid x, c)$, not a standalone reward model.

DPO as a reparameterized likelihood: log-ratios to a reference. DPO resolves this by parameterizing scores via a policy ratio to a fixed reference:

$$\hat{r}_{\theta,c}(x, y) \;:=\; \beta_c \log \frac{\pi_\theta(y \mid x, c)}{\pi_{\text{ref}}(y \mid x, c)}.$$

Here $\beta_c > 0$ is a type-specific temperature that determines how sharply preference information should tilt us away from the reference. Substituting $\hat{r}_{\theta,c}$ as the score function in a Bradley–Terry model yields the induced preference probability

$$\mathbb{P}_\theta(L = 1 \mid x, y_w, y_\ell, c) = \sigma\Big(\hat{r}_{\theta,c}(x, y_w) - \hat{r}_{\theta,c}(x, y_\ell)\Big),$$

and therefore the negative log-likelihood loss

$$\mathcal{L}_{\mathrm{DPO}}(\theta) := \mathbb{E}_{(x,y_w,y_\ell,c)}\left[-\log\sigma\Big(\hat{r}_{\theta,c}(x,y_w) - \hat{r}_{\theta,c}(x,y_\ell)\Big)\right]. \tag{4}$$

This is the central equivalence: type-conditional DPO is simply maximum likelihood for a Bradley–Terry preference model in which the latent utility differences are represented by log-probability *differences* under π_θ relative to π_{ref}. In particular, DPO does not require explicit normalization over $\mathcal{Y}$ (we never compute $Z_c(x)$), but it still learns to adjust relative log-probabilities in the direction indicated by preferences.

The β_c parameters play two roles. Statistically, they scale the logits in (4) and therefore the effective noise level in the preference model: larger β_c makes the same log-ratio correspond to a smaller implied utility gap, which yields more conservative updates. Normatively, β_c is exactly the temperature appearing in (3), so matching β_c across the preference-fitting objective and the planner benchmark pins down a consistent notion of how far each group may drift from its baseline.

Adding harm penalties: learning priced utility under caps. To incorporate groupwise caps $\mathbb{E}[h_m(X,Y) \mid c] \le \delta_{m,c}$, we augment the preference-fitting loss with Lagrange penalties. Let $\widehat{H}_{m,c}(\pi_\theta)$ denote an estimator of

$$H_{m,c}(\pi_\theta) = \mathbb{E}_{x\sim p_c}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x,c)}[h_m(x,y)].$$

Operationally, $\widehat{H}_{m,c}$ can be computed by sampling $y \sim \pi_\theta(\cdot \mid x,c)$ on prompts x from the logged distribution for type c, and then applying a harm classifier or rule-based audit to obtain $h_m(x,y)$. We then define the constrained multi-objective objective

$$\mathcal{L}_{\mathrm{CDPO}}(\theta;\lambda) := \mathcal{L}_{\mathrm{DPO}}(\theta) \;+\; \sum_{m,c}\lambda_{m,c}\Big(\widehat{H}_{m,c}(\pi_\theta) - \delta_{m,c}\Big), \qquad \lambda_{m,c} \ge 0. \tag{5}$$

Minimizing (5) for fixed λ corresponds to fitting preferences while treating each harm metric as a *priced negative attribute*. If, in addition, we update multipliers by a dual method (e.g., projected gradient ascent $\lambda_{m,c} \leftarrow [\lambda_{m,c} + \eta(\widehat{H}_{m,c} - \delta_{m,c})]_+$), the procedure implements a stochastic approximation to the KKT system described earlier: binding constraints push $\lambda_{m,c}$ upward, increasing pressure toward low-harm outputs, while slack constraints let $\lambda_{m,c}$ decay toward zero.

A useful way to see the conceptual alignment with (3) is to rewrite the per-type, per-response "effective reward" that the procedure is implicitly fitting. Under the dualized formulation, the priced utility has the form

$$\tilde{u}_{c,\lambda}(x,y) = u_c(x,y) - \sum_m \tilde{\lambda}_{m,c} h_m(x,y), \qquad \tilde{\lambda}_{m,c} = \frac{\lambda_{m,c}}{w_c},$$

so changing governance weights w_c changes the conversion rate between global multiplier $\lambda_{m,c}$ and the per-type price $\tilde{\lambda}_{m,c}$. In practice, the same interaction appears if we implement w_c by reweighting losses from different types in the empirical objective: increasing w_c increases the marginal benefit of fitting type-c preferences relative to the marginal penalty from its harm terms, and therefore permits a more aggressive tilt for that subgroup before the dual updates force $\lambda_{m,c}$ large.

Why type-conditional DPO is the right level of granularity. When preferences differ across types, a pooled objective fits a single score function that must compromise on disagreeing pairs. Conditioning on c removes this negative transfer: we fit separate log-ratio functions $\hat{r}_{\theta,c}$, each anchored to its own $\pi_{\mathrm{ref}}(\cdot \mid x, c)$ and scaled by its own β_c. This matters even if the underlying model parameters are shared, because the conditioning interface allows the function class to represent multiple tilts simultaneously. Moreover, the harm constraints are naturally group-indexed: what counts as unacceptable (or what level of risk is tolerated) is often set by policy for specific deployment contexts, and the penalty structure in (5) makes those governance choices explicit.

Consistency as a statement about the induced policy, not the utility scale. Because the Bradley–Terry likelihood only identifies reward differences, and because $\hat{r}_{\theta,c}$ is defined only up to $f_c(x)$ equivalence classes, the clean object to target is the *policy* π_θ. Under overlap and correct conditioning, minimizing the population version of (5) yields a policy whose log-ratio to the reference matches a representative of the priced utility:

$$\beta_c \log \frac{\pi_\theta(y \mid x, c)}{\pi_{\mathrm{ref}}(y \mid x, c)} \;\in\; \left[u_c(x, y) - \sum_m \tilde{\lambda}_{m,c} h_m(x, y) \right] \;+\; \{f_c(x)\}.$$

Combined with the normalization implied by $\pi_\theta(\cdot \mid x, c)$, this recovers exactly the exponential-tilt form from (3) for some multipliers satisfying complementary slackness (in the idealized limit where the constraint estimator matches the true expectations). The key point is that DPO-style training is not merely heuristic: it is an efficient way to fit the *log-density ratios* that characterize the KL-regularized constrained optimum.

Practical failure modes and why they foreshadow the next section. Two issues remain central in deployment. First, the constraint estimators $\widehat{H}_{m,c}$ are typically computed using proxy harm models; if these proxies are biased or can be gamed, the learned $\tilde{\lambda}_{m,c}$ will price the proxy rather than the intended harm. Second, and more subtly, the entire empirical objective is evaluated under the realized prompt distribution for each type. Since prompts are endogenous and may shift as users adapt to the deployed model,

the data used to fit $\hat{r}_{\theta,c}$ can concentrate on a narrow subset of the causal feature space. This is precisely where overlap becomes a bottleneck: without adequate coverage, the fitted policy may generalize poorly even if it attains low training loss and satisfies measured caps on logged prompts. We therefore next formalize robustness under endogenous prompts via overlap assumptions in a causal feature space and derive bounds that separate within-type estimation error from cross-type shift error.

7 Robustness under endogenous prompts: overlap in a causal feature space

Our consistency discussion above is inherently *in-distribution*: it describes what happens when the empirical objective is evaluated under the same conditional prompt distribution that we care about at deployment. In practice, that conditional distribution is neither fixed nor exogenous. Users adapt their prompts to the deployed policy, different subpopulations probe different capabilities, and the platform itself may change the user interface in ways that shift what is asked. This endogeneity matters because preference learning (including DPO-style likelihood fitting) can appear stable on logged data while failing sharply after seemingly minor shifts in how prompts are phrased or which response attributes are salient.

To state a robustness claim that is not merely a rephrasing of i.i.d. generalization, we need to separate two questions: (i) how well we estimate preferences *within the support* of the type-c data we actually see, and (ii) how fragile the learned mapping is when the relevant support changes. The standard device is to reason in a *causal feature space* rather than in raw prompts.

A causal feature map and induced distributions. Let $z = g(x, y) \in \mathcal{Z}$ denote a representation intended to capture the causally relevant aspects of the interaction between the prompt and the response (e.g., task category, requested action, presence of sensitive content, degree of refusal, etc.). For each type c, the deployment process induces a distribution over these features,

$$p_c(z) \;:=\; \mathbb{P}\big(g(X, Y) = z \;\big|\; C = c\big),$$

where $X \sim p_c$ and $Y \sim \pi(\cdot \mid X, c)$ for the current policy π. The dependence of $p_c(z)$ on π is exactly where endogeneity enters: changing the policy can change which y are sampled (and thus which z are observed), and user adaptation can change which x are produced (and thus which z become likely). This is the selection channel by which a model can "learn the wrong lesson" from preference data: it only sees preference labels on the slice of $\mathcal{Z}$ that the current interaction loop reveals.

To make progress, we adopt the structural restriction that utilities depend on (x, y) only through z and c:

$$u_c(x, y) = \bar{u}_c\big(g(x, y)\big) = \bar{u}_c(z).$$

This is not a claim that g is known or perfect; rather, it is the idealization that lets us cleanly state what kind of coverage is needed for robust learning. When g is misspecified, our guarantees become approximate in exactly the ways discussed in the next section.

Overlap (positivity) as the anti-confounding condition. We say that type-conditional overlap holds if the relevant parts of $\mathcal{Z}$ have positive probability under each type:

$$p_c(z) > 0 \quad \text{for all } c \in \mathcal{C} \text{ and all } z \in \mathcal{Z}_{\text{rel}},$$

where $\mathcal{Z}_{\text{rel}}$ denotes the feature region on which we want the learned policy to behave well (e.g., the region encountered after users adapt, or the region defined by a safety evaluation suite). A more quantitative version is to bound density ratios by an *overlap constant*

$$\kappa \;:=\; \max_{c,c''} \sup_{z \in \mathcal{Z}_{\text{rel}}} \frac{p_c(z)}{p_{c''}(z)} \;\in [1, \infty].$$

Small κ means that feature coverage is similar across types and prompt regimes; large κ means that some features are almost never observed for some types (or under some policies), so generalization requires extrapolation.

This overlap condition is the precise way in which "confounding manifests as overlap failure" in our setting. If type c tends to ask a narrow family of prompts, then many features z become type-predictive even when they are not utility-relevant. A flexible policy or score model can then use those type-predictive artifacts as shortcuts for predicting preferences. Under a shift in $p_c(z)$—for example, users learn to elicit different styles of answers or start probing boundaries—those shortcuts break, and the learned policy can degrade while still looking calibrated on the historical slice of $\mathcal{Z}$.

From preference learning to a risk decomposition. To connect overlap to a generalization bound, it is convenient to view preference learning as a classification problem in feature space. Consider a pairwise example with $(z_w, z_\ell) = (g(x, y_w), g(x, y_\ell))$ and label $L \in \{0, 1\}$. Under the Bradley–Terry model and the restriction $u_c = \bar{u}_c \circ g$, we have

$$\mathbb{P}(L = 1 \mid z_w, z_\ell, c) = \sigma\big(\bar{u}_c(z_w) - \bar{u}_c(z_\ell)\big).$$

Let $\hat{r}_{\theta,c}(x, y)$ be the DPO score and define the induced *feature score* $\phi_{\theta,c}(z)$ via $\phi_{\theta,c}(g(x, y)) = \hat{r}_{\theta,c}(x, y)$ (this is exact if $\hat{r}_{\theta,c}$ factors through g, and approximate otherwise). The population DPO objective is then a logistic risk

over pairs:

$$\mathcal{R}_c(\theta) \;:=\; \mathbb{E}\Big[\ell\big(\phi_{\theta,c}(Z_w) - \phi_{\theta,c}(Z_\ell),\, L\big) \;\big|\; C = c\Big],$$

for the log-loss $\ell(t, L) = -L \log \sigma(t) - (1 - L) \log(1 - \sigma(t))$. Standard arguments (uniform convergence, stability, or algorithm-dependent bounds) yield a within-type estimation guarantee of the form

$$\mathcal{R}_c(\hat{\theta}) - \inf_{\theta \in \Theta} \mathcal{R}_c(\theta) \;\le\; \mathrm{EstErr}_c(n_c, \Theta, \delta),$$

where n_c is the number of type-c examples and δ is a confidence parameter. This is the part we *can* buy with more labels for type c.

The harder part is that our eventual welfare is evaluated under a potentially different distribution over Z (because prompts and sampled responses change). Let $q_c(z)$ denote a target feature distribution (e.g., induced by the eventual deployed policy and adapted prompts). Then the gap between performance measured under p_c and performance under q_c is governed by density ratios:

$$\mathbb{E}_{z \sim q_c}[f(z)] - \mathbb{E}_{z \sim p_c}[f(z)] = \mathbb{E}_{z \sim p_c}\left[\left(\frac{q_c(z)}{p_c(z)} - 1\right) f(z)\right],$$

so any bound requires controlling $\frac{q_c(z)}{p_c(z)}$ on $\mathcal{Z}_{\mathrm{rel}}$. If we take q_c to be another type's feature distribution (or a mixture over types), then κ provides exactly this control.

A stylized regret bound and its interpretation. Translating logistic risk into welfare regret requires additional regularity, because welfare depends on the *policy* induced by the learned scores rather than on the scores themselves. Under bounded utilities and a smooth mapping from scores to π_θ (as in the softmax/exponential-tilt form), one obtains a decomposition of the schematic form

$$\mathrm{Regret}_c \;\lesssim\; \underbrace{\mathrm{EstErr}_c}_{\text{finite-sample within type}} \;+\; \underbrace{(\kappa - 1)\,\mathrm{ShiftErr}_c}_{\text{overlap / confounding amplification}}, \qquad (6)$$

where $\mathrm{ShiftErr}_c$ captures how much the learned rule relies on features whose frequencies differ across regimes (or across types), and the $(\kappa - 1)$ factor captures the fact that even small reliance on such features can be magnified when density ratios are large.

The safety-relevant reading of (6) is that "more data" is not a universal remedy: if endogenous interaction produces a narrow $p_c(z)$, then $n_c \to \infty$ can drive EstErr_c to zero while leaving the second term large, because the model is never forced to learn behavior on regions of $\mathcal{Z}_{\mathrm{rel}}$ it does not see. This is the formal sense in which confounding is a *coverage* problem, not merely a *noise* problem.

How overlap failure looks operationally. Overlap failure in this setting often presents as one of the following patterns:

- **Prompt-channel shortcuts.** Certain surface forms of prompts become predictive of preference labels in the logged data (because they correlate with the user type or with which candidate responses were sampled), even though they are not the causal drivers of utility. The learned policy then overreacts to phrasing changes.
- **Policy-induced missingness.** If the current policy refuses broadly for a type, then we may never observe z corresponding to borderline-but-allowable assistance for that type. Preference learning cannot "pull back" toward the feasible frontier because the relevant comparisons are absent.
- **Constraint masking.** Harm estimators $\widehat{H}_{m,c}$ computed on logged prompts can look compliant while violations occur on rare z that were not sampled historically. In dual terms, $\lambda_{m,c}$ is tuned to the observed slice of $p_c(z)$ rather than to $\mathcal{Z}_{\mathrm{rel}}$.

All three are confounding-by-selection: the training distribution depends on user behavior and on the policy, so spurious correlates are easy to learn and hard to detect without deliberate exploration or targeted evaluation.

Implications for data collection and governance. The immediate implication is that overlap is not just a statistical assumption; it is a *deployment design choice*. Interfaces, sampling schemes for candidate responses, and evaluation policies determine which parts of $\mathcal{Z}$ are observed. From a governance standpoint, the harm caps $\delta_{m,c}$ and their enforcement should therefore be tied to a specified evaluation distribution (or a worst-case envelope over plausible q_c), rather than to whatever distribution happened to be induced during training. Otherwise, the platform can satisfy the letter of the caps while violating their intent under adaptation.

These observations motivate the next step: if overlap in the causal feature space is the bottleneck, then representation learning should aim to (i) construct a g that removes type-predictive but utility-irrelevant artifacts, and (ii) encourage invariances that make $p_c(z)$ more comparable across types and prompt regimes. This is where architectural choices (shared encoders with type-conditional heads) and adversarial objectives enter, and where our formal guarantees begin to depend on how well learned representations approximate the causal g postulated here.

8 Representation learning to mitigate prompt-type confounding

The preceding section isolates overlap in a causal feature space as the key quantity controlling robustness under endogenous prompts. In practice, however, we do not get to directly observe the "right" $g(x, y)$ for which $u_c(x, y) = \bar{u}_c(g(x, y))$ holds. The platform instead learns internal representations whose inductive biases are shaped by the training loop—including precisely the selection effects that generate prompt-type confounding. This creates a tension: our theory wants a representation that discards type-predictive artifacts of the interaction while preserving the semantics that actually drive preferences and harms. In this section we describe a concrete training template (multihead architectures plus adversarial invariance) that aims to approximate the structural assumptions behind our guarantees, and we flag where the mapping becomes heuristic.

A multihead parameterization of type-conditional scoring and policy. A convenient way to "factor" the learning problem is to separate (i) a shared feature extractor from (ii) type-specific decision rules. Let e_ψ be a shared encoder that maps an interaction to a feature vector,

$$z \;=\; e_\psi(x, y) \;\in\; \mathbb{R}^d,$$

and let $\{s_{\eta,c}\}_{c\in\mathcal{C}}$ be type-specific score heads producing a scalar preference score (a proxy for $\hat{r}_{\theta,c}$),

$$\phi_{\theta,c}(x, y) \;:=\; s_{\eta,c}\big(e_\psi(x, y)\big), \qquad \theta = (\psi, \eta).$$

We then train using the same pairwise logistic form as in DPO, but with $\phi_{\theta,c}$ as the parametrization of the score difference. This architecture is aligned with the conceptual decomposition $u_c(x, y) = \bar{u}_c(z)$: the shared encoder approximates g, while the heads approximate the family $\{\bar{u}_c\}$ (up to the usual equivalence class induced by pairwise comparisons). Operationally, multihead sharing improves sample efficiency for rare types, while still permitting genuine heterogeneity (different heads can disagree on the same z).

Why sharing helps *and* why it can fail. Sharing the encoder is not merely a computational convenience; it is an implicit claim that there exists a representation in which the types differ primarily in how they *value* responses rather than in what responses *are*. This is exactly the situation in which type-conditioning yields Pareto gains: there are common capabilities and common harm dimensions, but heterogeneous tradeoffs. The failure mode is negative transfer: if types occupy disjoint regions of interaction space, a single encoder can learn shortcuts that are predictive in the logged mixture (e.g., prompt

style) but non-causal for any fixed type. In terms of the overlap constant, encoder sharing can either reduce κ (by expressing different prompt regimes in a common semantic basis) or effectively increase it (by collapsing distinct regions and inducing spurious correlations inside z). This is why we need an explicit mechanism that penalizes type-predictive artifacts inside the shared representation.

Adversarial invariance as a proxy for "removing confounding." A standard device from domain adaptation is to train the representation to be approximately invariant to a nuisance variable—here, the user type (or its proxy) as it is expressed through prompt surface form and sampling artifacts. Concretely, introduce a discriminator d_ω that tries to predict c (or $\hat{c}$) from z:

$$d_\omega : \mathbb{R}^d \to \Delta(\mathcal{C}), \qquad \widehat{p}_\omega(c \mid z) = d_\omega(z).$$

We then jointly optimize (i) the preference loss for the heads and (ii) an adversarial loss that makes z uninformative about c:

$$\min_{\psi,\eta} \max_{\omega} \left\{ \sum_c \mathbb{E}\big[\ell(s_{\eta,c}(Z_w) - s_{\eta,c}(Z_\ell), L) \;\big|\; C = c\big] - \alpha\, \mathbb{E}[\log d_\omega(C \mid e_\psi(X, Y))] \right\}, \tag{7}$$

where $\alpha > 0$ tunes the strength of invariance (implemented via gradient reversal in practice). Intuitively, this discourages the encoder from representing prompt style cues that are predictive of type in the logged data, forcing the type-specific heads to rely on features that transfer across types.

How this connects to overlap and the constant κ. The adversarial term in (7) can be read as an attempt to make the *representation-level* distributions $\{p_c(z)\}$ more similar. In idealized settings, bounding the discriminator's advantage implies bounds on an integral probability metric between $p_c(z)$ and $p_{c'}(z)$, and hence reduces worst-case density ratios on regions where the encoder is smooth and non-degenerate. While this is not the same as directly controlling κ, it targets the same pathology: if type can be reliably predicted from z, then z is likely to contain selection-induced correlates of the preference label that will not be stable under shifts in prompting. Making z less type-informative is a pragmatic way to shrink the confounding amplification term in decompositions like (6).

A key subtlety: we want invariance to artifacts, not invariance to semantics. There is an obvious objection: if types genuinely ask different tasks, then making z independent of c seems to throw away useful information. The resolution is that we do not actually want to erase all type information; we want to erase the information pathways by which the

model can use *spurious* type-predictive features as shortcuts. Two common refinements are therefore important in deployment:

- **Conditional invariance.** If we have (even noisy) side-information about task category or safety-relevant context, say $t = t(x)$, we can train a discriminator for C conditioned on t, or equivalently feed t to the discriminator so that the encoder is only penalized for encoding type beyond what is explained by t.
- **Factorized representations.** We can explicitly split z into $(z_{\text{sem}}, z_{\text{sty}})$ and only adversarially debias one component, while allowing the other to capture the parts of the prompt that legitimately differ across types. This makes the intended inductive bias auditable: invariance is applied where we believe confounding lives.

Either way, the conceptual target remains the same: approximate a causal g that is stable under prompt adaptation, while allowing type-specific heads to express heterogeneous values.

Integrating harm estimation and constraint enforcement into the representation. Because our platform objective includes harm caps, it is not enough for z to support preference prediction; it must also support reliable approximation of $h_m(x, y)$ on the relevant slice of interaction space. A simple approach is to add shared harm heads $\{a_{\nu,m}\}_m$ predicting $\hat{h}_m$ from the same z and to include their losses (or Lagrangian penalties) during training:

$$\min_{\psi,\eta,\nu} \left\{ \text{PreferenceLoss}(\psi, \eta) + \sum_{m,c} \lambda_{m,c}\, \widehat{H}_{m,c}(\pi_\theta) + \gamma\, \text{HarmPredLoss}(\psi, \nu) \right\}.$$

This encourages e_ψ to retain safety-relevant features even if they are weakly predictive of preference labels in the logged data. It also mitigates a common failure mode of adversarial invariance: if harmful content is correlated with type, a naive discriminator can incentivize the encoder to hide precisely the information that a harm classifier needs. Adding explicit harm supervision (or worst-case safety evaluation) counterbalances that pressure.

When the theoretical guarantees are closest to literal. Under strong idealizations, the architecture above can be mapped back to our assumptions. If (i) there exists a representation g such that $u_c(x, y) = \bar{u}_c(g(x, y))$ and $h_m(x, y) = \bar{h}_m(g(x, y))$, (ii) the shared encoder class contains a close approximation to g, and (iii) the heads contain close approximations to $\{\bar{u}_c\}$ and $\{\bar{h}_m\}$, then constrained DPO with multihead scoring is a statistically plausible route to recovering the exponential-tilt policy form type-by-type. Moreover, if the adversarial objective succeeds in making $p_c(z)$ comparable

across types (reducing representation-level shift), then the overlap constant relevant for generalization in z-space can be materially smaller than the overlap constant in raw prompt space, strengthening the practical meaning of the decomposition in (6).

When the mapping becomes heuristic (and what can go wrong). In realistic deployments, each link in the mapping can break. First, u_c may not factor through any low-dimensional, type-stable z; in that case, no representation learning objective can make the problem "causal" without sacrificing accuracy. Second, adversarial invariance can trade robustness for capability: if type correlates with legitimate task content, the encoder may erase distinctions that are needed for good responses, pushing the burden onto the heads and potentially inflating sample complexity. Third, when c is replaced by a proxy $\hat{c}$, invariance objectives can become miscalibrated: the discriminator learns to predict the proxy (which may encode demographic or interface artifacts), and the encoder learns to hide proxy-predictive information in ways that are orthogonal to actual preference stability. Finally, none of these methods create support where none exists: if a type never produces prompts that lead to certain safety-critical z, then representation alignment cannot substitute for targeted data collection or deliberate exploration; it can only smooth over differences within the observed envelope.

Practical governance implication: representation learning is part of the safety case, not a purely "performance" choice. Because the representation mediates both preference generalization and harm measurement, architectural choices (shared encoder vs. per-type encoders, strength of invariance, whether harm heads are shared) should be treated as safety-relevant parameters. In particular, one can audit the learned z by checking (i) how well type can be predicted from z under matched task conditions, (ii) whether harm predictors remain calibrated on stress tests that deliberately shift prompt phrasing, and (iii) whether the induced $p_c(z)$ exhibits improved overlap on a pre-specified evaluation suite. These checks do not prove that z is truly causal, but they can detect the most operationally dangerous forms of prompt-type confounding.

With this machinery in place—type-specific heads expressing heterogeneous values, a shared representation aimed at stability under shifts, and explicit harm-sensitive training—we can now return to the welfare question: when, and in what sense, do type-conditional policies dominate pooled ones once we price harm and conservatism through the multipliers and temperatures?

9 Pareto improvements and fairness under harm caps

We now make precise a central welfare point that is easy to miss when focusing only on optimization mechanics: once we admit genuine preference heterogeneity across user types, a *pooled* deployment policy $\pi(y \mid x)$ is typically forced into avoidable compromises. In contrast, a *type-conditional* policy $\pi(y \mid x, c)$ can shift probability mass in different directions for different groups, and can do so while respecting the same safety requirements expressed as groupwise harm caps. This is the sense in which personalization is not merely a product feature; it is a structural route to Pareto improvements (subject to feasibility) in the presence of heterogeneous objectives.

Pooled versus conditional feasibility. Recall that the harm constraints we study are type-indexed expectations,

$$H_{m,c}(\pi) \;=\; \mathbb{E}_{x\sim p_c}\,\mathbb{E}_{y\sim\pi(\cdot|x,c)}[h_m(x,y)] \;\le\; \delta_{m,c}.$$

If we instead deploy a pooled policy $\pi(\cdot \mid x)$, the constraint becomes

$$H_{m,c}(\pi) \;=\; \mathbb{E}_{x\sim p_c}\,\mathbb{E}_{y\sim\pi(\cdot|x)}[h_m(x,y)] \;\le\; \delta_{m,c} \qquad \forall c,$$

so a single mapping $x \mapsto \pi(\cdot \mid x)$ must satisfy *all* groupwise bounds, despite the fact that the evaluation measure over prompts differs across types through p_c. This creates a characteristic "worst-case across types" pressure: even if type c rarely visits a risky region of prompt space, the pooled policy must behave conservatively there if some other type c' visits it often and faces a tight cap. Conditioning relaxes this coupling by allowing the platform to satisfy each set of constraints *on the distribution where it is actually evaluated.*

A local Pareto-improvement argument. The core Pareto claim can be seen through a local deviation. Fix a prompt x that is in the support of (at least) two types, and suppose there exist responses y^+ and y^- such that types disagree:

$$u_c(x,y^+) - u_c(x,y^-) \;>\; 0, \qquad u_{c'}(x,y^+) - u_{c'}(x,y^-) \;<\; 0.$$

Consider any pooled policy $\pi(\cdot \mid x)$ that assigns positive probability to both y^+ and y^-. For a small $\varepsilon > 0$, define a type-conditional perturbation that shifts ε mass from y^- to y^+ for type c, and shifts ε mass from y^+ to y^- for type c', leaving all other types unchanged. The first-order welfare changes satisfy

$$\Delta W_c \;\approx\; \varepsilon\cdot\big(u_c(x,y^+) - u_c(x,y^-)\big) \;>\; 0, \qquad \Delta W_{c'} \;\approx\; \varepsilon\cdot\big(u_{c'}(x,y^-) - u_{c'}(x,y^+)\big) \;>\; 0,$$

and $\Delta W_{c''} = 0$ for $c'' \notin \{c, c'\}$, up to the weighting of how often x occurs under p_c. In words: if the groups disagree on a pair, the pooled policy must place shared probability on a compromise mixture, but a conditional policy can move each group along its own gradient without creating an inter-group tradeoff on *that same pair*.

Accounting for harm caps. The same perturbation can be chosen to preserve (or improve) harm constraints. If, for a particular harm metric m, the swap increases harm for type c (i.e., $h_m(x, y^+) > h_m(x, y^-)$), we can either (i) take ε sufficiently small so that constraints remain satisfied when slack exists, or (ii) choose a disagreement pair (y^+, y^-) that is approximately harm-neutral for binding metrics. More generally, because the constraints are linear in the policy, we can construct a feasible improving direction by projecting the welfare-improving direction onto the tangent cone of the feasible set at π; feasibility is guaranteed whenever (a) each type's feasible set is nonempty and (b) there is at least one direction of improvement that does not increase all binding harms simultaneously. This is the operational content of our earlier feasibility assumption: constraints must not be so tight that any welfare-improving change necessarily violates safety.

Why KL regularization does not destroy the Pareto logic. KL regularization changes the geometry but not the underlying compromise story. Under the KL-regularized objective, the relevant first-order change is in

$$W_c(\pi) \;-\; \beta_c \,\mathrm{KL}_c(\pi \| \pi_{\mathrm{ref}}),$$

so moving probability mass away from π_{ref} incurs a cost. However, the KL term is also type-indexed through p_c and β_c, and hence can be paid independently by each type under conditioning. A pooled policy must pay KL costs to accommodate the most demanding combination of groups; a conditional policy can "spend" KL budget where it yields the most within-type welfare. This is precisely what the exponential-tilt optimum encodes: for each type, we tilt away from π_{ref} in proportion to a *priced* utility,

$$u_c(x, y) \;-\; \sum_m \tilde{\lambda}_{m,c}\, h_m(x, y), \qquad \tilde{\lambda}_{m,c} := \lambda_{m,c}/w_c,$$

with β_c scaling how sharply the policy concentrates on high-priced-utility responses.

Fairness as explicit constraint choices, not an implicit byproduct. The preceding Pareto improvement is compatible with many fairness concepts, but it does not by itself pick one. In our framework, fairness enters through *what is constrained* and *how welfare is aggregated*. Groupwise harm caps $\delta_{m,c}$ can be read as a form of safety parity constraint: each group is

entitled to an upper bound on expected exposure to harm metric m under its endogenous prompt distribution. Separately, the planner weights w_c encode distributive priorities over welfare (e.g., population share, a Rawlsian tilt, or a regulator-mandated priority weight). Importantly, these are design parameters that can be deliberated and audited, rather than emergent artifacts of model training.

Lagrange multipliers as "group conservatism prices." At an optimum, the multipliers $\lambda_{m,c}$ (equivalently $\tilde{\lambda}_{m,c}$) admit a standard shadow-price interpretation:

$$\lambda_{m,c} \;=\; \frac{\partial}{\partial \delta_{m,c}} \left(\max_{\pi} \sum_{c'} w_{c'} \big(W_{c'}(\pi) - \beta_{c'} \mathrm{KL}_{c'}(\pi \| \pi_{\mathrm{ref}}) \big) \;\text{s.t.}\; H_{m,c'}(\pi) \le \delta_{m,c'} \right),$$

whenever differentiability holds. Thus $\lambda_{m,c}$ measures the platform's marginal value of relaxing the m-harm cap for type c; if the cap is slack, complementary slackness forces $\lambda_{m,c} = 0$. From a safety perspective, the more relevant object is often $\tilde{\lambda}_{m,c}$, which is the *per-unit welfare weight* price that actually enters the exponential tilt. When $\tilde{\lambda}_{m,c}$ is large, the optimal policy behaves more conservatively for that group along metric m: the model requires a larger utility gain to justify even a small increase in h_m.

This connects directly to interpretability in deployment. If we can estimate h_m and the induced policy shift, we can treat $\tilde{\lambda}_{m,c}$ as a legible knob summarizing a complex tradeoff: it tells us how the system is pricing a particular harm for a particular group. Moreover, because $\tilde{\lambda}_{m,c} = \lambda_{m,c}/w_c$, changes in planner weights w_c mechanically change the effective conservatism price even if the regulator's cap $\delta_{m,c}$ is unchanged. This is a governance-relevant nonlinearity: prioritizing a group in welfare aggregation implicitly reduces the *relative* penalty of binding constraints for that group, which can be desirable (e.g., to ensure accessibility) or undesirable (e.g., if it increases safety risk). Making this dependence explicit helps avoid accidental policy choices.

Failure modes and audit questions. Two practical failure modes deserve emphasis. First, if harms are systematically mismeasured for some type—for example because h_m is calibrated on one prompt regime but evaluated on another—then the inferred $\tilde{\lambda}_{m,c}$ may look "reasonable" while the realized $H_{m,c}$ violates the intended cap. Second, if the type label is itself socially sensitive, then using per-type multipliers can create disparate *model behavior* even when caps are identical, because different groups may face different baseline feasibility regions under p_c. This is not automatically unfair, but it is exactly the kind of differential treatment that demands explicit justification and monitoring. In both cases, the multiplier view suggests concrete checks: measure which constraints are empirically binding, estimate slack-

ness, and stress-test whether the policy response to tightening $\delta_{m,c}$ matches the predicted marginal effect.

Where we go next. So far we have assumed that type-conditioning is available and correctly specified, so that the Pareto logic is not confounded by routing errors or strategic behavior. In deployment, however, c is often only partially observed (via a proxy $\hat{c}$), and users may have incentives to misreport if different types induce different levels of conservatism. These considerations motivate the next set of extensions: proxy routing robustness, equilibrium reporting, mixture-of-experts implementations, and online adaptation when objectives drift over time.

10 Extensions: partial observability, strategic reporting, modular policies, and nonstationarity

Our preceding analysis treated the type label c as if it were cleanly available for both training and deployment. In practice, the platform typically sees only a proxy $\hat{c}$ (e.g., coarse user setting, locale, subscription tier, classroom mode), and even when a self-declared label is available it may be strategically chosen. Moreover, the engineering implementation of $\pi_\theta(\cdot \mid x, c)$ is often a modular mixture rather than a monolithic conditional network, and both objectives and prompt distributions drift over time. We sketch how the framework extends in these directions, and where the clean Pareto and consistency conclusions become contingent on additional assumptions.

Partial observability and proxy routing. Suppose deployment routes on $\hat{c}$ and executes $\pi_\theta(\cdot \mid x, \hat{c})$, while true welfare and constraints are evaluated under c. A convenient abstraction is a (possibly x-dependent) confusion matrix $Q(\hat{c} \mid c, x)$ and the induced deployed mixture

$$\pi_\theta^{\mathrm{dep}}(y \mid x, c) \;:=\; \sum_{\hat{c} \in \mathcal{C}} Q(\hat{c} \mid c, x)\, \pi_\theta(y \mid x, \hat{c}).$$

Even if each per-proxy policy $\pi_\theta(\cdot \mid x, \hat{c})$ is feasible for the corresponding subgroup, feasibility for the true subgroup c is not automatic because (i) the prompt law is p_c and (ii) the deployed policy is a mixture over $\hat{c}$. A sufficient condition is to treat routing uncertainty as part of the constraint set and enforce robust caps of the form

$$\sup_{Q \in \mathcal{Q}} \; \mathbb{E}_{x \sim p_c}\, \mathbb{E}_{y \sim \sum_{\hat{c}} Q(\hat{c}|c,x)\pi_\theta(\cdot|x,\hat{c})}[h_m(x,y)] \;\le\; \delta_{m,c},$$

for an uncertainty class $\mathcal{Q}$ (e.g., all Q within total variation ϵ of a nominal router). This converts proxy error into a safety margin problem: one

can tighten $\delta_{m,c}$ during training to compensate for a bounded misrouting rate, or equivalently add an additional penalty term that upper-bounds the harm inflation due to routing noise. The welfare side behaves similarly: the realized welfare is that of a convex combination of near-correct policies, so bounds of the form $O(\epsilon)$ follow under bounded utilities, but only if the map $\hat{c} \mapsto \pi_\theta(\cdot \mid x, \hat{c})$ is not too discontinuous (a point we revisit under mixture-of-experts).

Identification under proxy labels. When training data is labeled by $\hat{c}$ rather than c, we face a mixture identification problem. The preference model becomes

$$\mathbb{P}(L = 1 \mid x, y, y', \hat{c}) \;=\; \sum_{c \in \mathcal{C}} \mathbb{P}(c \mid x, \hat{c})\, \sigma\big(u_c(x, y) - u_c(x, y')\big),$$

so the naive within-$\hat{c}$ Bradley–Terry fit targets a mixture of utilities rather than any single u_c. If $\mathbb{P}(c \mid x, \hat{c})$ varies with x (as is typical), this can introduce precisely the kind of prompt–type confounding that type-conditioning was meant to reduce. Two mitigation routes fit our formalism: (i) explicitly model c as latent and perform EM-style learning of $\mathbb{P}(c \mid x, \hat{c})$ jointly with per-type reward surrogates; or (ii) learn a representation $z = g(x, y)$ that reduces dependence on x-specific artifacts, thereby improving overlap in the causal feature space and making mixture deconvolution less brittle. Both routes shift the burden to assumptions about identifiability and coverage: without some form of overlap across types in z (or anchor points where $\hat{c}$ is informative), there is no guarantee that per-type utilities are recoverable from proxy-labeled comparisons.

Strategic misreporting and equilibrium routing. If users can influence $\hat{c}$ through self-report $\tilde{c}$, then routing becomes a game: the platform commits to a menu $\{\pi_\theta(\cdot \mid x, \tilde{c})\}_{\tilde{c}}$ and users select $\tilde{c}$ to maximize their own welfare (possibly trading off safety restrictions). A minimal model is

$$\tilde{c}(x) \;\in\; \arg\max_{\tilde{c} \in \mathcal{C}} \; \mathbb{E}_{y \sim \pi_\theta(\cdot \mid x, \tilde{c})}[u_c(x, y)],$$

with realized harms evaluated under the true type c. This immediately creates a failure mode: if some route has looser effective conservatism (lower $\tilde{\lambda}_{m,\tilde{c}}$ or larger $\delta_{m,\tilde{c}}$), then users with sufficiently aligned objectives may pool into that route, invalidating both welfare accounting and constraint guarantees. In equilibrium, the relevant prompt distribution for route $\tilde{c}$ is endogenously selected, which can destroy overlap and change which constraints bind.

Designing for incentive compatibility (or at least bounded damage). There are two broad governance-compatible responses. The first

is mechanism-design flavored: make reporting less payoff-relevant by constraining policies to be "approximately monotone" in safety with respect to observable risk factors, or by limiting cross-route divergence via shared KL budgets, so that no route is an obvious "escape hatch." Formally, we can impose coupling constraints such as

$$\mathrm{KL}\big(\pi_\theta(\cdot \mid x, \tilde{c}) \,\|\, \pi_\theta(\cdot \mid x, \tilde{c}')\big) \;\le\; \rho \quad \forall x,\ \tilde{c}, \tilde{c}',$$

which caps the utility gain from misreporting at the cost of reducing personalization benefits. The second response is robust accounting: treat $\tilde{c}$ as cheap talk, infer a posterior over c from behavioral signals, and enforce constraints against the induced mixture (worst-case or Bayesian). Either way, the key extension is that p_c is no longer exogenous: equilibrium reporting changes the distribution on which harms are realized, so feasibility must be stated with respect to the induced equilibrium prompt-and-route distribution, not the nominal p_c.

Mixture-of-experts as an implementation of type-conditioning. A practical architecture for $\pi_\theta(\cdot \mid x, c)$ is a mixture-of-experts (MoE): experts $\{\pi_{\theta_k}(\cdot \mid x)\}_{k=1}^{K}$ paired with a gating model $q_\phi(k \mid x, \hat{c})$, yielding the composite policy

$$\pi_{\theta,\phi}(y \mid x, \hat{c}) \;:=\; \sum_{k=1}^{K} q_\phi(k \mid x, \hat{c})\, \pi_{\theta_k}(y \mid x).$$

This view unifies clean type-conditioning (hard routing where q_ϕ is a delta at $k = \hat{c}$) with soft routing that can hedge when $\hat{c}$ is noisy. It also clarifies a subtlety: even if each expert is trained to satisfy harm caps under some distribution, the *mixture* can violate caps if the gate concentrates experts on regions of x that were underrepresented during their constraint calibration. Thus constraint enforcement should be done on the end-to-end mixture *under the realized gating distribution*, i.e.,

$$\mathbb{E}_{x \sim p_c}\, \mathbb{E}_{y \sim \sum_k q_\phi(k|x,\hat{c})\pi_{\theta_k}(\cdot|x)}[h_m(x,y)] \;\le\; \delta_{m,c},$$

with $\hat{c}$ itself drawn from the router. Operationally, this argues for logging and auditing at the *post-gate* level: the safety object is not the expert in isolation but the routed composition.

Training dynamics with MoE gates. From an optimization perspective, the Lagrangian picture survives but with new degrees of freedom. The platform can spend capacity either by sharpening experts (improving within-expert utility at fixed harms) or by improving the gate (routing each context to an expert with better priced utility). This creates a new failure mode: the gate can become a high-leverage classifier that exploits spurious correlates of c in x, making performance fragile under prompt shift. In terms of

our overlap constant κ, MoE systems can inadvertently increase effective κ by specializing experts too narrowly. A conservative remedy is to regularize the gate (entropy floors, Lipschitz constraints, or explicit domain-invariance penalties on $g(x, y)$), trading some personalization for better worst-case behavior under routing error and distribution shift.

Time-varying objectives and online updates. Finally, both p_c and u_c may drift with time: norms change, user populations shift, and products add new affordances. Indexing time by t, we may face $p_{c,t}$ and $u_{c,t}$ and thus a moving target π_t^*. Static optimality statements then become tracking statements. A natural extension is an online, constrained mirror-descent interpretation of DPO: at each epoch t, update θ to improve a preference likelihood term while penalizing KL drift from a stable reference and enforcing (or penalizing) empirical harms. One desideratum is a *dynamic regret* guarantee,

$$\sum_{t=1}^{T} \Big(W_{c,t}(\pi_t^*) - W_{c,t}(\pi_{\theta_t}) \Big) \;\le\; \text{poly(complexity)} \;+\; \text{Var}_T(\pi^*),$$

where $\text{Var}_T(\pi^*)$ measures how fast the optimum moves. Safety complicates this: constraint violations are not symmetric losses, and online learning under delayed or noisy harm signals can accumulate unacceptable debt. This motivates conservative update rules (small effective step sizes via larger β_c), explicit safety buffers (train to $\delta_{m,c} - \eta$), and change-detection triggers that freeze learning when harm metrics become unreliable under shift.

Open problems at the interface of alignment and governance. These extensions highlight a common theme: once types are noisy, strategic, or nonstationary, the main object of concern is not just maximizing a regularized welfare functional but controlling the end-to-end socio-technical feedback loop that determines which prompts are asked, which routes are selected, and which harms are realized. Formally, we move from optimization with fixed p_c to equilibrium selection with endogenous data, and from static feasibility to safety under distribution shift. Bridging this gap likely requires combining (i) causal identification tools for u_c and h_m under selection, (ii) mechanism constraints that limit incentives to route-shop, and (iii) verifiable monitoring that can certify, from logs, whether the deployed mixture respects the intended caps over time.

Empirical plan: datasets, synthetic counterfactual types, and stress tests. To make the preceding framework operational, we want an empirical protocol that (i) induces *controlled* preference heterogeneity across types, (ii) exposes the learner to realistic prompt distributions with type–prompt correlations, and (iii) measures both welfare improvements and safety failures

under the kinds of shifts that arise from proxy routing, modular policies, and endogenous user behavior. Concretely, we propose a suite of benchmarks built around an HH-style comparison dataset, augmented with synthetic counterfactual objectives and multi-domain preference logs, and evaluated under targeted stress tests designed to probe overlap, confounding, and strategic selection.

Base data: HH-style comparisons with typed prompts. We start from a standard "helpful–harmless" (HH) pairwise preference corpus: each instance consists of a prompt x, two candidate responses (y_w, y_ℓ), and a label L indicating the preferred response. We then attach a *type label* $c \in \mathcal{C}$ to each instance by either (a) collecting preferences from distinct rater pools with different instructions (e.g., "optimize terseness," "optimize pedagogy," "optimize creativity," "optimize compliance within policy"), or (b) simulating such pools using counterfactual relabeling rules defined below. The key requirement is that $p_c(x)$ differs meaningfully across c (endogenous prompt choice), so we stratify prompts into domains (e.g., programming, education, health, relationship advice, creative writing) and sample type-conditional mixtures over domains to create realistic prompt–type correlations.

Synthetic counterfactual objectives to induce ground-truth heterogeneity. To test identification and Pareto claims in a setting where we can compute "true" utilities, we construct synthetic objectives by defining a latent score

$$u_c(x, y) \;:=\; \alpha_c^\top \varphi(x, y) \;-\; \sum_m \gamma_{m,c}\, h_m(x, y),$$

where $\varphi(x, y)$ are precomputed features (e.g., task success, factuality proxy, verbosity, politeness, refusal style, citation presence) and $h_m(x, y)$ are harm detectors aligned with our deployed metrics. We then sample $\alpha_c, \gamma_{m,c}$ so that there exist response pairs with genuine sign disagreements across types (realizing the heterogeneous-preference condition underlying our Pareto separation). This construction yields two complementary datasets:

1. *Observed-preference dataset:* we retain human labels L and treat u_c as latent, using the synthetic objective only for evaluation (a realism-oriented regime).

2. *Counterfactual-label dataset:* we resample L from the Bradley–Terry model $\mathbb{P}(L = 1 \mid x, y, y', c) = \sigma(u_c(x, y) - u_c(x, y'))$ (a ground-truth regime).

The counterfactual regime is particularly useful for stress-testing overlap and proxy routing, because we can hold u_c fixed while perturbing p_c or the proxy channel $Q(\hat{c} \mid c, x)$.

Multi-domain preference logs and endogenous prompt choice. To emulate deployment, we extend beyond static datasets and create *multi-domain preference logs* with two feedback channels: (i) pairwise comparisons on model-sampled candidates (the standard DPO interface), and (ii) implicit engagement signals that correlate with u_c but also with spurious prompt features (a source of confounding). We operationalize endogenous p_c in two ways. First, we build domain-conditioned prompt pools and let each type draw prompts from a type-specific mixture, $p_c = \sum_d \eta_{c,d}\, p_d$, where p_d is a domain distribution and $\eta_{c,d}$ controls specialization. Second, in an interactive setting, we allow the *next prompt* to depend on the previous response via a simple user simulator (e.g., if the response is too verbose, the user asks for a shorter version; if the response is unhelpful, the user re-prompts with more detail). This endogenizes data in the minimal sense needed to observe how policies can induce distribution shift through their own outputs.

Training protocols and baselines. We compare (a) pooled DPO (single $\pi_\theta(y \mid x)$), (b) type-conditional DPO ($\pi_\theta(y \mid x, c)$ with observed c), (c) proxy-routed conditional DPO ($\pi_\theta(y \mid x, \hat{c})$), and (d) MoE implementations $\pi_{\theta,\phi}(y \mid x, \hat{c}) = \sum_k q_\phi(k \mid x, \hat{c})\pi_{\theta_k}(y \mid x)$ with both hard and soft gates. Across all conditions we fix a common reference π_{ref} and sweep β_c (either shared or type-specific) to trace the helpfulness–robustness frontier. For constraints, we implement harm caps either via explicit Lagrange multiplier updates (dual ascent with clipping) or via penalty surrogates calibrated to match target caps on a held-out audit set. Crucially, feasibility is evaluated on *deployed* mixtures when routing is noisy, i.e., under $\pi_\theta^{\text{dep}}(y \mid x, c) = \sum_{\hat{c}} Q(\hat{c} \mid c, x)\pi_\theta(y \mid x, \hat{c})$.

Stress tests: prompt–type shift, overlap degradation, and artifacts. We then evaluate under targeted distribution shifts designed to isolate the failure modes suggested by our formalism.

1. *Prompt–type shift:* hold u_c fixed but change p_c at test time (e.g., rotate the domain mixture weights $\eta_{c,d}$, or introduce novel prompt templates). This probes whether the learned policy relies on spurious correlates of type that do not transport across prompts.

2. *Overlap degradation:* explicitly reduce support overlap in a causal feature space $z = g(x, y)$ by filtering training examples so that some regions of z are absent for some types, while remaining present at test time. We can parameterize severity by an empirical density-ratio estimate of $\kappa \approx \max_{c,c'} \sup_z \hat{p}_c(z)/\hat{p}_{c'}(z)$, and sweep κ to test whether regret grows in the predicted manner.

3. *Length/format artifacts:* inject confounders that labelers systematically prefer (e.g., longer answers, more hedging, bullet lists) while hold-

ing task success constant, and then flip these correlations at test time. This checks whether KL-regularization and representation choices actually suppress artifact exploitation.

4. *Proxy noise and misrouting:* vary a controlled confusion channel $Q(\hat{c} \mid c, x)$, including both x-independent noise (simple ϵ-misclassification) and x-dependent noise (hard cases concentrated in particular domains). This tests the continuity requirement implicit in $O(\epsilon)$ welfare degradation claims.

5. *Strategic misreporting:* simulate a reporting game where users select $\tilde{c}$ based on observed policy behavior. Empirically, we implement this by allowing a user simulator to query multiple routes on the same x and pick the route with higher realized utility. We then re-estimate harms under the induced equilibrium route distribution, measuring whether any route becomes an "escape hatch" that breaks subgroup caps.

Metrics: welfare, constraints, and conservatism. Our primary outcome is *subgroup win rate*: for each c, we compare $\pi_\theta(\cdot \mid x, \text{route})$ against a baseline policy (pooled or π_{ref}) using held-out pairwise tests drawn from p_c, reporting $\mathbb{P}(\text{win} \mid c)$. When synthetic u_c is available, we also report estimated welfare $W_c(\pi)$ and welfare regret relative to an oracle computed by sampling from the closed-form tilt with known u_c and h_m. Safety is measured by empirical harm cap violations: for each (m, c) we estimate

$$\widehat{H}_{m,c}(\pi) \;:=\; \mathbb{E}_{x\sim\widehat{p}_c}\, \mathbb{E}_{y\sim\pi(\cdot|x,\text{dep}(c))}[h_m(x,y)],$$

and report both absolute violation $\max\{0, \widehat{H}_{m,c}(\pi) - \delta_{m,c}\}$ and tail risk (e.g., conditional-on-high-risk prompts). Finally, we track *KL drift* per type, $\text{KL}_c(\pi_\theta \| \pi_{\text{ref}})$, as a diagnostic for over-optimization and as a common currency for comparing personalization gains across architectures.

What we learn from the suite. This empirical plan is designed to separate three questions that are often conflated in practice: (i) does conditioning (or modularization) increase subgroup welfare on in-distribution prompts; (ii) does it remain robust under prompt shift and reduced overlap; and (iii) do proxy routing and strategic behavior convert nominally safe per-route policies into unsafe deployed mixtures. By sweeping β_c, caps $\delta_{m,c}$, overlap severity κ, and proxy error ϵ, we can map where the clean comparative statics survive and where they fail, thereby informing the discussion section's governance-relevant claims about auditing, logging, and the circumstances under which personalization is worth its added surface area.

Discussion and policy implications. Our formalism makes a simple but operational point: once we allow user objectives to differ, the platform is no longer choosing *a* single helpfulness policy, but rather a bundle of type-conditional policies plus a routing mechanism. This bundle can strictly improve welfare relative to pooling when genuine preference heterogeneity is present, but it also enlarges the safety and governance surface area. The correct policy question is therefore not "personalize or not," but "under what evidentiary and monitoring requirements does personalization remain both welfare-improving and constraint-feasible under deployment-realistic routing and distribution shift?"

Audit and logging as first-class safety requirements. Personalization breaks many common evaluation shortcuts because the deployed behavior depends on the joint distribution of prompts, routes, and outputs. As a result, the minimal audit trail must include, for each interaction, the prompt x, the chosen route (observed c or proxy $\hat{c}$), the model output y, and enough auxiliary metadata to re-estimate harms and KL drift over time. In our notation, the relevant safety object is not $H_{m,c}(\pi_\theta(\cdot \mid x, \hat{c}))$ evaluated route-by-route in isolation, but the *deployed mixture* induced by the proxy channel,

$$\pi_\theta^{\mathrm{dep}}(y \mid x, c) \;:=\; \sum_{\hat{c}} Q(\hat{c} \mid c, x)\, \pi_\theta(y \mid x, \hat{c}),$$

and audits should target $H_{m,c}(\pi_\theta^{\mathrm{dep}})$ rather than the easier per-route quantities. Concretely, this implies logging the proxy model version (to track changes in Q), the gate confidence (to detect "hard cases" where Q is brittle), and periodic replay or counterfactual evaluation on a fixed audit set to isolate policy drift from user-population drift. If regulators impose caps $\delta_{m,c}$, then regulators (or independent auditors) need sufficient access to estimate $\widehat{H}_{m,c}$ with confidence intervals, including tail-risk slices (e.g., conditional on high-risk prompt templates) rather than only unconditional averages.

Overlap and confounding require representation-aware audits. Because our generalization claims hinge on overlap in a causal feature space $z = g(x, y)$, logging requirements should anticipate the need to diagnose overlap degradation. In practice, we rarely know the true causal g, but we can approximate it via model-based embeddings or task-relevant feature extractors and track empirical density ratios across types. A governance-relevant implication is that "we satisfied the cap last quarter" is not itself stable evidence of future feasibility: if the support of z shifts for a subgroup (e.g., new prompt genres appear), then the effective overlap constant κ can worsen, and the same policy can become both less useful and less safe. We therefore view *overlap monitoring*—density-ratio alarms, coverage tests on z, and periodic targeted data collection for under-covered regions—as analogous to distribution-shift monitoring in other safety-critical ML deployments,

but with higher stakes because failures are type-correlated and can look like disparate impact.

When is personalization necessary (or justified)? Personalization is most justified when three conditions jointly hold. First, there is evidence of *genuine* preference heterogeneity: not merely different prompt frequencies, but sign-disagreements on response pairs after controlling for harms. Second, routing is sufficiently reliable (small effective ϵ) or the policy family is smooth enough in c that misrouting costs are bounded. Third, the type-conditional prompt distributions have adequate overlap in the relevant feature regions, so that learning does not devolve into fitting spurious prompt–type correlates. When these conditions fail, pooling (or weaker forms of adaptation, such as soft style knobs that do not alter high-risk behavior) may dominate on robustness grounds even if it leaves welfare on the table. A pragmatic decision rule is to treat personalization as an *intervention* that must earn its keep: we should require that, at a fixed conservatism budget (e.g., a KL cap relative to π_{ref}), personalization yields a clear subgroup welfare gain *and* does not increase the worst-case harm risk under plausible routing noise and prompt shifts.

Avoiding "escape hatches" and route-induced safety regressions. A recurring failure mode in personalized systems is the emergence of a route that becomes an implicit jailbreak: users (or user simulators) learn that selecting a particular $\hat{c}$ yields more permissive behavior, even if that route was intended for benign preferences. Our framework makes this legible: if constraints are enforced only conditionally on reported type, then strategic selection can change the effective distribution over routes and break the intended subgroup caps. One policy implication is that cap enforcement should include *mixture-robust* constraints (audited under the empirically observed routing equilibrium), and in high-risk domains may need *cross-type* or *global* caps that cannot be circumvented by misreporting. Where misreporting is plausible, platforms should treat the routing interface itself as a safety-critical component: rate limits on route switching, friction for high-privilege routes, and anomaly detection for unusual route/query patterns become part of the safety case rather than mere product concerns.

Setting β_c: conservatism as an auditable control knob. The temperature β_c plays two roles that matter for governance. Statistically, larger β_c reduces over-optimization on noisy or confounded preferences by keeping π_θ closer to π_{ref}; operationally, it provides a common currency for comparing personalization variants under a shared "amount of change." We therefore recommend treating β_c not as a fixed hyperparameter chosen once, but as an *audited control knob* with a documented rationale. In deployments where

type labels are noisy or overlap is poor, raising β_c for the affected routes is a principled way to bound regret and contain tail risks, albeit at a welfare cost. Conversely, when a route is well-instrumented and well-covered, lowering β_c can be justified if accompanied by stronger monitoring and tighter harm multipliers. A practical governance artifact is a per-route "KL budget" (or upper confidence bound on $\mathrm{KL}_c(\pi_\theta \| \pi_{\mathrm{ref}})$) that triggers review when exceeded.

Setting harm caps $\delta_{m,c}$ and interpreting Lagrange multipliers. Caps $\delta_{m,c}$ are ultimately normative choices, but our Lagrangian perspective helps interpret what the system is doing when caps bind: the multipliers $\lambda_{m,c}$ become implicit harm prices that trade off utility against safety. This suggests two implementation-facing recommendations. First, cap-setting should be accompanied by *multiplier monitoring*: sudden growth in $\lambda_{m,c}$ often indicates either an infeasible cap given the current model class and data, or a shift in prompts such that previously rare high-harm regions are now common. Second, when routing is imperfect, caps should be tightened to preserve feasibility under misclassification, effectively budgeting for the $O(\epsilon)$ leakage of unsafe behavior into subgroups where it is unacceptable. More broadly, we should distinguish between *measured* harms (those captured by h_m) and *unmeasured* hazards (specification gaps); tighter caps can only compensate for the former, while the latter require red-teaming, policy updates, and sometimes restricting personalization entirely in certain domains.

Limitations: what the model abstracts away. Several assumptions are doing real work. The Bradley–Terry preference model is convenient, but real labelers exhibit context effects, rater drift, and multi-attribute tradeoffs that violate the logistic difference structure. The feature-space overlap condition is both critical and hard to verify: without a defensible $g(x, y)$, we can at best approximate overlap and hope it correlates with true transportability. Endogenous prompt choice is only partially modeled; in reality, users adapt strategically to the system over long horizons, and the platform's policy updates feed back into p_c. Finally, groupwise caps presuppose that harms are measurable and that subgroup definitions are stable; both can fail in practice (e.g., latent vulnerable populations, or harms that manifest downstream rather than in single-turn outputs).

Open questions for alignment and governance. Three research directions appear especially urgent. (i) *Dynamic equilibrium*: we need models where p_c evolves with π_θ, so that "safe at time t" does not imply safe after users learn the system. (ii) *Incentive-compatible routing*: rather than assuming an exogenous proxy $\hat{c}$, we need mechanisms that elicit types (or preferences) while limiting the gains from misreporting and preserving pri-

vacy. (iii) *Robust constraint satisfaction*: enforcing $H_{m,c}(\pi^{\text{dep}}) \leq \delta_{m,c}$ under distribution shift, imperfect detectors, and adaptive adversaries likely requires worst-case or distributionally robust formulations, and may force more conservative choices of β_c than those suggested by in-sample evaluations. Addressing these questions would turn our present "static" safety case into one that better matches the realities of deployment.

Bottom line. Personalization can be a Pareto improvement in welfare terms, but only when supported by strong instrumentation: mixture-aware audits, overlap monitoring, and explicit controls for conservatism and constraint enforcement under routing noise. From a policy perspective, the key shift is to regulate and evaluate *systems* (policy plus router plus monitoring) rather than single models, because the relevant safety object is the deployed mixture interacting with endogenous users.