

# Vector-Lagrangian Safe RLHF: Multi-Category Risk Budgets and Shadow Prices for LLM Safety Governance

Liz Lemma Future Detective

January 22, 2026

## Abstract

Modern alignment methods (Constitutional AI and Safe RLHF) operationalize safety using a small set of principles or a single learned ‘cost’ model, then trade off safety and performance via RLHF-style optimization. In 2026 deployment regimes, however, regulation and liability are intrinsically multi-dimensional: self-harm, hate, fraud, privacy, bio/weaponization, and terrorism face separate standards, enforcement, and social costs. This paper argues that treating safety as a single scalar cost is economically and operationally mis-specified, and proposes Vector-Lagrangian Safe RLHF: a multi-constraint formulation where each harm category is a separate constraint with its own endogenously determined shadow price  $\lambda_i$ . We develop a clean convex policy model that yields closed-form KKT characterizations and interprets  $\lambda$  as category-specific ‘risk budget prices’. We prove a risk-budget allocation theorem and an impossibility/inefficiency result for scalarized safety objectives: fixed-weight scalar aggregation induces cross-category substitution (category leakage) and cannot generally implement category-wise compliance. We outline empirical tests using a multi-head cost model aligned to a harm taxonomy (e.g., 14 categories as in Safe RLHF) and targeted red-teaming that demonstrates reduced leakage and improved tail-risk control relative to scalar cost training.

## Table of Contents

1. 1. Introduction: why single-score safety breaks under 2026 multi-standard governance; motivate ‘risk budgets’ and category leakage; contributions and roadmap.
2. 2. Related work: Constitutional AI principles as implicit multi-axes; Safe RLHF as single-constraint CMDP; safe RL multi-constraint CMDPs; Goodharting/over-optimization and evaluation design.

3. 3. A minimal model of LLM deployment: prompts, responses, reward (helpfulness), and category costs; define compliance as category-wise constraints; define leakage/substitution formally.
4. 4. Vector-constrained optimization and shadow prices: primal program, dual, KKT; interpret  $\lambda_i$  as category shadow prices; envelope theorem identification.
5. 5. Risk-Budget Allocation Theorem: characterization and (where possible) closed-form policy structure (e.g., per-prompt softmax with linear penalties); uniqueness and stability conditions.
6. 6. Why scalar safety fails: impossibility/inefficiency of fixed-weight scalarization; explicit counterexamples; conditions under which scalarization works (knife-edge).
7. 7. Comparative statics and governance: how  $b$ , prompt mix, and capacity shift  $\lambda$ , reward, and category costs; regulatory interpretation (tightening one budget shifts pressure to others).
8. 8. Empirical design sketch (no full experiments here): multi-head cost model, vector-Lagrangian updates, and targeted red teaming to measure leakage; evaluation protocol and metrics.
9. 9. Discussion: audit reporting, differentiated liability, and practical deployment recommendations; limitations and extensions (multi-turn, robust/adversarial prompt distributions).
10. 10. Conclusion: takeaways for alignment as economic institution; open questions.

## 1 Introduction

In current deployments, we rarely have the luxury of treating “safety” as a single axis. By 2026, frontier model providers are typically accountable to multiple, partially independent standards: internal policies and constitutional guidelines; platform integrity rules; sectoral obligations around privacy, consumer protection, and non-discrimination; and domain-specific restrictions (e.g., dual-use biology, cybersecurity, financial wrongdoing). These regimes do not collapse cleanly into one number. They are enforced by different stakeholders, measured with different instruments, and—crucially—they often bind in different parts of the input distribution. When we optimize a model as if safety were a single score, we implicitly assume we can trade one kind of harm against another at a fixed exchange rate. That assumption is operationally brittle: regulators and platforms do not, in general, accept such exchange, and adversaries actively seek prompts that exploit mismatched tradeoffs.

We therefore start from a simple observation: most deployed “single-metric” safety tuning pipelines behave like scalar optimization. We take a base reward for helpfulness, subtract a weighted penalty for unsafe behavior, and tune until aggregate evaluation looks acceptable. This works passably when the world is dominated by one safety concern and measurement is stable. It breaks when there are multiple constrained risks, each with its own threshold and audit process. The failure mode is not merely philosophical; it is geometric. Optimizing a scalar objective selects a single supporting hyperplane of the achievable set of (reward, harms). But compliance is defined by a *box* of constraints—one threshold per category—and the identity of the binding constraint depends on the budget vector, the prompt mix, and the available behavioral repertoire of the model. Fixing weights in advance hard-codes an exchange rate between categories, even though the “right” exchange rate is endogenous to the constraint set and shifts as governance priorities (and distributions of use) change.

This mismatch manifests as what we will call *category leakage*. Suppose a developer tightens a filter for one harm type, say fraud facilitation. Under a scalarized training signal, the easiest way to improve the aggregate score may be to shift the model toward behaviors that are still penalized—but less so—such as providing borderline privacy-violating guidance, producing more abrasive content, or offering speculative claims that skirt misinformation thresholds. Even if the aggregate scalar score improves, the model may newly violate a different standard. In practice, this is exactly the scenario auditors and red teams report: targeted mitigations reduce the measured incidence of a focal harm while increasing failure rates on neighboring dimensions, often in ways that are less visible under the current evaluation suite. From a governance standpoint, leakage is not an edge case; it is the expected response of any optimizer confronted with incomplete or misweighted

objectives.

We propose to formalize multi-standard safety as *risk budgets*. Rather than asking the model to maximize a single safety score, we treat each harm category as a separate constraint with an explicit threshold. The developer then solves a constrained optimization problem: maximize expected helpfulness subject to satisfying each category budget. The Lagrange multipliers of this program have a direct interpretation as *shadow prices* for risk: they quantify, at the optimum, how much helpfulness we must sacrifice to reduce expected harm in a particular category. This provides a principled replacement for ad hoc penalty weights. Instead of tuning a fixed vector of weights once and hoping it remains valid across deployments, we allow the dual variables to adapt to the binding constraints implied by the current budget vector and prompt distribution. In other words, the “exchange rates” between harms are not assumed; they are *learned* as part of satisfying the constraints.

This perspective is motivated by how compliance is actually verified. External governance rarely inspects an internal scalar objective; it inspects category-specific rates (or tail risks) under specified evaluations. Auditors produce labels and measurements per category, sometimes with noisy instrumentation and changing taxonomies. Platforms maintain separate enforcement tracks (e.g., privacy incidents versus self-harm facilitation). Even within a single organization, different teams own different risk registers. A risk-budget formulation mirrors this reality: it aligns the optimization target with how the world measures failure. It also makes explicit where we are making assumptions. In our baseline model we treat category costs as known functions and prompts as drawn i.i.d. from a fixed distribution; in deployments, both are learned and adversarially stress-tested. The value of starting with the clean constrained program is that it reveals the structure of the tradeoffs that any practical training and evaluation pipeline must manage.

Our first contribution is conceptual: we articulate why single-score safety is structurally misaligned with multi-standard governance, and we name the resulting substitution behavior as category leakage. Importantly, leakage does not require malicious intent or distribution shift. It arises under good-faith optimization whenever the set of feasible behaviors contains non-collinear harm tradeoffs across categories. This is the typical case: there are many ways for a model to be unhelpful or harmful, and improvements along one axis often open room for regressions elsewhere.

Our second contribution is formal: we develop a finite, convex model of a stochastic policy over prompts and responses with multiple harm constraints. In this setting, the constrained optimum admits a Lagrangian characterization with a vector of nonnegative multipliers. Under standard regularity, these multipliers are unique and can be interpreted as category-specific “prices” that implement the constrained solution. This yields a clean

language for discussing compliance and incentives: budgets are policy parameters chosen by a regulator or governance body, while multipliers summarize the marginal cost of compliance and predict how the optimal behavior changes as standards tighten.

Our third contribution is diagnostic: we show that naive scalarization with fixed weights cannot, in general, implement the constrained optimum across varying budget vectors. This is not a critique of scalar penalties as a heuristic; it is a statement about impossibility in the presence of multiple independent constraints. Fixed weights may coincidentally work for a particular governance regime and prompt mix, but there exist nearby regimes where the same weights either violate some constraint or choose a dominated policy. This provides a theoretical grounding for a common operational pain point: the endless retuning of safety weights as evaluation suites evolve.

Our fourth contribution is methodological: we connect the dual variables to practical training and monitoring procedures. If per-category costs are estimated by learned classifiers or preference models, then the multipliers naturally suggest a primal-dual loop: update the policy to improve helpfulness net of the current risk prices, and update the risk prices when audits detect budget violations. This suggests a governance-relevant interface: regulators or internal safety committees can adjust budgets  $b_i$ , and developers can report the implied shadow prices  $\lambda_i$  as a quantitative measure of how tight each constraint is. While this does not solve measurement and gaming on its own, it does create a transparent coupling between standards, optimization, and observed tradeoffs.

Finally, we emphasize the safety implications and limitations that motivate the rest of the paper. First, multi-constraint optimization reduces one Goodhart channel (hiding tradeoffs behind a scalar), but it does not eliminate Goodharting: if the cost models are misspecified, the policy will optimize against the proxy, and leakage may occur into unmeasured sub-categories or distributional tails. Second, the prompt distribution is not exogenous in adversarial settings; attackers can shift mass toward high-risk contexts, effectively tightening budgets and increasing shadow prices. Third, the clean convex picture is a benchmark: real training is nonconvex, costs are noisy, and constraints may be better represented as tail probabilities rather than expectations. We treat these as extensions rather than reasons to avoid formalization; without a baseline, it is difficult to state precisely what is failing and what needs to be audited.

Roadmap: Section 2 situates our approach relative to Constitutional AI and Safe RLHF, and to the literature on constrained and multi-objective reinforcement learning, as well as work on Goodhart effects and evaluation design. Section 3 introduces the formal model and derives the KKT and dual interpretations of risk budgets. Section 4 analyzes why fixed-weight scalarization fails and characterizes leakage under tightening standards. Section 5 discusses implications for training loops, auditing protocols, and reporting,

including how shadow prices can serve as a governance-facing summary statistic. We conclude with open problems around robust budgets under distribution shift, tail-risk constraints, and incentive-compatible auditing.

## 2 Related Work

Our starting point—that deployed systems face multiple, partially independent safety standards—sits at the intersection of three literatures that are often discussed separately: (i) principle-based alignment methods (notably Constitutional AI) that encode multiple normative desiderata, (ii) optimization-based alignment methods (RLHF, Safe RLHF) that usually train against a small number of learned objectives, and (iii) work on safe and multi-objective decision-making in reinforcement learning and operations research, including constrained Markov decision processes (CMDPs), multi-constraint control, and risk-sensitive optimization. A fourth thread, increasingly central in practice, concerns Goodhart effects and evaluation design: how measurement, auditing, and adaptive optimization interact to produce overfitting and substitution into unmeasured failure modes.

Constitutional AI makes the multi-axis nature of safety especially explicit. In its canonical form, a developer specifies a *constitution*—a list of principles such as “avoid harassment,” “respect privacy,” “do not facilitate wrongdoing,” and “be helpful and honest”—and trains the model to critique and revise its own outputs with respect to these principles, sometimes followed by preference optimization using synthetic or human comparisons <sup>7</sup>. Operationally, the constitution is a structured object that enumerates multiple constraints or desiderata, and the training pipeline attempts to produce behavior that satisfies them across diverse contexts. This is close in spirit to a risk-register view of governance: different harms are tracked separately, and compliance is assessed in a category-wise manner. However, in many implementations the multi-principle structure is eventually collapsed into a smaller number of training signals (e.g., a scalar preference model, or a single reward function combining helpfulness and harmlessness), and the degree of permissible tradeoff between principles is implicit in the data generation and aggregation scheme. Our framing can be read as making this implicit exchange rate explicit: even if the normative object is a list of principles, the optimizer must still confront the question of how to arbitrate conflicts, and that arbitration is effectively governed by the dual variables (or by whatever surrogate weights the pipeline induces).

A related line of work uses constitutions, policies, or taxonomies to *generate* labels for multiple categories (e.g., “self-harm,” “sexual content,” “privacy,” “hate”) and then trains separate classifiers, critics, or reward models per category <sup>7</sup>. In deployment, these tools often appear as a stack: a base model, a set of safety classifiers or “guards” gating outputs, and a policy

model trained to avoid triggering the guards. The stack is inherently multi-headed, but the training of the policy layer frequently reduces to scalar penalties, a cascade of hard filters, or a prioritized ordering of constraints. While these heuristics can work well, they tend to obscure the underlying substitution incentives: when one guard becomes stricter, the policy may shift toward outputs that are acceptable under that guard but more likely to trigger another. This is one reason we emphasize a unified constrained optimization picture rather than a purely procedural description of guardrails.

Safe RLHF and closely related methods provide a more explicitly optimization-centric entry point. Classical RLHF pipelines learn a reward model from preference data and then optimize a policy via reinforcement learning with regularization to a reference model (often implemented as a KL penalty) ?. “Safety” can enter this process in at least three ways: (i) by shaping the preference data to disfavor unsafe outputs, thereby baking safety into a single reward model; (ii) by introducing explicit refusal or harmlessness rewards as additional terms in a scalar objective; or (iii) by treating safety as a *cost* and imposing a constraint or penalty during policy optimization ???. The third approach aligns most directly with the CMDP formalism: maximize expected utility subject to an expected cost constraint. Much of the practical algorithmic toolkit (Lagrangian relaxation, primal–dual updates, clipping, and trust-region constraints) is inherited from this literature, even when papers describe the system in RLHF-specific language.

At the same time, much of the Safe RLHF discussion is implicitly *single-constraint*: one defines a safety cost (sometimes a composite score) and constrains or penalizes it. This is a natural first step because it is simpler to implement and evaluate, and because many early safety interventions targeted a dominant category (e.g., toxicity). But when there are multiple categories with distinct audit thresholds, the single-cost abstraction becomes brittle. A composite safety score is itself a scalarization choice; it fixes an exchange rate across harms and invites the optimizer to “spend” risk in underweighted categories. This observation parallels results in multi-objective optimization: weighted sums can recover Pareto-optimal points only under restrictive conditions and generally fail to represent non-convex parts of the frontier or policy-dependent constraint sets. In other words, the scalar objective is not merely an engineering simplification; it can change which tradeoffs are implementable as governance requirements change.

The broader safe reinforcement learning literature has long studied CMDPs and constrained control ???. In a CMDP, an agent maximizes reward subject to one or more expected cost constraints; Lagrangian methods convert constraints into penalties with multipliers updated by dual ascent. In tabular or convex settings, strong duality and convergence guarantees are available, while in large-scale function approximation settings the guarantees are weaker and the practical focus shifts to stability, constraint satisfaction under estimation error, and conservative updates. Two aspects are particularly

relevant for LLM alignment. First, CMDP methods provide a principled interpretation of penalty weights as *dual variables* rather than arbitrary hyperparameters, which motivates treating “safety weights” as endogenous and audit-driven. Second, the multi-constraint CMDP literature explicitly recognizes that different constraints can bind in different regions of state space and can interact in unintuitive ways (substitution, complementarity, and constraint cycling), which resembles the empirical phenomenon of targeted mitigations shifting failure mass across categories.

Multi-constraint CMDPs and vector-valued costs have also been studied under the umbrella of multi-objective reinforcement learning and constrained optimization [??](#). Here, a key methodological distinction is between (a) seeking a Pareto frontier (treating the problem as inherently multi-objective) and (b) implementing externally specified constraints or budgets (treating some objectives as hard requirements). LLM safety in regulated deployments is typically of the second type: the relevant question is not “which point on the frontier do we like,” but “can we meet these category-wise thresholds while remaining useful?” This distinction matters because many multi-objective methods assume the designer is free to choose tradeoffs, whereas governance often dictates them. Our emphasis on budgets and shadow prices is intended to bridge that gap: budgets encode externally imposed requirements, while multipliers summarize the induced tradeoffs at the optimum.

A complementary thread concerns *risk-sensitive* constraints: rather than constraining an expected cost, one constrains tail probabilities, quantiles, or coherent risk measures such as CVaR [?](#). In safety applications, this is often the more faithful representation of stakeholder concerns: auditors care about rare catastrophic failures, not just average rates. Recent work in safe RL and robust optimization explores chance constraints, distributionally robust objectives, and adversarially chosen environments. For LLMs, the analogue is evaluation under red-teaming and distribution shift: prompt distributions are not fixed, and the relevant constraint may involve worst-case or stress-test performance. Our baseline model focuses on expected costs for tractability, but the related literature clarifies where this abstraction is likely to break and suggests natural extensions (e.g., robust budgets, tail-risk costs, or adversarial  $D$ ).

Goodhart effects and evaluation design form the final pillar of related work. The core observation is that once a proxy metric becomes a target, optimization pressure induces both *overfitting* to the measurement process and *substitution* into unmeasured channels [?](#). In LLM alignment, this shows up as reward hacking, jailbreak susceptibility, and benchmark saturation: a model can learn to satisfy a particular classifier or rubric without improving the underlying property, and improvements on one eval suite can coincide with regressions on another. Importantly, Goodharting interacts with multi-category governance in a specific way: even if each category has a reasonable proxy, optimizing a scalar aggregate of proxies incentivizes reallocation of er-

ror mass to whichever proxy is easiest to evade or least weighted. This is distinct from (though compatible with) classic over-optimization within a single metric. It suggests that the evaluation problem is not only to build better per-category measurements, but also to design training and monitoring loops that respect the *vector* nature of the constraints.

Recent practical work on auditing, red teaming, and systematic evaluation frameworks reinforces this point. External evaluations are typically reported as a table of category-wise rates, sometimes with confidence intervals and scenario breakdowns, rather than a single scalar score. Governance mechanisms (platform policies, sectoral regulation, internal risk committees) similarly operate with separate risk registers and escalation paths. This institutional fact motivates treating category thresholds as first-class objects in the optimization problem. It also motivates transparency tools that report how tight each constraint is in marginal terms (e.g., “how much helpfulness are we giving up to meet privacy budgets?”) rather than only reporting pass/fail outcomes.

In sum, existing alignment pipelines already contain the ingredients of a multi-constraint perspective—multiple principles, multiple evaluators, multiple harms, and iterative tuning. What is often missing is a minimal formal model that makes the implied exchange rates and substitution incentives explicit, and that cleanly separates (i) externally set standards (budgets), (ii) developer optimization (policy choice), and (iii) measurement and adaptation (audits and updates). We develop such a model next, using a finite prompt–response formulation that is deliberately simple but rich enough to capture the central phenomenon: category leakage induced by scalarization and mitigated by vector-valued constraints and their associated shadow prices.

### 3 A Minimal Model of LLM Deployment: Prompts, Responses, Rewards, and Category Costs

We model deployment as a repeated interaction between an environment that generates *prompts* and a system that returns *responses*. The purpose of the model is not to capture the full complexity of language, but to isolate a structural feature of regulated settings: safety is assessed along multiple partially independent dimensions, each with its own threshold, and optimization pressure can move failures across dimensions rather than eliminating them. To make this interaction explicit, we start with a finite prompt–response abstraction that can be viewed as a discretization of a continuous process (e.g., prompt clusters, scenario templates, or an evaluation suite with representative contexts).

Let  $\mathcal{X}$  be a finite set of prompts/contexts and  $D$  a distribution over  $\mathcal{X}$ . A realized prompt  $x \sim D$  may represent an end-user query, a tool-augmented

context window, or a red-team scenario. The model produces a response  $y \in \mathcal{Y}$ , where  $\mathcal{Y}$  is a finite set of available actions (e.g., answer styles, refusal modes, or response candidates after decoding). The developer chooses a (possibly stochastic) policy  $\pi \in \Pi$ , where

$$\Pi := \{\pi : \pi(\cdot | x) \in \Delta(\mathcal{Y}) \quad \forall x \in \mathcal{X}\},$$

and deployment draws  $y \sim \pi(\cdot | x)$ . This stochasticity is not essential but is useful both technically (convexity) and descriptively (real systems randomize via sampling, beam search ties, or non-deterministic tool calls).

We distinguish two kinds of outcomes. First, the system generates *helpfulness* (or more broadly, task utility), represented by a bounded function  $r : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ . In regulated deployments,  $r$  should be read as the value of providing correct, relevant, and appropriately calibrated assistance, net of generic quality concerns (verbosity, latency, etc.). Second, the system incurs *category-specific harms*. We index harm categories by  $i \in \{1, \dots, m\}$  and write  $c_i : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  for the contribution of response  $y$  to harm category  $i$  when prompted with  $x$ . Categories may correspond to policy-relevant buckets such as privacy leakage, facilitation of wrongdoing, self-harm encouragement, harassment/hate, regulated advice, or misinformation. While costs can in principle be signed, we will interpret  $c_i$  as nonnegative or bounded below, consistent with an auditing pipeline that flags and scores violations.

Given a policy  $\pi$ , the induced expected reward and expected category costs are

$$R(\pi) := \mathbb{E}_{x \sim D, y \sim \pi(\cdot | x)}[r(x, y)], \quad C_i(\pi) := \mathbb{E}_{x \sim D, y \sim \pi(\cdot | x)}[c_i(x, y)].$$

The regulator (or an internal risk committee acting as a proxy) specifies *budgets*  $b \in \mathbb{R}_+^m$ , where  $b_i$  is the maximum permitted expected cost in category  $i$  over the relevant prompt distribution. We say that a policy  $\pi$  is *compliant* if it satisfies the category-wise constraints

$$C_i(\pi) \leq b_i \quad \forall i \in \{1, \dots, m\}.$$

This definition matches how many deployments are actually governed: safety reporting and external audits are typically presented as a table of per-category rates or severities, and failure in one category is not offset by success in another. Importantly, compliance is a property of the *policy* under the *prompt distribution*  $D$ : changing the user population, adding a new product surface, or inducing attacker adaptation effectively changes the distribution being averaged over, even if the underlying model weights are unchanged.

The developer-facing decision problem is then to choose a policy  $\pi$  that is as helpful as possible while meeting all budgets. Conceptually, we can interpret  $\pi$  as the result of a training pipeline: data collection and filtering, supervised fine-tuning, RLHF (or direct preference optimization), system

prompts and tool policies, and post-training guardrails. Our abstraction collapses these degrees of freedom into a single object—a mapping from prompts to distributions over responses—because what matters for governance is the induced joint distribution over  $(x, y)$ , and hence over  $(r, c_1, \dots, c_m)$ .

Two modeling choices deserve emphasis. First, we treat  $\mathcal{X}$  and  $\mathcal{Y}$  as finite. This ensures that  $R(\pi)$  and  $C_i(\pi)$  are linear in  $\pi$ , and that the feasible set is a product of simplices. In practice, one can regard  $\mathcal{X}$  as a finite set of prompt types (including adversarial types), and  $\mathcal{Y}$  as a finite set of response modes (e.g., “comply with high detail,” “comply with low detail,” “safe completion,” “refuse with resources,” etc.). Second, we take the functions  $r$  and  $c_i$  as primitives. Empirically, these correspond to reward models, safety classifiers, human labels, and audit procedures. The gap between the true latent harms and the measured costs is a central source of Goodhart effects, but the present goal is to isolate a different issue: even if each  $c_i$  were perfectly measured, *optimizing the wrong aggregation of them* can still induce predictable substitution failures.

**Heterogeneity across prompts and the meaning of budgets.** A useful mental model is that each category cost is highly *context-dependent*. For many prompts  $x$ , the privacy cost  $c_{\text{priv}}(x, y)$  is essentially zero for all reasonable  $y$ ; for a small subset (e.g., “summarize this medical note”), privacy risk dominates. Similarly, self-harm costs are concentrated on a narrow slice of the prompt space, while harassment costs concentrate on identity-referent prompts. Therefore, an expected constraint  $C_i(\pi) \leq b_i$  is not merely a global rate limit; it is an *allocation rule* for how much risk the policy is allowed to take on the subset of contexts where category  $i$  is salient. This observation foreshadows why category interaction matters: if two categories become salient on overlapping but not identical subsets of  $\mathcal{X}$ , then a mitigation aimed at one subset can move probability mass of problematic behavior to another subset.

Budgets also implicitly encode a *measurement granularity*. If auditors measure  $c_i$  only on certain scenarios or with certain detectors, then  $C_i(\pi)$  is effectively an expectation over the audited distribution rather than the true deployment distribution. Our baseline takes  $D$  as given to keep the analysis crisp; later extensions can treat  $D$  as stress-tested or adversarially perturbed to model red-teaming and adaptive misuse.

**Formalizing substitution opportunities.** The key structural assumption behind “leakage” is that the action space offers multiple ways to achieve similar helpfulness with different harm profiles. Formally, for a given prompt  $x$ , consider two responses  $y, y' \in \mathcal{Y}$  such that

$$r(x, y) \approx r(x, y') \quad \text{but} \quad c(x, y) \neq \gamma c(x, y') \text{ for any scalar } \gamma > 0,$$

where  $c(x, y) = (c_1(x, y), \dots, c_m(x, y))$ . The non-collinearity condition says that the two responses induce different *mixes* of harms across categories. In deployment terms, the model might be able to answer a sensitive question in a way that is less likely to violate privacy but more likely to constitute regulated advice, or less likely to produce harassment but more likely to disclose personal data via over-specific examples. When such tradeoffs exist on a nontrivial set of prompts, there is no single “safe” action that dominates across all categories; meeting multiple budgets requires coordinated control rather than one-dimensional tuning.

This substitution structure is not an exotic edge case. It arises naturally whenever (i) categories are only partially overlapping (privacy vs. harassment vs. fraud), (ii) responses have multiple components (tone, specificity, refusal strategy, citations), and (iii) users can adversarially reframe requests. The finiteness of  $\mathcal{Y}$  does not remove this complexity; it simply forces us to represent it as a discrete menu of options.

**Defining leakage at the policy level.** We use “category leakage” to mean a policy response to pressure on one category that reallocates harm into other categories rather than reducing harm uniformly. One clean way to state this is via comparative statics over budgets. Fix a baseline budget vector  $b$  and let  $\pi^*(b)$  denote an optimal compliant policy (assuming existence and, where needed, uniqueness). Consider tightening category  $k$  by replacing  $b_k$  with  $b'_k < b_k$ , keeping other budgets fixed, yielding a new optimal policy  $\pi^*(b')$  where  $b' = (b_1, \dots, b'_k, \dots, b_m)$ . We say there is *leakage from  $k$  into  $j$*  (under this tightening) if

$$C_k(\pi^*(b')) \leq C_k(\pi^*(b)) \quad \text{and} \quad C_j(\pi^*(b')) > C_j(\pi^*(b))$$

for some  $j \neq k$ . The first inequality is the intended effect of tightening; the strict increase in  $C_j$  captures the unintended shift. In words: making the system safer with respect to one audited dimension can make it less safe with respect to another, even when both dimensions are measured and constrained.

This definition separates leakage from the trivial case where all costs move down together because the system becomes uniformly more conservative (e.g., always refuse). Leakage is specifically about *recomposition* of the cost vector under optimization pressure. It is also distinct from classic single-metric Goodharting: even if each  $c_i$  is perfectly measured, the system can still respond to a change in one budget by exploiting substitutability that increases another cost.

**Why scalar safety scores are structurally fragile.** A common engineering simplification is to collapse category costs into a scalar “safety score” and optimize a single penalized objective. In our notation, this corresponds

to choosing weights  $\nu \in \mathbb{R}_+^m$  and a penalty intensity  $\kappa > 0$ , then maximizing an objective of the form

$$\mathbb{E}[r(x, y)] - \kappa \mathbb{E}[\nu \cdot c(x, y)].$$

This construction fixes an exchange rate across harm categories: one unit of privacy risk is treated as  $\nu_{\text{priv}}/\nu_{\text{hate}}$  units of harassment risk, regardless of the regulator's budgets or the current operating point of the system. The fragility comes from the fact that budgets are *not* stable across deployments: they differ by jurisdiction, product surface, user population, and time, and they can tighten after incidents. If the system is trained against a fixed scalarization, it will generally not satisfy arbitrary category-wise thresholds without either excessive conservatism (sacrificing  $R(\pi)$  unnecessarily) or leakage into underweighted categories.

In the minimal model, this failure mode is driven entirely by geometry: the feasible set defined by  $C_i(\pi) \leq b_i$  is an intersection of halfspaces in the space of achievable cost vectors, while scalarization optimizes along a single supporting hyperplane determined by  $\nu$ . When the developer must satisfy a *vector* of constraints that can bind in different combinations, a single fixed hyperplane cannot, in general, implement the correct tradeoff across all  $b$ . The next section formalizes this observation using the primal–dual perspective and shows how category-specific shadow prices arise endogenously as the right notion of “exchange rates” at the optimum.

## 4 Vector-Constrained Optimization and Shadow Prices

The developer problem is a linear program over a product of simplices, but it is more illuminating to view it through the primal–dual lens. The reason is conceptual rather than computational: in multi-category regimes, the correct “exchange rates” across harms are not design parameters that we pick once, but endogenous objects that depend on which budgets bind in the current deployment. The dual variables make this dependence explicit and provide a clean bridge to training and governance procedures that update safety pressure category-by-category.

**Primal program and value function.** Fix a budget vector  $b \in \mathbb{R}_+^m$ . The developer chooses  $\pi \in \Pi$  to solve

$$\max_{\pi \in \Pi} R(\pi) \quad \text{s.t.} \quad C_i(\pi) \leq b_i \quad \forall i \in \{1, \dots, m\}. \quad (1)$$

Let  $V(b)$  denote the optimal value:

$$V(b) := \max_{\pi \in \Pi: C(\pi) \leq b} R(\pi).$$

Because  $\Pi$  is compact and  $R(\pi)$  is linear, an optimizer exists; because constraints are linear and we assumed strict feasibility (Slater), the feasible region has nonempty relative interior. The central question is how to characterize  $\pi^*(b)$  and how it moves as  $b$  changes. This is exactly the sort of comparative static problem for which Lagrange multipliers provide the right coordinates.

**Lagrangian and the dual problem.** Introduce multipliers  $\lambda \in \mathbb{R}_+^m$  and form the Lagrangian

$$\mathcal{L}(\pi, \lambda) := R(\pi) - \sum_{i=1}^m \lambda_i (C_i(\pi) - b_i) = \left( R(\pi) - \sum_{i=1}^m \lambda_i C_i(\pi) \right) + \sum_{i=1}^m \lambda_i b_i. \quad (2)$$

For fixed  $\lambda$ , maximizing  $\mathcal{L}(\pi, \lambda)$  over  $\pi \in \Pi$  is an *unconstrained* optimization problem: we are no longer enforcing category budgets directly, but instead penalizing them with category-specific prices  $\lambda_i$ . This motivates defining the dual function

$$d(\lambda) := \max_{\pi \in \Pi} \mathcal{L}(\pi, \lambda). \quad (3)$$

The dual problem is then

$$\min_{\lambda \geq 0} d(\lambda). \quad (4)$$

Under Slater's condition, strong duality holds in this finite setting:  $V(b) = \min_{\lambda \geq 0} d(\lambda)$ , and there exists at least one  $\lambda^*$  attaining the minimum. This is not merely a technical convenience. Strong duality is what allows us to interpret multipliers as meaningful marginal quantities and to view policy selection as (approximately) maximizing a penalized score with *the right* penalty weights.

**Dual decomposition and per-prompt structure.** The dual function separates across prompts because the objective is an expectation. Writing out the Lagrangian score per  $(x, y)$ ,

$$s_\lambda(x, y) := r(x, y) - \sum_{i=1}^m \lambda_i c_i(x, y),$$

we can rewrite

$$\mathcal{L}(\pi, \lambda) = \mathbb{E}_{x \sim D} \left[ \mathbb{E}_{y \sim \pi(\cdot|x)} [s_\lambda(x, y)] \right] + \lambda \cdot b.$$

Hence

$$d(\lambda) = \lambda \cdot b + \mathbb{E}_{x \sim D} \left[ \max_{\pi(\cdot|x) \in \Delta(\mathcal{Y})} \mathbb{E}_{y \sim \pi(\cdot|x)} [s_\lambda(x, y)] \right]. \quad (5)$$

Without additional regularization, the inner maximization for each  $x$  places all mass on any response  $y$  that maximizes  $s_\lambda(x, y)$ . Thus, for fixed  $\lambda$ , a best response  $\pi_\lambda$  can be chosen to be (almost everywhere) deterministic:

$$\pi_\lambda(y | x) \in \arg \max_{\tilde{\pi}(\cdot | x) \in \Delta(\mathcal{Y})} \mathbb{E}_{y \sim \tilde{\pi}(\cdot | x)} [s_\lambda(x, y)] \Rightarrow \text{supp}(\pi_\lambda(\cdot | x)) \subseteq \arg \max_{y \in \mathcal{Y}} s_\lambda(x, y). \quad (6)$$

This decomposition highlights why a *vector* of multipliers is qualitatively different from a single scalar penalty: the effective score  $s_\lambda(x, y)$  can shift sharply as  $\lambda$  changes component-wise, enabling targeted pressure on whichever categories are close to binding.

**KKT conditions and the meaning of  $\lambda_i$ .** A pair  $(\pi^*, \lambda^*)$  is primal–dual optimal if and only if it satisfies the Karush–Kuhn–Tucker conditions. In our setting they take a simple form:

$$C_i(\pi^*) \leq b_i \quad \forall i \quad (\text{primal feasibility}), \quad (7)$$

$$\lambda_i^* \geq 0 \quad \forall i \quad (\text{dual feasibility}), \quad (8)$$

$$\lambda_i^*(C_i(\pi^*) - b_i) = 0 \quad \forall i \quad (\text{complementary slackness}), \quad (9)$$

$$\pi^* \in \arg \max_{\pi \in \Pi} \mathcal{L}(\pi, \lambda^*) \quad (\text{stationarity / optimality}). \quad (10)$$

Complementary slackness encodes an operational governance fact: if a category budget is loose at the optimum (the measured cost is strictly below the threshold), then its multiplier is zero and that category exerts no marginal pressure on behavior. Conversely, if a category constraint binds, then  $\lambda_i^* > 0$  and the corresponding term  $-\lambda_i^* c_i(x, y)$  appears as an active penalty in the per-prompt score  $s_{\lambda^*}(x, y)$ . In this sense,  $\lambda_i^*$  is the developer-facing “risk price” that rationalizes the regulator’s vector of budgets.

The stationarity condition (10) also clarifies why fixed-weight scalarization is structurally brittle across different  $b$ . A fixed scalarization corresponds to forcing  $\lambda$  to lie on a one-dimensional ray (e.g.,  $\lambda = \kappa \nu$ ), whereas the KKT system generally selects  $\lambda^*(b)$  in an  $m$ -dimensional orthant, with support and magnitudes that change as the set of active constraints changes. When the binding set flips (say, privacy becomes tight after a product change), the appropriate  $\lambda^*$  can rotate sharply, producing exactly the kind of cross-category substitution that a single fixed  $\nu$  cannot anticipate.

**Shadow prices and envelope identification.** The most policy-relevant interpretation of  $\lambda^*$  comes from the value function  $V(b)$ . Under standard regularity (which holds generically in the finite model away from degeneracies), the envelope theorem yields

$$\frac{\partial V(b)}{\partial b_i} = \lambda_i^*(b) \quad \text{for each } i. \quad (11)$$

Thus  $\lambda_i^*$  is the *marginal helpfulness value* of relaxing category  $i$ ’s budget: if an auditor or regulator increases  $b_i$  by a small amount (holding other budgets fixed), the best achievable expected reward increases at rate  $\lambda_i^*$ . Equivalently, tightening a budget by  $\Delta b_i < 0$  imposes a first-order reward loss of approximately  $-\lambda_i^* \Delta b_i$ . This is a concrete quantity that can be reported and tracked: it is an endogenous measure of how expensive compliance is in a particular category at the current operating point.

Two immediate implications are worth flagging. First, large  $\lambda_i^*$  indicates that the system is operating on a steep portion of the frontier in category  $i$ : the developer must sacrifice a lot of helpfulness to buy a small reduction in expected harm. This can signal either genuine technical difficulty (limited model capacity or limited action richness) or a mis-specified measurement pipeline (the cost head is overly sensitive, or the audited distribution overweights hard cases). Second, because  $\lambda^*$  depends on  $D$ , shadow prices are *deployment-specific*: a shift in the prompt mix toward adversarial or high-stakes contexts effectively changes the frontier and can raise the multipliers even if the underlying model is unchanged. In governance terms,  $\lambda^*$  is a compact summary of how much “safety pressure” the environment and the budgets jointly induce.

**Uniqueness, degeneracy, and instability.** In the linear finite setting,  $\lambda^*$  need not be unique if there are multiple supporting hyperplanes at the optimum (e.g., if the optimal cost point lies on a flat face of the achievable set, or if multiple constraints are active with dependent normals). Our non-degeneracy assumptions are meant to rule out the most pathological cases and align with deployment intuition: typically, a small number of categories are near-binding and trade off sharply with reward, producing well-identified shadow prices. Still, even when  $\lambda^*$  is unique, the *primal* optimizer  $\pi^*$  can be non-unique because multiple actions may tie in the penalized score  $s_{\lambda^*}(x, \cdot)$  for some prompts.

This non-uniqueness is not merely aesthetic. If we implement  $\pi^*$  via a deterministic argmax of  $s_{\lambda^*}$ , then small estimation errors in  $r$  or  $c_i$ , or small changes in  $\lambda$ , can cause discontinuous flips in the chosen action for a prompt. Such brittleness is a recognizable failure mode in LLM systems: a tiny prompt rephrasing or logit perturbation switches the model from refusal to compliance, or from one compliance mode to another with a different harm profile. This motivates adding a stabilizing regularizer (most naturally, an entropy or KL term) so that the induced policy varies smoothly with  $\lambda$  and ties are resolved continuously rather than arbitrarily.

**Practical reading: multipliers as trainable safety knobs.** Although we have presented  $\lambda$  as an analytical device, it has an immediate operational analogue. In any pipeline that optimizes a penalized objective—

whether via RLHF-style policy gradients, rejection sampling, or decoding-time reweighting—we can interpret the coefficients multiplying different cost heads as multipliers. The dual view suggests that these coefficients should not be fixed global constants, but should be adapted until the measured constraints are met:

$$\lambda_i \text{ increases if audits estimate } C_i(\pi) > b_i, \quad \lambda_i \text{ decreases (or stays at zero) if } C_i(\pi) < b_i.$$

This is precisely the qualitative logic of dual ascent: raise the price of whatever budget is being violated. The key conceptual point is that this update is *category-wise* and does not require committing to a single scalar safety score.

At the same time, the dual perspective surfaces two limitations that matter for real audits. First, if cost estimates  $\hat{c}_i$  are noisy or systematically biased, then the learned  $\lambda$  will misprice risk, potentially inducing either over-conservatism or hidden leakage into unmeasured regions of  $\mathcal{X}$ . Second, because  $\lambda^*(b)$  is environment-dependent, a multiplier vector learned under one prompt distribution may not enforce budgets under another; distribution shift effectively changes the meaning of the constraint. Both issues argue for coupling multiplier learning to ongoing monitoring and for stress-testing under shifted (or adversarial) prompt mixes.

**Transition to closed-form structure.** So far, the primal–dual story identifies what the correct exchange rates  $\lambda^*$  *mean* and how they relate to budgets  $b$ . To turn this into a useful characterization of behavior, we next add a small entropic regularization. This does not change the economic interpretation of  $\lambda$ , but it yields a clean closed-form for  $\pi_\lambda$  (a per-prompt softmax over penalized scores), improves uniqueness and stability, and makes the comparative statics of leakage particularly transparent via smooth dependence of  $\pi^*$  on  $b$  through  $\lambda^*(b)$ .

**Entropy-regularized risk-budget allocation.** To obtain a closed-form and to remove the discontinuities induced by per-prompt argmax tie-breaking, we add a small entropic regularizer to the primal program. Concretely, for a temperature parameter  $\tau > 0$ , consider

$$\max_{\pi \in \Pi} \left\{ R(\pi) + \tau \mathbb{E}_{x \sim D} [H(\pi(\cdot | x))] \right\} \quad \text{s.t.} \quad C_i(\pi) \leq b_i \quad \forall i, \quad (12)$$

where  $H(\pi(\cdot | x)) := -\sum_{y \in \mathcal{Y}} \pi(y | x) \log \pi(y | x)$  is the Shannon entropy.<sup>1</sup> The economic content is unchanged: the regulator still supplies budgets  $b$ , and the developer still trades helpfulness against category costs. The role of

---

<sup>1</sup>In implementations, one often uses a KL regularizer to a reference policy  $\pi_0$  (e.g., an SFT model). Everything below extends by replacing  $H$  with  $-\text{KL}(\pi(\cdot | x) \parallel \pi_0(\cdot | x))$ ; the resulting closed-form is a softmax around  $\pi_0$  with the same linear penalty term  $-\lambda \cdot c$ .

$\tau$  is purely to pick a stable point on the frontier by making the per-prompt best response smooth and unique.

**Risk-Budget Allocation Theorem (closed-form policy and uniqueness).** The regularized Lagrangian is

$$\mathcal{L}_\tau(\pi, \lambda) := R(\pi) + \tau \mathbb{E}_x[H(\pi(\cdot | x))] - \sum_{i=1}^m \lambda_i (C_i(\pi) - b_i), \quad \lambda \geq 0. \quad (13)$$

As before, the problem decomposes pointwise in  $x$ . The entropy term makes the inner maximization *strictly* concave in  $\pi(\cdot | x)$ , which yields a Gibbs form.

**Theorem 4.1** (Risk-Budget Allocation / Gibbs Policy Structure). *Fix  $\tau > 0$  and budgets  $b \in \mathbb{R}_+^m$ . Assume Slater's condition. Then:*

1. *For any  $\lambda \geq 0$ , the maximizer  $\pi_\lambda \in \arg \max_{\pi \in \Pi} \mathcal{L}_\tau(\pi, \lambda)$  is unique and satisfies, for every  $x \in \mathcal{X}$ ,*

$$\pi_\lambda(y | x) = \frac{\exp\left(\frac{1}{\tau}(r(x, y) - \sum_{i=1}^m \lambda_i c_i(x, y))\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\frac{1}{\tau}(r(x, y') - \sum_{i=1}^m \lambda_i c_i(x, y'))\right)}. \quad (14)$$

*Equivalently,  $\log \pi_\lambda(\cdot | x)$  is an affine function of  $(r(x, \cdot), c_1(x, \cdot), \dots, c_m(x, \cdot))$  up to a normalizing constant.*

2. *The dual function  $d_\tau(\lambda) := \max_{\pi \in \Pi} \mathcal{L}_\tau(\pi, \lambda)$  is convex and continuously differentiable with gradient*

$$\nabla_\lambda d_\tau(\lambda) = b - C(\pi_\lambda), \quad (15)$$

*so dual optimality is equivalent to meeting budgets in the usual KKT sense.*

3. *If, at the dual optimum, the active categories have non-degenerate variation under  $\pi_{\lambda^*}$  (formally: the covariance matrix of the active-cost vector  $c(x, Y)$  under  $x \sim D, Y \sim \pi_{\lambda^*}(\cdot | x)$  is positive definite on the active coordinates), then the dual minimizer  $\lambda^*$  is unique, and hence the primal optimizer  $\pi^* = \pi_{\lambda^*}$  is unique.*

Theorem 4.1 is the precise sense in which a vector of risk budgets induces a vector of *endogenous* “category prices” that linearly penalize costs in the model’s effective score. In deployment terms, (14) says that  $\lambda$  acts like a category-wise logit adjustment: increasing  $\lambda_i$  subtracts  $\lambda_i c_i(x, y)/\tau$  from the log-probability of response  $y$  at prompt  $x$ , holding everything else fixed. This is exactly the structural form used implicitly by many multi-head “safety critics”—the theorem clarifies when such a form is not merely heuristic but actually optimal for the regularized constrained program.

**Stability and comparative statics through the dual geometry.** A central advantage of the entropy term is that it turns discontinuous argmax behavior into smooth dependence on  $\lambda$ , and therefore on the budgets  $b$  (through  $\lambda^*(b)$ ). Differentiating (15) and using the well-known softmax calculus yields a particularly interpretable curvature identity: the Hessian of the dual is (up to a  $1/\tau$  factor) a covariance of costs under the induced policy. Writing  $C(\pi_\lambda) = (C_1(\pi_\lambda), \dots, C_m(\pi_\lambda))$ ,

$$\nabla_\lambda^2 d_\tau(\lambda) = -\nabla_\lambda C(\pi_\lambda) = \frac{1}{\tau} \text{Cov}_{x \sim D, Y \sim \pi_\lambda(\cdot|x)}(c(x, Y)), \quad (16)$$

where  $c(x, Y) \in \mathbb{R}^m$  is the vector of category costs for the sampled response. Two operational points follow immediately. First,  $\nabla^2 d_\tau(\lambda) \succeq 0$  makes convexity tangible: dual optimization is well-behaved because curvature is literally “how much the policy randomizes across actions with different cost profiles.” Second, uniqueness of  $\lambda^*$  is tied to identifiable variation: if the system, under  $\pi_{\lambda^*}$ , never explores responses that trade off among the active categories, then those categories can become locally indistinguishable from the dual’s perspective, reintroducing degeneracy.

The same geometry yields a clean stability statement. When the active-cost covariance is well-conditioned (and  $\tau$  not too small), the mapping  $\lambda \mapsto C(\pi_\lambda)$  is Lipschitz, and hence small changes in budgets induce small changes in the learned multipliers and in behavior. By contrast, in the  $\tau \rightarrow 0$  limit,  $\pi_\lambda$  concentrates on per-prompt maximizers and  $\text{Cov}(c)$  collapses, precisely when we observe brittle “mode switching” between qualitatively different response types. In practice, this is one reason to view  $\tau$  (or an RLHF KL coefficient) as a governance-relevant parameter: it controls not only average performance but also the continuity of the safety–helpfulness tradeoff as audits tighten or relax budgets.

**Risk-budget allocation as a learnable control layer.** Theorem 4.1 also tells us what it means, algorithmically, to “allocate” safety effort across categories. Because  $\nabla_\lambda d_\tau(\lambda) = b - C(\pi_\lambda)$ , a canonical projected dual-ascent update is

$$\lambda^{t+1} = \left[ \lambda^t + \alpha_t (C(\pi_{\lambda^t}) - b) \right]_+, \quad (17)$$

where  $[\cdot]_+$  is projection onto  $\mathbb{R}_+^m$ . This update has a direct compliance interpretation: if audits estimate that category  $i$  exceeds budget, we raise its price; if it is comfortably below budget, the price decays toward zero. Importantly, the update is coordinate-wise: we do not need to convert harms into a single scalar score to decide how to respond to violations.

In real systems, we do not observe  $c_i(x, y)$  directly; we observe  $\hat{c}_i$  from classifiers, red-team labels, or post-hoc incident reports. The theorem therefore should be read as an *identification target*: the intended policy class is a softmax over a linear combination of a helpfulness score and multiple cost

heads, and  $\lambda$  is the set of coefficients that should be tuned until measured constraints are met on the relevant prompt distribution. When monitoring is noisy or the prompt mix shifts,  $\lambda$  must be treated as an adaptive state variable rather than a fixed hyperparameter.

**Why this is not scalarization in disguise.** It is tempting to read (14) as “just” a scalar reward  $r - \lambda \cdot c$ . The crucial distinction is that  $\lambda$  is *not* a fixed weight vector: it is an equilibrium object pinned down by budgets  $b$ , distribution  $D$ , and the available actions  $\mathcal{Y}$ . As budgets change (or as the audited distribution changes),  $\lambda^*(b)$  can rotate in the orthant, turning on new categories and turning off old ones via complementary slackness. This is exactly the mechanism by which vector constraints prevent category leakage: if tightening  $b_k$  makes category  $k$  bind, then  $\lambda_k^*$  rises and directly suppresses responses with high  $c_k$ , rather than indirectly hoping that a single global scalar penalty happened to put enough weight on that dimension.

This sets up the next section. Once we accept that the “right” penalty weights are budget- and environment-dependent, it becomes clear why fixed-weight scalar safety objectives are structurally misaligned with multi-category regulation: a single  $\nu$  cannot generally track the endogenous  $\lambda^*(b)$  as the binding set changes, and the mismatch shows up either as infeasibility (violated budgets) or as dominated tradeoffs (avoidable helpfulness loss for the same or higher harm).

**Why fixed-weight scalar safety objectives fail.** Many deployed training recipes implicitly assume that “safety” can be represented by a single scalar penalty—either by collapsing multiple incident types into one score, or by choosing fixed weights  $\nu \in \mathbb{R}_+^m$  and optimizing the scalarized objective

$$\max_{\pi \in \Pi} R(\pi) - \kappa \mathbb{E}_{x \sim D, y \sim \pi(\cdot|x)} [\nu \cdot c(x, y)], \quad \kappa > 0. \quad (18)$$

This looks superficially similar to the Gibbs form in (14), but the resemblance is precisely the trap:  $\nu$  is fixed by design-time preference, whereas  $\lambda^*$  is an equilibrium object that changes with budgets  $b$ , the prompt mix  $D$ , and the action set  $\mathcal{Y}$ . When the regulator is genuinely multi-dimensional (different categories with separate caps), a single fixed direction  $\nu$  cannot generally represent the full set of feasible supporting hyperplanes needed to implement constrained optima across different  $b$ .

A useful way to say this geometrically is: the constrained problem chooses a point on a Pareto frontier in  $(R, C_1, \dots, C_m)$  space by intersecting the feasible region with an axis-aligned box  $C \leq b$ . The associated KKT vector  $\lambda^*(b)$  is the normal to a supporting hyperplane at the optimum, and as the box changes shape (budgets tighten in one coordinate but not others), the supporting hyperplane generically rotates. Fixed-weight scalarization, by contrast, picks points supported by the single normal vector  $\nu$  (up to a global

scale  $\kappa$ ), so it can only track optima along budget changes for which  $\lambda^*(b)$  stays proportional to  $\nu$ . In non-degenerate multi-category environments, that proportionality is a knife-edge event.

**An explicit counterexample: deterministic “leakage” into underweighted categories.** We can exhibit the core failure mode with an intentionally simple finite instance where scalarization provably pushes harm into the wrong category. Let  $m = 2$ ,  $\mathcal{X} = \{x_1, x_2\}$ , and  $D(x_1) = D(x_2) = 1/2$ . Let  $\mathcal{Y} = \{A, B\}$ , and define rewards and costs by

$$r(x_1, A) = r(x_1, B) = r(x_2, A) = r(x_2, B) = 1,$$

and

$$\begin{aligned} c(x_1, A) &= (1, 0), & c(x_1, B) &= (0, 1), \\ c(x_2, A) &= (0, 1), & c(x_2, B) &= (1, 0). \end{aligned}$$

Intuitively: at each prompt there are two equally helpful responses, but they “swap” which category they harm. Now fix a scalar weight vector  $\nu = (2, 1)$  (category 1 is weighted twice category 2) and any  $\kappa > 0$ . Because the scalarized objective (18) is linear in  $\pi$  and decomposes pointwise in  $x$ , any scalar optimizer  $\pi^S$  can be chosen to minimize  $\nu \cdot c(x, y)$  at each prompt. Here,

$$\begin{aligned} \nu \cdot c(x_1, A) &= 2, & \nu \cdot c(x_1, B) &= 1 \quad \Rightarrow \quad \pi^S(B \mid x_1) = 1, \\ \nu \cdot c(x_2, A) &= 1, & \nu \cdot c(x_2, B) &= 2 \quad \Rightarrow \quad \pi^S(A \mid x_2) = 1. \end{aligned}$$

Therefore the induced expected costs are

$$C(\pi^S) = \frac{1}{2}(0, 1) + \frac{1}{2}(0, 1) = (0, 1).$$

Now choose any budgets  $b$  with  $b_2 < 1$ , for instance  $b = (0.9, 0.4)$ . The scalarized optimizer is infeasible for the true constrained problem, despite there existing feasible policies with the same reward. Indeed, for any  $p, q \in [0, 1]$  define a stochastic policy by

$$\pi(A \mid x_1) = p, \quad \pi(A \mid x_2) = q.$$

Then

$$C_1(\pi) = \frac{1}{2}p + \frac{1}{2}(1-q) = \frac{1}{2}(p+1-q), \quad C_2(\pi) = \frac{1}{2}(1-p) + \frac{1}{2}q = \frac{1}{2}(1-p+q).$$

Picking  $p = 0.6$  and  $q = 0.4$  yields  $C(\pi) = (0.6, 0.4)$ , which is feasible for  $b = (0.9, 0.4)$  and attains reward  $R(\pi) = 1$  (the same as the scalar optimum). What fails is not the existence of a good policy, but the *selection rule* induced by fixed scalar weights: because category 2 is relatively cheap in  $\nu \cdot c$ , the

scalar optimizer deterministically routes all harm into category 2. This is the simplest formal version of category leakage: scalarization implements a single exchange rate between harms, so the optimizer concentrates harm in whichever category is cheapest at that exchange rate, regardless of whether that category has a tight regulatory cap.

**Dominated outcomes: avoidable helpfulness loss at the same or higher harm.** The preceding example emphasized infeasibility. A second, more subtle pathology is inefficiency even when the scalar optimum happens to be feasible. The issue is that a fixed scalar penalty can force the developer to “pay” in reward along the wrong margin, selecting a policy that is Pareto-dominated in  $(R, -C)$  by another feasible policy.

One way to see this is to construct responses that trade off categories differently across prompts. Suppose again  $m = 2$  and  $D$  uniform on  $\{x_1, x_2\}$ , but now allow  $\mathcal{Y} = \{S, A, B\}$ , where  $S$  is a conservative “safe refusal/deflection” action with low reward and zero cost. Let

$$r(\cdot, S) = 0, \quad c(\cdot, S) = (0, 0),$$

and for a small  $\delta > 0$ ,

$$r(x_1, A) = 1, \quad c(x_1, A) = (1, 0), \quad r(x_1, B) = 1 - \delta, \quad c(x_1, B) = (0, 1),$$

$$r(x_2, A) = 1 - \delta, \quad c(x_2, A) = (0, 1), \quad r(x_2, B) = 1, \quad c(x_2, B) = (1, 0).$$

Here each prompt has one “high reward” action that harms category 1, and one slightly lower reward action that harms category 2 (and vice versa across prompts). A multi-budget regulator might set  $b = (0.5, 0.5)$ , under which a natural constrained optimum is to choose *both* high-reward actions half the time and both lower-reward actions half the time (or, more directly, to mix per prompt to hit  $C_1 = C_2 = 0.5$ ) without ever using  $S$ , yielding average reward close to  $1 - \delta/2$ . But for some fixed  $\nu$  and  $\kappa$ , the scalar objective can instead prefer routing both prompts to the same “cheaper” harm category (as above), and then compensate for violated budgets by using the conservative action  $S$  too often once additional heuristics are applied (e.g., post-training hard filters or refusal triggers). In such pipelines, the scalar training signal creates a model that is *structurally biased* toward one harm dimension; downstream enforcement then removes outputs to satisfy caps, producing a dominated final system: lower reward *and* no lower harm than could have been achieved by training directly with vector constraints. Put differently, scalarization encourages the wrong internal substitution pattern, and hard compliance layers pay an avoidable helpfulness tax.

This dominated-outcome phenomenon is easiest to interpret through KKT: the constrained optimum uses  $\lambda^*$  to shape the model’s indifference surfaces so that it trades off reward against the *binding* budgets. Fixed

scalarization uses  $\nu$  to hard-code an exchange rate between categories, which will generally not match  $\lambda^*(b)$ ; the result is either (i) a violation (necessitating blunt post hoc rejection) or (ii) an interior point that is not on the constrained frontier (leaving reward on the table for the same harm vector).

**When does scalarization work? (Knife-edge regimes).** There are regimes where fixed-weight scalarization can implement the constrained solution, but they are precisely those where multi-category control is not actually needed. Formally, for a given  $\nu$ , scalarization can match the constrained optimum for a family of budgets  $b$  only if the corresponding KKT multipliers satisfy

$$\lambda^*(b) \in \{\kappa \nu : \kappa \geq 0\} \quad \text{for all budgets in that family.} \quad (19)$$

Condition (19) is restrictive: it requires the *direction* of  $\lambda^*$  to remain constant as  $b$  varies, i.e., the set of binding constraints must not change in a way that rotates the normal vector.

Concretely, scalarization can work in (at least) three knife-edge cases:

1. *Effectively one-dimensional harm:* there exists  $w \in \mathbb{R}_+^m$  and scalars  $\rho(x, y)$  such that  $c(x, y) = w \rho(x, y)$  for all  $(x, y)$ . Then all cost vectors are collinear, and the constraints reduce to a single effective constraint  $\mathbb{E}[\rho] \leq \min_i b_i/w_i$ . Any  $\nu$  proportional to  $w$  is equivalent.
2. *Only one constraint binds (generically) over the budget range of interest:* if, for all relevant  $b$ , exactly one category  $k$  is active and the others are slack, then  $\lambda^*$  is supported on coordinate  $k$ , so any  $\nu$  that also puts all mass on  $k$  works. This is not a multi-category setting operationally; it says the other constraints are non-binding.
3. *Budgets vary only along a single ray with fixed shadow-price direction:* even when multiple constraints can bind, it may happen that the regulator only considers budgets  $b(t) = b_0 + t \Delta$  for which  $\lambda^*(b(t))$  stays proportional to some  $\nu$ . This can occur in highly symmetric or separable environments, but it is not stable to distribution shift in  $D$  or changes in  $\mathcal{Y}$ .

Outside these cases, the mismatch between fixed  $\nu$  and endogenous  $\lambda^*(b)$  is not a tuning problem; it is structural. One can of course fit  $\nu$  on a particular audit distribution and a particular set of budgets, but as soon as the binding set changes (tightening one category, or shifting the prompt mix toward a different incident type), the correct exchange rate between harms changes, and scalarization has no degrees of freedom to follow it.

**Safety implication: scalarization confuses “how much safety” with “which safety.”** From a governance standpoint, the key distinction is that  $\kappa$  only adjusts the *overall intensity* of the scalar penalty, whereas multi-category compliance requires adjusting the *composition* of safety effort across categories. Fixed scalarization can say “be safer in aggregate,” but it cannot say “shift safety effort from category 1 to category 2” when category 2 becomes binding. The predictable result is substitution: the system learns to avoid whatever is expensive under  $\nu \cdot c$  and to route failures into whatever is cheap, even if that cheap category has the strictest regulatory cap.

This is why the dual interpretation is more than mathematical convenience. Once we treat  $\lambda$  as a learnable control layer, we can ask comparative-static questions that are governance-relevant: how does tightening  $b_k$  change  $\lambda_k^*$  and spill over into other costs  $C_j$ ? how does a shift in the prompt distribution  $D$  (e.g., more adversarial queries) reallocate shadow prices? and how does expanding model capacity change the achievable frontier and reduce the shadow price of safety? We turn to these questions next, because they determine whether multi-category regulation is stable under the operational realities of distribution shift, adaptive attackers, and evolving model capabilities.

**Comparative statics as a governance interface.** Once we treat multi-category compliance as the constrained program (P) with dual variables  $\lambda^*$ , we obtain a language for how governance choices and operational realities propagate through the trained system. The central object is not merely the optimal reward  $R(\pi^*)$  or the realized cost vector  $C(\pi^*)$ , but the *shadow-price vector*  $\lambda^*(b, D, \mathcal{Y})$ , which summarizes which categories are effectively binding and how costly further tightening would be at the margin. In deployment terms,  $\lambda_i^*$  measures the marginal helpfulness sacrificed (in expectation) per unit of reduced expected harm in category  $i$ , *given* the current prompt mix and available behaviors. This is exactly the quantity a regulator implicitly manipulates when it moves budgets  $b$ , and exactly the quantity a developer must track as  $D$  and  $\mathcal{Y}$  evolve.

**Budgets  $b$  and the envelope theorem: why  $\lambda$  is the correct “price.”** Define the value function  $V(b) := \max_{\pi \in \Pi: C(\pi) \leq b} R(\pi)$ . Under the regularity assumptions already invoked for KKT (convexity/compactness and Slater), the envelope theorem yields, for each coordinate  $i$  where differentiability holds,

$$\frac{\partial V(b)}{\partial b_i} = \lambda_i^*(b). \quad (20)$$

This identity is more than a mathematical convenience. It gives a direct operational interpretation: if a regulator relaxes the allowable expected harm in category  $i$  by a small amount  $db_i$ , the best-achievable expected reward

increases by approximately  $\lambda_i^* db_i$ . Conversely, tightening  $b_i$  decreases reward at rate  $\lambda_i^*$ . Thus,  $\lambda^*$  provides an economically meaningful *unit conversion* between the regulator’s safety units and the developer’s performance units, and it does so separately for each harm category.

A second implication is diagnostic. If an audit reveals that tightening a particular budget produces a large degradation in helpfulness, that corresponds precisely to an increase in the relevant  $\lambda_i^*$ . In other words, the magnitude of  $\lambda_i^*$  is a quantitative measure of “compliance pressure” in category  $i$ : large  $\lambda_i^*$  indicates the developer is operating near the edge of feasibility for that category, and small  $\lambda_i^*$  indicates slack.

**Tightening one budget can increase other harms: formalizing leakage via cross-effects.** A regulator typically tightens one category at a time (e.g., a new privacy rule) while holding others fixed. The comparative statics of this operation are subtle because it changes the *shape* of the feasible set (the box  $C \leq b$ ), and the optimal policy responds by re-allocating probability mass across  $\mathcal{Y}$ . Even when the tightened category cost  $C_k(\pi^*)$  decreases, other category costs  $C_j(\pi^*)$  may increase: this is the formal counterpart of category leakage under multi-constraint control.

One clean way to see the mechanism is through the Gibbs form under entropy regularization (Proposition 2). For any  $\lambda$ , the induced policy  $\pi_\lambda$  behaves like a softmax over the penalized score  $r - \lambda \cdot c$ . As we raise  $\lambda_k$ , we shift probability mass away from actions with high  $c_k$ ; whether this mass flows toward low- $c_j$  actions depends on the joint structure of costs. In fact, in the regularized finite case, the mapping  $\lambda \mapsto C(\pi_\lambda)$  is smooth, and its Jacobian can be expressed in covariance form:

$$\frac{\partial C_i(\pi_\lambda)}{\partial \lambda_j} = -\frac{1}{\tau} \mathbb{E}_{x \sim D} \left[ \text{Cov}_{y \sim \pi_\lambda(\cdot|x)} (c_i(x, y), c_j(x, y)) \right]. \quad (21)$$

Equation (21) makes the governance concern precise. If, under the current policy, categories  $i$  and  $j$  are *substitutes* in the sense that actions tend to trade off  $c_i$  against  $c_j$  (negative covariance), then increasing  $\lambda_j$  (to enforce category  $j$ ) can *increase*  $C_i$ . This is exactly the leak: enforcement in one dimension pushes the model into regions of behavior where another dimension becomes worse unless that other dimension is also constrained.

From a regulatory-design perspective, this implies that budgets cannot be set independently as if categories were separable. If the covariance structure is strongly negative between two categories (e.g., privacy vs. refusal harms, or fraud vs. harassment in certain domains), then tightening only one budget predictably shifts violations into the other category, and the observed incident mix changes even if total “safety effort” increases.

**Prompt-mix shifts  $D$ : distributional governance and adversarial pressure.** The formal model treats  $D$  as exogenous, but in deployment  $D$

changes: user populations evolve, new product surfaces appear, and attackers actively search for high-cost regions. Comparative statics with respect to  $D$  explain why a system can become non-compliant without any change in weights. If the prompt distribution shifts mass toward contexts where certain costs are harder to avoid (higher irreducible  $c_i$  on the support), then  $C_i(\pi)$  increases for *every* fixed policy  $\pi$ , shrinking the feasible region and typically increasing the corresponding shadow prices  $\lambda_i^*$ .

In governance terms, this yields a concrete warning: compliance assessed on one audit distribution does not automatically transfer to another. It also suggests a natural role for “stress-test” distributions. If we evaluate not only under  $D$  but under a family  $\{D'\}$  that upweights rare but high-risk prompts (e.g., targeted red teaming), then we are effectively probing whether the system remains feasible under adversarially tilted environments. A robust extension makes this explicit by replacing  $D$  with an uncertainty set  $\mathcal{U}(D, \epsilon)$  and requiring constraints to hold under the worst case:

$$\sup_{D' \in \mathcal{U}(D, \epsilon)} \mathbb{E}_{x \sim D', y \sim \pi(\cdot|x)} [c_i(x, y)] \leq b_i.$$

Even without adopting full robustness, the comparative statics already tell us what to monitor: when prompt-mix indicators shift toward a category, we should expect  $\lambda_i^*$  to rise, and unless budgets are adjusted or capacity improves, either reward falls or other categories deteriorate via substitution.

**Capacity and action richness  $\mathcal{Y}$ : why better models reduce compliance pressure.** A frequent confusion in policy discussions is to treat safety and capability as opposing axes. In this framework, the interaction is conditional. Expanding the feasible response set  $\mathcal{Y}$  (or equivalently increasing model capacity so that more response distributions are attainable) *expands* the feasible frontier in  $(R, C)$  space. Holding budgets fixed, this expansion weakly increases the maximum achievable reward  $V(b)$  and weakly decreases the minimal required multipliers  $\lambda_i^*$  for active constraints: intuitively, with more expressive behaviors, we can satisfy the same caps with less distortion of helpfulness.

Governance implication: observed decreases in  $\lambda_i^*$  after a model upgrade are not necessarily evidence that the regulator has become laxer; they can reflect genuine technological improvements that make compliance cheaper. Conversely, if  $\lambda_i^*$  rises after a capability increase, that is evidence of either (i) a prompt-mix shift toward harder contexts, (ii) newly enabled behaviors that increase certain costs (a capability externality), or (iii) measurement drift in the cost model. This motivates treating  $\lambda$  time series as an operational telemetry signal: it is a compressed summary of the safety–capability tradeoff *as realized* under current conditions.

**Regularization and stability: avoiding brittle corner solutions.** Entropy (or KL) regularization parameter  $\tau$  does not merely smooth optimization; it affects governance-relevant behavior such as variance across prompts and susceptibility to leakage through rare, extreme actions. Smaller  $\tau$  concentrates  $\pi_\lambda$  on near-argmax actions of  $r - \lambda \cdot c$ , producing sharper compliance but potentially more brittle behavior: small shifts in  $D$  or in the estimated costs can flip which action is optimal, causing discontinuous changes in incident profiles. Larger  $\tau$  spreads mass and reduces such discontinuities, but can leave slack unexploited (lower  $R$  even when budgets are not tight) or fail to fully suppress tail events if the cost model is imperfect. This suggests a governance tradeoff between *responsiveness* (rapidly adapting to budgets) and *stability* (predictable behavior under drift). In practice, we can view  $\tau$  as a “policy inertia” knob that regulators may wish to constrain indirectly (e.g., via requirements on variance of outcomes across prompt slices).

**Regulatory interpretation: budgets as rights,  $\lambda$  as revealed priorities.** Budgets  $b$  are the regulator’s explicit policy instrument: they encode rights or limits per category (e.g., privacy leakage must be below a threshold). The dual vector  $\lambda^*$  is the system’s *revealed* prioritization needed to satisfy those rights under current conditions. Two governance uses follow.

First,  $\lambda^*$  supports *marginal impact reporting*. If an agency contemplates tightening  $b_k$ , a developer can report an estimated  $\lambda_k^*$  to quantify expected helpfulness loss per unit tightening, and can also report cross-category elasticities using estimates of (21) to anticipate leakage into other incident types. This is a more informative negotiation object than a single “safety score” because it decomposes the tradeoff by category.

Second,  $\lambda^*$  supports *audit targeting*. Large  $\lambda_i^*$  indicates the system is operating near the boundary in category  $i$ , meaning small measurement errors or distribution shifts are most likely to cause violation there. An auditor can use  $\lambda^*$  to allocate testing effort across categories and prompt slices: high- $\lambda$  categories warrant heavier sampling, adversarial search, and tighter confidence intervals.

**Limitations and open problems for governance-grade comparative statics.** These comparative statics are clean in the planner model but become fragile in practice for three reasons. (i) Costs  $c_i$  are estimated by imperfect classifiers; errors can induce spurious substitution patterns, effectively warping (21). (ii) The prompt process is strategic; attackers adapt to the current policy, making  $D$  endogenous and potentially invalidating static sensitivity analysis. (iii) Budgets often reflect tail-risk concerns (rare catastrophic events) rather than expectations; replacing  $C_i(\pi) = \mathbb{E}[c_i]$  with CVaR-type constraints changes the dual interpretation and may require richer multipliers indexed by quantile levels.

Nevertheless, the core governance lesson remains: multi-category regulation is inherently about *vector* tradeoffs. Comparative statics of  $\lambda^*$  provide a principled way to anticipate how tightening one category reallocates harm, how distribution shift can silently break compliance, and how increasing capability can either alleviate or exacerbate safety pressure depending on which parts of the frontier expand. This sets up the empirical question we address next: how to estimate these objects reliably in a real training-and-audit pipeline, and how to detect leakage empirically rather than assuming it away.

**Empirical design sketch: from the vector program to a training-and-audit loop.** The formalism above is intentionally static, but it naturally suggests an empirical design pattern: (i) learn a *multi-head* cost model that approximates the category cost functions  $c_i(x, y)$  under an evolving prompt distribution, (ii) train the deployed policy with a *vector Lagrangian* objective whose multipliers  $\lambda$  are updated from measured constraint violations, and (iii) run *targeted red teaming* that is explicitly designed to estimate cross-category substitution (leakage) rather than only marginal per-category incidence. We emphasize that we are not proposing a single canonical experiment; rather, we are sketching a governance-grade protocol that produces the objects regulators and auditors actually need: per-category compliance estimates, uncertainty bounds, and a leakage map that predicts what happens when one budget is tightened.

**Multi-head cost modeling: shared representation, category-specific calibration.** Operationally, the developer rarely has direct access to  $c_i(x, y)$ ; instead they have labeled samples  $(x, y, \ell_i)$  from auditors or internal review, with label noise and selection bias (because the reviewed outputs are not i.i.d. from  $D$ ). A pragmatic architecture is a shared backbone  $f_\phi(x, y)$  feeding  $m$  heads  $g_{\psi_i}$ , producing  $\hat{c}_i(x, y) = g_{\psi_i}(f_\phi(x, y))$ . Two design choices matter for governance.

First, *calibration by category* is not optional. If budgets  $b_i$  are interpreted as hard compliance targets, then systematic miscalibration in  $\hat{c}_i$  translates into predictable under- or over-enforcement. We therefore want post-hoc calibration curves per category and per prompt slice (e.g., topic clusters, languages, user segments), and we want to report calibrated estimates  $\tilde{c}_i$  (or conservative upper confidence bounds) rather than raw logits.

Second, we want *uncertainty* in the cost estimates, because the binding-constraint regime is precisely where small errors cause violations. In practice we can approximate uncertainty via ensembles  $\{\hat{c}_i^{(k)}\}$ , dropout-based estimates, or conformal prediction on held-out audited data. A simple governance-aligned rule is to enforce constraints on an upper bound,

$$\hat{C}_i^{\text{UCB}}(\pi) \leq b_i,$$

where  $\hat{C}_i^{\text{UCB}}$  is computed from the empirical mean plus a confidence radius (possibly slice-dependent). This turns the static feasibility condition into a form of *statistical compliance*, making explicit how much risk is being carried in measurement error.

**Vector-Lagrangian training: dual variables as a controllable interface.** Given reward modeling or preference data for helpfulness, we can train the policy with a regularized Lagrangian objective of the form

$$\max_{\pi \in \Pi} \mathbb{E}[r(x, y) - \lambda \cdot \tilde{c}(x, y)] - \tau \mathbb{E}[\log \pi(y | x)],$$

with  $\lambda \geq 0$ . This can be implemented with standard RLHF-style machinery by replacing the scalar reward with a *penalized reward*  $\tilde{r}_\lambda(x, y) = r(x, y) - \lambda \cdot \tilde{c}(x, y)$  and maintaining the KL/entropy term as a stability constraint. The key empirical point is not the exact optimizer, but the update loop for  $\lambda$ . A minimal dual update is projected ascent on constraint violations:

$$\lambda_{t+1,i} = \left[ \lambda_{t,i} + \alpha_t (\hat{C}_i(\pi_t) - b_i) \right]_+, \quad (22)$$

where  $\hat{C}_i(\pi_t)$  is an audit-estimated expected cost under the current policy (ideally corrected for selection bias), and  $[\cdot]_+$  is coordinate-wise projection onto  $\mathbb{R}_+$ . This is attractive for two reasons: it makes category priorities explicit and auditable (via the time series  $\lambda_t$ ), and it avoids the failure mode of fixed scalarization, where underweighted categories silently absorb harm.

In a deployment setting, we typically cannot estimate  $\hat{C}_i(\pi_t)$  from purely online traffic without introducing unacceptable risk. A common compromise is a staged loop: update  $\lambda$  on a controlled evaluation mixture that includes (a) a naturalistic sample approximating current  $D$ , and (b) stress-test slices described below. This yields  $\lambda$  that is tuned to both ordinary usage and foreseeable adversarial pressure, rather than to whichever distribution happens to dominate logs this week.

**Separating three distributions: training, audit, and stress tests.** To measure leakage credibly, we need to stop pretending there is a single  $D$ . We propose maintaining three distributions (or families) throughout the pipeline.

1. A *training distribution*  $D_{\text{train}}$  reflecting product usage and standard data collection.
2. An *audit distribution*  $D_{\text{audit}}$  designed for unbiased estimation of expected category costs (with documented sampling and annotation procedures).

3. A set of *stress-test distributions*  $\{D_{\text{stress}}^{(k)}\}$  that intentionally upweight risky regions (prompt templates, adversarial prompt generators, or curated red-team corpora).

The compliance claim should be explicitly indexed:  $\hat{C}_i(\pi; D_{\text{audit}}) \leq b_i$  with confidence  $1 - \delta$ , plus separate reporting of  $\hat{C}_i(\pi; D_{\text{stress}}^{(k)})$  as a robustness diagnostic. This avoids the common governance failure where a model is “compliant” on a benign audit mix while being brittle under predictable misuse.

**Targeted red teaming as leakage measurement, not only worst-case discovery.** Red teaming is often treated as a search for isolated bad prompts. For leakage, we need something more structured: we want to learn how tightening one category changes the incident mix. Concretely, we can run red-team prompt generation in two modes.

*Single-category maximization:* for each category  $i$ , generate prompts  $x$  that maximize  $\mathbb{E}_{y \sim \pi(\cdot|x)}[\tilde{c}_i(x, y)]$  subject to basic plausibility constraints. This estimates where the model is near the boundary in category  $i$ , and provides high-signal evaluation data for that head.

*Cross-category tradeoff search:* generate prompts that maximize a contrast such as  $\tilde{c}_j(x, y) - \tilde{c}_i(x, y)$  (or that maximize  $\tilde{c}_j$  while constraining  $\tilde{c}_i$  to be low). This specifically looks for regions where behaviors that reduce category  $i$  tend to increase category  $j$ , which is the empirical signature of substitution.

The second mode is the core governance contribution: it produces a *map of likely leakage channels* that can be used both to adjust budgets jointly and to target additional mitigations (e.g., adding refusal-safe completions that are low cost in multiple categories rather than merely shifting mass between them).

**Metrics: compliance, leakage, and stability.** Beyond reporting  $\hat{C}_i$  and  $R$ , we need metrics that correspond to the comparative statics objects in the theory.

*Constraint violation probability and margin:* report not only  $\hat{C}_i - b_i$  but also the estimated probability that  $C_i(\pi) > b_i$  under annotation uncertainty, and a standardized margin  $(\hat{C}_i - b_i)/\text{SE}(\hat{C}_i)$ . This is what makes “near-binding” operational.

*Shadow-price telemetry:* track  $\lambda_i$  over training and across model versions. Large  $\lambda_i$  indicates that category  $i$  is expensive to satisfy under current conditions; a sudden jump in  $\lambda_i$  is an early warning for distribution shift, measurement drift, or a newly enabled harmful capability.

*Leakage matrix estimation:* estimate the Jacobian of costs with respect to multipliers (or budgets). In the regularized setting, Equation (21) suggests

an estimator:

$$\hat{J}_{ij} := \frac{\partial \hat{C}_i(\pi_\lambda)}{\partial \lambda_j} \approx -\frac{1}{\tau} \frac{1}{n} \sum_{t=1}^n \text{Cov}_{y \sim \pi_\lambda(\cdot|x_t)}(\tilde{c}_i(x_t, y), \tilde{c}_j(x_t, y)),$$

with  $x_t \sim D_{\text{audit}}$  (and separately for each stress-test slice). Negative  $\hat{J}_{ij}$  (equivalently negative covariance) indicates substitutability and thus potential leakage: increasing  $\lambda_j$  may increase  $C_i$ . For interpretability, we can also report finite-difference elasticities by re-solving for  $\lambda$  after small perturbations to  $b$ , yielding an empirical approximation of  $\partial C_i(\pi^*(b))/\partial b_j$ .

*Frontier reporting:* when feasible, compute a local approximation to the Pareto frontier by sweeping a subset of budgets  $b$  (or sweeping  $\lambda$ ) and reporting  $(\hat{R}, \hat{C})$  pairs. This directly addresses the governance question “what performance is achievable at what safety profile?” without collapsing everything into a single score.

**Evaluation protocol: avoiding selection bias and gaming.** Two implementation details are easy to get wrong.

First, if  $\hat{C}_i$  is computed on outputs that have already been filtered by safety systems, the estimate is biased downward. We therefore want an *evaluation-only sampling mode* that logs unfiltered candidate outputs under controlled access (or uses offline sampling from the policy) and then applies auditing to those outputs. When online logging is unavoidable, we can use importance sampling to correct for known filters, but the variance can be large; in high-stakes categories, conservative bounds may be more appropriate than point estimates.

Second, once developers know the exact audit prompts, Goodhart effects appear. Stress-test sets should therefore be periodically refreshed, partially held out, and complemented by adversarial prompt generation that is not fully disclosed. The goal is not secrecy as a substitute for rigor, but rather ensuring that the reported  $\hat{C}_i(\cdot; D_{\text{audit}})$  generalizes to a documented class of shifts  $\{D_{\text{stress}}^{(k)}\}$ .

**What this design buys us, and what it cannot.** If implemented, this pipeline yields three governance-relevant artifacts: (i) a calibrated estimate of per-category compliance with uncertainty, (ii) an explicit vector  $\lambda$  that can be reported and monitored as “compliance pressure,” and (iii) an empirical leakage matrix that predicts which incident types will rise when another is suppressed. The limitations are equally important: if the cost heads miss a harm mode, dual updates will faithfully enforce the wrong constraints; if auditors disagree on definitions,  $\lambda$  becomes a price on an unstable commodity; and if adversaries can move  $D$  faster than audits update, static estimates will lag.

These caveats motivate the next section: how audit reporting should be structured, how liability could be differentiated by category and by demonstrated leakage controls, and what deployment commitments (monitoring, refresh cadence, red-team scope) are plausible as enforceable requirements rather than aspirational best practices.

## 5 Discussion: audit reporting, differentiated liability, and deployment recommendations; limitations and extensions

The vector-constraint formulation reframes “alignment” as a compliance problem with an explicit interface between (i) a regulator who sets per-category budgets  $b$  and (ii) a developer who produces a policy  $\pi$  plus auditable evidence that  $\mathbb{E}[c_i] \leq b_i$ . The practical question is therefore not merely how to reduce a scalar “toxicity score,” but how to make a *credible claim* of multi-category compliance under measurement noise, distribution shift, and strategic adaptation. In this section we discuss what an audit report should contain, how liability could be differentiated in a way that discourages category leakage rather than rewarding it, and what deployment commitments are plausibly enforceable. We then flag limitations of the static model and sketch extensions that matter in practice (multi-turn interaction and robust/adversarial prompt distributions).

**Audit reporting as a contract over distributions and estimators.** A central governance failure mode is ambiguity about what distribution the compliance claim ranges over. An audit report should therefore state (a) the prompt distribution(s) on which each estimate is computed, (b) the estimator used, and (c) uncertainty quantification. Concretely, for each category  $i$  the report should include an estimate  $\hat{C}_i(\pi; D_{\text{audit}})$  with a confidence bound, e.g.,

$$\Pr\left(C_i(\pi; D_{\text{audit}}) \leq \hat{C}_i(\pi; D_{\text{audit}}) + \rho_i\right) \geq 1 - \delta_i,$$

and it should declare whether the compliance criterion is the point estimate  $\hat{C}_i \leq b_i$  or the conservative criterion  $\hat{C}_i + \rho_i \leq b_i$ . This distinction is not pedantic: in near-binding regimes, the difference between point estimates and upper bounds is often the difference between de facto violation and reliable compliance.

We also want the report to expose the *structure* of safety tradeoffs rather than compressing them into a single number. Two objects are especially informative. First is a time-stamped record of the multipliers used to train or tune the model (or, in a post-training setting, to configure a safety layer),  $\lambda$ , along with the measured violations that induced changes. While  $\lambda$  is not itself a guarantee, it is operational telemetry: persistent large  $\lambda_i$  is a signal

that category  $i$  is expensive to satisfy and thus likely to be brittle under shift; sudden changes in  $\lambda_i$  are an early warning for drift in either the model or the measurement pipeline. Second is a leakage summary that captures cross-category substitution. At minimum, auditors should require either a local Jacobian estimate  $\widehat{J}$  (however computed) or a set of finite-difference stress results showing how costs move when one budget is tightened. Without such a leakage artifact, developers can meet a narrowly measured target while silently moving harm into underweighted categories—a Goodhart channel that is predictable from the theory.

**Slicing, tails, and conditional compliance.** Expected costs  $\mathbb{E}[c_i]$  are often insufficient in high-stakes categories, because low-probability contexts can dominate real-world harm. A governance-aligned audit report should therefore include slice-based estimates and at least one tail-sensitive statistic. Slice reporting can be formalized as a family of conditional constraints  $\mathbb{E}[c_i | x \in S] \leq b_{i,S}$  for documented slices  $S$  (e.g., user age group, language, topic cluster, tool-use mode). Tail sensitivity can be implemented via quantile constraints, conditional value-at-risk, or exceedance probabilities. For example, for a threshold  $t_i$  meaningful to auditors, one can report

$$p_i(\pi) := \Pr(c_i(x, y) \geq t_i),$$

and treat  $p_i(\pi) \leq \beta_i$  as a supplementary budget. This is not a purely technical refinement: it aligns reporting with how regulators and courts reason about rare but severe incidents.

**Differentiated liability: pricing leakage rather than rewarding it.** If budgets  $b_i$  are to function as enforceable constraints, liability should track (i) realized violations, (ii) the reasonableness of measurement and monitoring, and (iii) the foreseeability of leakage. A naive liability regime that penalizes only whichever incident type is most salient invites substitution: the developer reduces that visible category and lets other categories rise. Our framework suggests a more robust approach: liability should be *vector-valued* (or at least indexed by category), and safe-harbor provisions should depend on the presence of leakage controls.

One plausible regime is a two-part standard. First, strict or negligence-like liability attaches to *ex post* category exceedances above declared budgets on a stated audit distribution (plus documented stress tests), subject to clearly specified measurement procedures. Second, a safe harbor (reduced damages, reduced penalties, or a rebuttable presumption of due care) is available only if the developer can show: (a) calibrated per-category measurement, (b) monitoring sufficient to detect drift, and (c) an explicit leakage evaluation demonstrating that tightening one category does not predictably increase another beyond declared tolerances. In other words, the developer

is not rewarded for merely hitting a target; they are rewarded for demonstrating that the system behaves like a controlled multi-constraint optimizer rather than a brittle scalar scorer.

Shadow prices  $\lambda^*$  provide an additional lever for differentiated treatment. While  $\lambda^*$  is not directly observable as a legal fact, it is indirectly identifiable from comparative statics and training telemetry, and it encodes which constraints are binding. If a developer reports that a category has a tight budget  $b_i$  yet trains with effectively zero pressure on that category (low  $\lambda_i$  and no evidence of near-binding behavior), this is a red flag: either the measurement pipeline is miscalibrated, or the compliance claim is not being operationalized. Conversely, a developer that demonstrates sustained enforcement pressure and still observes high estimated costs can credibly argue that the frontier is technologically constrained, motivating either revised budgets or targeted investment in mitigations. This is the institutional interpretation of the dual variables: they separate “we chose not to pay for safety” from “the current technology makes this safety level costly.”

**Practical deployment recommendations: commitments that can be audited.** Beyond the training details, what can regulators plausibly require in deployment? We think the right level of abstraction is to require commitments over *interfaces and processes* rather than over any single model architecture.

First, require a budget interface: the deployed system must expose controllable parameters that implement per-category tradeoffs (directly as  $\lambda$ , indirectly via policy variants, or via a certified safety wrapper). The point is to avoid a regime where the developer can only offer “the model” and cannot respond to tightened budgets without retraining from scratch.

Second, require monitoring and refresh cadence: the developer must specify how often  $\hat{C}_i$  is re-estimated, how drift is detected, and what triggers a rollback or reconfiguration. Because prompt distributions change (product features, user populations, adversaries), a one-time audit is closer to a snapshot than a guarantee.

Third, require stress-test disclosure and reproducibility: auditors should be able to reproduce the compliance estimate on  $D_{\text{audit}}$  and independently evaluate on a declared family of stress tests. The goal is not to disclose every red-team prompt (which can itself induce gaming), but to disclose the *method class* of stress testing and to commit to periodic refresh and partial holdout.

Fourth, require incident handling aligned with categories: when a severe incident occurs, the response should include which constraint(s) failed, how measurement missed it (if it did), and whether the incident is consistent with predicted leakage channels. This closes the loop between theory (substitution) and operational accountability (postmortems that do not collapse

everything into “model was unsafe”).

**Limitations of the static one-shot model.** Our baseline program treats  $\mathcal{X}$  as drawn from an exogenous  $D$ , assumes bounded costs, and enforces expectations. Each of these assumptions breaks in recognizable deployment scenarios. The most severe limitation is misspecification of  $c_i$ : if auditors cannot reliably label a harm mode, or if the harm is inherently contextual and only emerges over time, then the constraint is enforcing the wrong object. A second limitation is that “category” boundaries are neither natural nor stable; correlated harms can make per-category budgets incomplete, and new capabilities can create new categories. Third, the convexity of the finite mixture model hides nonconvexities in real training dynamics; the KKT picture is an equilibrium idealization, not a guarantee about gradient-based training in large models. These limitations do not invalidate the framework, but they shift the governance emphasis from “solve the optimization” to “maintain a measurement-and-control system that approximates it and fails loudly when assumptions break.”

**Extension: multi-turn interaction as a constrained control problem.** Many high-stakes harms are multi-turn: persuasion, grooming, incremental disclosure of private data, or tool-mediated fraud unfold over trajectories rather than single responses. A natural extension is to replace prompts  $x$  with histories  $h_t$  and treat the model as a policy in a partially observed decision process. Let  $y_t \sim \pi(\cdot | h_t)$  and define per-step rewards  $r(h_t, y_t)$  and costs  $c_i(h_t, y_t)$ . For a horizon  $T$  (or discount  $\gamma$ ), the constrained objective becomes

$$\max_{\pi} \mathbb{E} \left[ \sum_{t=1}^T r(h_t, y_t) \right] \quad \text{s.t.} \quad \mathbb{E} \left[ \sum_{t=1}^T c_i(h_t, y_t) \right] \leq b_i \quad \forall i.$$

This is a standard constrained RL problem, but with an alignment-specific twist: auditors often only observe sparse or delayed labels (e.g., the conversation becomes unsafe after several turns). In that regime, the multi-head cost model must be defined over trajectories or augmented with credit assignment. Governance-wise, the implication is that single-turn audits can be systematically optimistic: a system may look compliant on isolated prompts yet reliably drift into unsafe regions in longer interactions. Hence, a credible audit program should include multi-turn evaluations and budgets over trajectory-level costs, even if the deployed system is primarily single-turn in ordinary usage (because users can chain prompts).

**Extension: robust and adversarial prompt distributions.** The other key limitation is treating  $D$  as fixed. In practice, both benign shifts (new product features) and strategic shifts (attackers) change the prompt mix. A

minimal robustness upgrade is distributionally robust optimization: define an uncertainty set  $\mathcal{U}_\epsilon(D)$  (e.g., an  $f$ -divergence ball or a Wasserstein ball) and require constraints to hold in the worst case,

$$\max_{\pi} R_D(\pi) \quad \text{s.t.} \quad \sup_{Q \in \mathcal{U}_\epsilon(D)} C_i(\pi; Q) \leq b_i \quad \forall i.$$

This formalizes the governance intuition that compliance should not be a knife-edge property of the current audit mix. It also clarifies what “adversarial robustness” means in a compliance setting: not necessarily worst-case over *all* strings, but worst-case over a documented class of distribution shifts that are plausible given the product surface.

A more game-theoretic extension endogenizes  $D$  by introducing an attacker who chooses  $Q$  (within feasible constraints) in response to  $\pi$ . One can model this as a Stackelberg game (developer commits to  $\pi$ , attacker chooses  $Q$ ) or as a repeated interaction where attackers learn over time. The resulting equilibrium typically increases the effective shadow prices on the attacked categories and makes leakage more salient: suppressing one harm invites attackers to search for the next cheapest category. Institutionally, this argues for joint budgeting (avoid leaving “slack” categories that become attack targets) and for monitoring that detects changes in the empirical  $\lambda$  or in slice-specific costs as a signal of adaptive pressure.

**A final governance implication: compliance is an ongoing measurement problem.** Taken together, these discussion points reinforce a common theme: the core object that regulators need is not a single safety score, but a system of measurement and control that (i) supports vector-valued constraints, (ii) reports uncertainty and tail risk, (iii) detects leakage, and (iv) is robust to predictable shifts in how the system is used and attacked. The formalism motivates these requirements by making the failure modes legible: whenever safety is scalarized, or whenever compliance is claimed on an ill-defined  $D$ , we should expect substitution, brittle generalization, and incentives to optimize the metric rather than the harm.

## 6 Conclusion: alignment as an economic institution, and open questions

We can now summarize the main lesson of the paper in one sentence: once we move from a single, vaguely defined notion of “safety” to a regulated environment with multiple harm categories, alignment is naturally described as an *institutional* problem—a system of budgets, measurements, incentives, and adaptive control—rather than as the minimization of any one scalar metric. The formalism is deliberately spare (finite  $\mathcal{X}$ , finite  $\mathcal{Y}$ , expectations under  $D$ ), but it forces clarity about what a compliance claim means and

why many intuitive training heuristics break once categories are numerous, heterogeneous, and politically salient.

A first takeaway is conceptual. The vector-constrained program turns alignment from an aesthetic preference into a *contractible interface*: the regulator selects a vector  $b$  of permissible risk, and the developer supplies a policy  $\pi$  plus evidence that the induced expected costs satisfy  $C_i(\pi) \leq b_i$ . In this picture,  $\lambda^*$  is not merely a mathematical artifact; it is the canonical representation of the marginal difficulty of compliance across categories. Shadow prices provide a language for discussing tradeoffs that is simultaneously technical and legible to governance: when  $\lambda_i^*$  is persistently large, category  $i$  is not “ignored,” it is expensive, brittle, and likely to fail under shift; when  $\lambda_i^* \approx 0$  despite a supposedly tight budget, we should suspect either slackness, mismeasurement, or performative compliance. This is the sense in which the Lagrangian is an institutional object: it mediates between what is demanded (budgets), what is feasible (the frontier induced by  $\mathcal{Y}$  and model capacity), and what is actually enforced (training and deployment control knobs).

A second takeaway is negative but practically important. Fixed-weight scalarization is not just “suboptimal”; in multi-category settings it is structurally incapable of implementing a regulator’s menu of possible budgets. The reason is the familiar geometry of multi-objective optimization: a single weight vector  $\nu$  can support at most a subset of Pareto-optimal points, and which point is implementable depends on the supporting hyperplane that itself changes with  $b$ . The governance translation is leakage: if the developer optimizes a single score, we should expect harm to migrate into whichever categories are least represented by that score or least salient in enforcement. This does not require malice; it follows from optimization. As a result, a regulatory regime that treats compliance as “hit the metric” is, in our view, an incentives regime that selects for Goodharting.

A third takeaway concerns verification. The formal model makes clear that the object of interest is a statement about a *distribution* (and often about tails or slices), not about an unconditional average in a vacuum. In deployment, distributions move, measurement pipelines drift, and attackers search over  $\mathcal{X}$  to find slack. Thus, the practical endgame is not a one-time demonstration of low  $\hat{C}_i$ , but an ongoing control loop that keeps estimated costs within budgets as the effective  $D$  changes. This is where the economics and the computer science meet: the regulator is effectively designing an environment in which developers internalize shadow prices and invest in measurement, while developers are solving a constrained learning/control problem with partial observability and noisy labels. The model is simple enough to expose the moving parts, yet expressive enough to explain why organizations repeatedly fail when they collapse compliance into a single number.

A fourth takeaway is about the division of labor between technical and

institutional interventions. The mathematics suggests a clean decomposition. Technical work expands the feasible frontier (better model capacity, better refusal policies, better tool-use containment, better cost modeling), which reduces the required  $\lambda^*$  to satisfy a given  $b$ . Institutional work clarifies  $b$ , specifies the evidentiary standard for claims about  $C_i(\pi)$ , and aligns incentives so that developers cannot profitably substitute across categories. Importantly, neither side alone solves the problem: stronger institutions without better technical frontiers can force socially costly restrictions (large decreases in  $R(\pi^*)$ ), while better models without vector-valued oversight can simply reallocate harm.

These conclusions motivate a set of open questions that we view as both technically substantive and governance-relevant.

**(1) Measurement validity and identifiability of costs.** Our framework assumes that the category costs  $c_i(x, y)$  exist as well-defined objects. In reality they are constructed from labels, policies, and legal categories, and are therefore subject to misspecification and strategic pressure. What does it mean to have a “calibrated” cost model for harms that are context-dependent, rare, or only legible after downstream consequences? When can we reliably estimate  $C_i(\pi)$  under distribution shift, especially when the system itself changes user behavior? A particularly sharp question is identifiability: under what conditions can auditors infer anything like  $\lambda^*$  or the location of the frontier from partial telemetry, without access to proprietary training details?

**(2) Tail risk objectives beyond expectations.** We gestured at tail-sensitive reporting, but a deeper issue remains: which tail notions are institutionally stable and technically tractable? Quantiles and CVaR are attractive, but they can be brittle under small sample sizes and slice granularity. Exceedance probabilities are interpretable but depend on threshold choices that can be gamed. More broadly, we lack a mature theory of *vector-valued* tail constraints (multiple  $i$ , multiple slices  $S$ ) that yields both computationally feasible training procedures and audit procedures with meaningful statistical power.

**(3) Dynamics: learning, adaptation, and non-stationarity.** The static program is an equilibrium idealization. In practice,  $\pi$  is produced by a nonconvex training process, and the effective constraints evolve as the developer updates models, filters, policies, and tooling. Moreover, the environment adapts: users learn what works, and attackers probe for failure modes. A robust institutional theory should therefore treat compliance as a repeated game with feedback, where budgets and measurement protocols may themselves update. This raises questions familiar from mechanism de-

sign: how should budgets  $b$  be revised when technology improves, when categories become obsolete, or when new categories emerge? How should we prevent “budget shopping” across jurisdictions or product variants? And how should regulators reason about transient violations during model updates?

**(4) Compositionality and systems-level costs.** Many high-impact deployments are not a single policy  $\pi$  but a system: multiple models, routing, tools, retrieval, and post-processing filters. Category costs can interact non-linearly across components (e.g., a safe base model paired with an unsafe tool can increase fraud risk). We need compositional guarantees: if each subsystem satisfies certain budgets under certain conditions, what can be said about the composed system? Conversely, if the system violates a budget, can we localize which component is responsible in a way that supports accountability and remediation?

**(5) Incentive-compatible disclosure and anti-gaming design.** Auditing requires disclosure; disclosure invites gaming. The question is not whether to disclose, but what to disclose so that the resulting equilibrium improves welfare. What is the right “minimal sufficient statistic” for compliance that allows verification while limiting exploitation (for example, disclosure of stress-test methodology classes rather than specific prompt sets)? How should we treat confidentiality when the very act of revealing slices and thresholds can create new attack surfaces? This is a core institutional design problem, not merely a technicality.

**(6) Computational constraints and approximate optimality.** Even in finite settings, scaling vector-constraint methods to frontier-scale models involves approximations: learned proxies for  $c_i$ , stochastic estimates of  $C_i$ , and imperfect optimization. A key open problem is to characterize what kinds of approximation error are tolerable under a regulatory lens. When does approximate primal feasibility imply anything meaningful about real-world incident rates? How can we design training and monitoring procedures that come with auditable error bars on both reward and constraint satisfaction, rather than only on in-distribution performance?

**(7) Normative uncertainty and category definition.** Finally, category budgets presume categories. But categories are socially contested, and different jurisdictions will draw boundaries differently (e.g., what counts as “political persuasion” or “self-harm assistance”). A realistic alignment institution must therefore handle normative uncertainty: how do we design budgets and audits that can be updated as categories evolve, without creating perverse incentives to exploit definitional gaps? Technically, this suggests designing flexible cost representations and stress-test suites; institutionally,

it suggests procedures for revising category taxonomies that do not reset accountability.

Stepping back, the formalism offers a pragmatic stance. We should expect frontier constraints: there will be domains where satisfying certain  $b$  vectors is genuinely costly, and society must decide whether to pay that cost (in reduced functionality, increased friction, delayed deployment) or to relax budgets. What the model provides is a way to make that decision explicit and contestable. It separates questions of feasibility (what is attainable given  $\mathcal{Y}$  and model capability) from questions of preference and policy (what  $b$  should be), and it highlights the predictable failure mode of pretending that a scalar score can substitute for that separation.

If we take alignment seriously as an economic institution, our goal is not to claim that a model is “safe” in the abstract. Our goal is to build systems in which (i) the relevant harms are measured with known error, (ii) tradeoffs are represented transparently as vectors rather than hidden in scalar objectives, (iii) incentives discourage leakage and reward robust control, and (iv) compliance remains meaningful under distribution shift and strategic pressure. The mathematical apparatus is, in that sense, only a map—but it is a map that makes the hard parts visible, and therefore makes them governable.