Contextual-MAIL: Decentralized Incentivized Learning in Hierarchical Principal–Agent Contextual Bandits

Liz Lemma Future Detective

January 16, 2026

Abstract

Hierarchical delegation is now a dominant pattern in agentic AI systems: an upstream orchestrator routes tasks to sub-agents, which further subcontract to tools and data services. The resulting externalities are local but propagate up the hierarchy, and naïve independent learning generally fails to maximize total welfare. Building on the tree-structured principal–agent bandit framework with action-contingent transfers, we extend the model to contextual tasks and propose Contextual-MAIL, a fully decentralized algorithm that learns context-dependent incentives and actions. The key technical challenge is to estimate, from local interaction alone, the minimal transfer required to induce a child's context-dependent best response while maintaining learnability of the parent's reward function. We address this by combining (i) context-indexed transfer estimation under repeated contexts and (ii) contextual bandit learning on shifted rewards that subtract estimated incentive costs. Under action observability, bounded rewards, and mild separability that avoids exponential joint-action enumeration, we prove sublinear individual regret for every node and sublinear social-welfare regret for the whole tree. The results push transfer-based efficiency restoration toward the 2026 reality of heterogeneous tasks (LLM toolchains, cloud API orchestration, delegated data collection) while clarifying the tradeoffs between context richness, payment burn, and error propagation across layers.

Table of Contents

3. 3. Model: tree structure, contexts, observability, contracts; separability assumptions enabling polynomial-time decision-making; definitions of welfare and regret.

4. 4. Optimal contextual incentives in hindsight: recursive definition of continuation utilities $\mu_v^\star(c, a)$; characterization of minimal context-dependent transfers $\tau_w^\star(c, b)$; welfare equivalence identity (telescoping).

5. 5. Algorithm (Contextual-MAIL): warm-up, context-indexed incentive estimation, and contextual bandit learning on shifted rewards; implementation details and computational complexity.

6. 6. Theory: regret decomposition (action/payment/deviation) in contextual setting; concentration and uniform-in-time guarantees; propagation lemma across layers; main theorems (node regret and welfare regret).

7. 7. Extensions and variants: continuous contexts (Lipschitz/linear), approximate separability, limited budgets, and partial observability; where numerical methods are needed.

8. 8. Experiments: synthetic contextual hierarchies; LLM toolchain proxy environment; ablations (payments vs compliance vs welfare); sensitivity to $|\mathcal{C}|$, depth, and context frequency.

9. 9. Discussion: design guidance for platforms (credits, bonuses, routing incentives); limitations; open problems (DAGs, MDPs, moral hazard).

1 Introduction and motivation

Hierarchical delegation is no longer a stylized organizational chart; in 2026 it is a software architecture. Modern agent ecosystems routinely implement *stacks* of decision makers: a top-level orchestrator decomposes a user objective into tasks, routes tasks to specialized sub-agents (planning, retrieval, execution), and those sub-agents further delegate to tools or downstream services. This layered design is attractive because it exploits specialization and modularity, but it also reintroduces a classical economic friction at machine speed: each agent optimizes a local objective under partial feedback, while the system designer cares about global performance. When downstream agents are adaptive learners, the upstream principal cannot simply treat them as static best responses; incentives and learning interact, and misalignment can accumulate as it propagates up the hierarchy.

A second practical feature of these deployments is *context dependence*. The same delegated task—"summarize a document," "execute a database query," "choose a pricing action," "allocate compute budget"—can have sharply different payoffs depending on observable side information: user type, market regime, safety tier, latency constraints, or regulatory jurisdiction. In contemporary toolchains, such context is often shared (or at least inferable) across components. Yet most incentive-learning abstractions either ignore context or treat it implicitly through nonstationarity. Our premise is that making context explicit is not a cosmetic refinement: it is the difference between an incentive scheme that is reliably stabilizing and one that is brittle, overpays, or fails precisely in rare but high-stakes regimes.

The core difficulty is easy to state informally. Consider a principal who can recommend an action to each child agent, and can attach a transfer (a monetary payment, a quota, a compute credit, a reputation score, or any other scarce resource) conditional on compliance. The principal observes whether the child complied, but does not observe the child's latent preferences, internal reward, or learning state. At the same time, the principal's own reward depends on the joint action profile along the tree. Without carefully designed transfers, the principal's recommendation is merely cheap talk: a self-interested child will deviate whenever the locally optimal action differs from the recommendation. With transfers, the principal can in principle "buy" compliance, but only if the payment is calibrated to the child's *opportunity cost* in the current context. Overpaying burns budget and can distort exploration incentives; underpaying fails to induce compliance and can render upstream learning meaningless because the observed outcomes correspond to off-policy, endogenous responses.

These issues are amplified by the fact that every node is simultaneously a principal and an agent. A mid-level agent faces its own parent's contract while simultaneously contracting with its children. If incentives are misestimated at one layer, deviations below can corrupt the reward signal above, leading to a feedback loop: the parent responds to noisy, noncompliant behavior by changing recommendations, which changes the child's learning environment, which further destabilizes behavior. In practice this manifests as oscillatory policies, intermittent noncompliance, and an excessive reliance on conservative, high transfers that guarantee short-run compliance but destroy long-run efficiency. Our goal is to formalize this tradeoff and provide a decentralized learning procedure whose regret guarantees match the reality of layered delegation.

We therefore study a contextual version of bandit learning in a rooted principal–agent tree. Each round features an observed context and stochastic local rewards. Each principal can offer each child a simple action-contingent transfer and can observe the child's realized action. This contract space is intentionally minimal: it matches what many deployed systems can implement (bonuses for following a recommended plan, credits for using an approved tool, penalties for unsafe actions) while remaining analyzable. The model highlights three failure modes that practitioners repeatedly confront. First, *decision error*: even if compliance were guaranteed, a principal may recommend suboptimal actions because it is still learning a contextual mapping from information to choices. Second, *subsidy burn*: to induce compliance, the principal may pay more than necessary due to uncertainty about the child's opportunity cost. Third, *noncompliance*: if payments are too low, or if the child's learner has not stabilized, the child may deviate, and the principal's observed reward becomes a biased sample from the wrong action profile. Any meaningful performance guarantee must control all three components simultaneously.

Our starting point is an economic observation with algorithmic consequences: under action observability, the minimal transfer that induces a child to take a recommended action is exactly the child's context-dependent utility gap between its best alternative and the recommended action. This is the familiar "pay the opportunity cost" logic, but here it must be defined recursively because a child's own utility depends on the behavior of its descendants. Once continuation values are defined, the transfer has a closed form, and (under an additive separability condition) the principal's optimization decomposes across children rather than requiring a combinatorial search over joint child action profiles. This decomposition is not merely a convenience: it is what allows decentralized learning to scale with the branching factor without enumerating K^B possibilities at each node.

Context enters in two distinct places. Economically, the child's opportunity cost depends on context, so a single global transfer level cannot robustly induce compliance across regimes; instead, the principal must learn a *context-indexed* transfer schedule. Statistically, learning these schedules requires repeated encounters with each context. We therefore focus on a finite-context baseline that captures common engineering practice (contexts as discrete tags, buckets, or task types) and allows uniform high-probability guarantees via concentration and union bounds. This choice makes the role of the minimum context probability explicit: if a context is too rare, no algorithm can reliably estimate the needed incentives, and any global guarantee must degrade. We view this as a feature rather than a limitation, because it mirrors real deployments where tail contexts are precisely where policies are least validated.

Our main contribution is a fully decentralized algorithmic template, *Contextual-MAIL*, that couples incentive learning with contextual bandit learning on *shifted* rewards. At a high level, each principal uses observed child compliance to estimate, for each context and recommended action, the minimal transfer needed to make that recommendation individually optimal for the child. These estimates are then used to transform the principal's own learning problem into a standard contextual bandit: the principal chooses its action and recommended child actions to maximize an induced utility that subtracts the estimated inducing payments. As estimates improve, recommended actions become increasingly incentive compatible, deviations become rare, and the upstream bandit problems become well behaved. The algorithm is modular: it only requires that each agent run a contextual no-regret learner that satisfies a suitable high-probability action-regret condition after some warmup, which aligns with how many deployed learners are monitored and tuned.

On the theory side, we provide a set of structural results that make this approach precise. We establish (i) closed-form optimal contextual incentives and, under separability, a decomposition of the parent's induced objective into independent maximizations across children; (ii) a contextual welfare telescoping identity showing that the sum of decentralized induced objectives aligns with the centralized welfare benchmark context-by-context, clarifying why transfers can be interpreted as internal shadow prices; (iii) a regret decomposition that isolates action learning, overpayment, and deviation losses, thereby mapping theoretical error terms to operational failure modes; and (iv) a transfer-estimation guarantee under repeated contexts, using only the principal's observable signals (context, offered transfer, and compliance inferred from the child's action). Combining these ingredients yields a layer-wise induction argument proving that, under our assumptions, every node achieves sublinear contextual pseudo-regret and the overall social-welfare regret is sublinear as well, with rates scaling polynomially in the natural problem parameters.

We also emphasize what the model does *not* claim. Action observability is strong in some environments and natural in others; it corresponds to audit logs, tool-use traces, or cryptographic attestations, and without it the principal must infer compliance indirectly, which changes the incentive problem. Separability is an approximation: it rules out certain cross-child complementarities that arise, for example, when two downstream tools must be coordinated. Likewise, the finite-context baseline abstracts away from continuous covariates and representation learning. We adopt these assumptions deliberately to isolate the delegation-and-incentives mechanism and to obtain guarantees that are interpretable and implementable. In settings where these assumptions are violated, our framework still suggests diagnostic quantities (opportunity-cost gaps, payment errors, deviation frequencies) that can guide robustification.

Roadmap. In the next section we position our approach relative to principal–agent online learning and contextual bandits, and to recent work on steering learning agents in delegated ML toolchains. We then formalize the model and equilibrium notion, derive the closed-form incentive characterization and the welfare telescoping identity, and develop the regret decomposition that motivates Contextual-MAIL. Finally, we present the contextual transfer-estimation procedure, prove the layer-wise no-regret and welfare no-regret theorem, and discuss how the comparative statics clarify when hierarchical delegation is statistically feasible versus when it inevitably requires conservative, costly incentives.

2 Related work

Our problem sits at the intersection of three literatures that rarely speak the same technical language: principal–agent theory (with its focus on incentive compatibility and continuation values), online learning (with its focus on bandit feedback and regret), and systems-oriented work on delegated ML and agent toolchains (with its focus on modular architectures and observability constraints). We briefly position our contribution relative to each, emphasizing where existing models either abstract away from hierarchy, from context, or from the fact that the “agents” we contract with are themselves adaptive learners.

Principal–agent models under learning and limited feedback. Classical contract theory studies incentive compatibility when the agent has private information or unobservable effort, typically in static or discounted dynamic settings **??**. In those models, the principal chooses a contract to induce a best response, and the comparative statics describe how information frictions shape risk sharing and effort provision. The online-learning analogue replaces equilibrium comparative statics with finite-time performance: the principal does not know the agent’s payoff function and must learn how much incentive is required, while the agent may also be learning. Recent work on *incentivized bandits*, *strategic arms*, or *bandits with payments* formalizes variants of this tension, where a platform pays users (or arms) to select certain actions and the platform observes only partial feedback **??**. In many such formulations, the strategic entity is myopic (or has a simple threshold response), and the platform’s objective is to steer behavior while learning reward parameters.

Our setting differs in two ways that materially change the structure of feasible algorithms. First, incentives are *recursive*: a child agent's opportunity cost depends on how it in turn incentivizes its own children, so the object that substitutes for a static "type" is a context-dependent continuation utility. Second, incentives must be computed and learned *locally* on a tree rather than with a single principal interacting with many independent agents. This combination makes naive reductions to standard Stackelberg learning insufficient: even if each edge in the tree is a principal–agent pair, the learning dynamics couple across layers because miscalibrated transfers change the distribution of downstream actions and thus the upstream reward process.

Online mechanism design and Stackelberg learning. A related thread studies online mechanism design where a leader commits to prices, bonuses, or allocation rules while learning demand or preferences, sometimes with strategic responses and sometimes with adversarially chosen types **??**. In the learning-in-games literature, one also finds regret-based convergence analyses for repeated Stackelberg or bilevel interactions, often under strong smoothness or monotonicity assumptions **?**. These works clarify that when both sides adapt, stability depends on how quickly incentives and actions converge. We adopt a complementary stance: rather than prove equilibrium convergence in a general dynamic game, we engineer a contract form (action-contingent transfers under action observability) for which the minimal inducing transfer has a closed form, enabling a clean separation between (i) estimating opportunity-cost gaps and (ii) running a contextual bandit on the induced objective. This is closer in spirit to mechanism design "by construction": we restrict the contract space to obtain a tractable incentive rule that can be learned with bandit feedback.

Contextual bandits and structured generalization. On the learning side, our benchmark is the contextual bandit model **?**. Most contextual bandit analyses assume that the learner observes a context c_t and chooses an action to maximize expected reward, with regret measured against the best context-dependent policy in a class. Extensions cover linear and generalized linear models, kernelized and nonparametric settings, and representation learning **??**. Our baseline takes $\mathcal{C}$ to be finite, which matches common engineering practice where "contexts" are discrete tags (task type, safety tier, jurisdiction) and yields uniform high-probability guarantees via concentration and union bounds. The key difference from standard contextual bandits is that, for a principal, the effective reward of a recommendation depends on whether children comply; thus the principal does not directly observe samples from the induced objective it wishes to optimize until incentives are sufficiently accurate. In that sense, our principal faces a contextual bandit with *endogenous action realization*, where the realized action profile is itself a strategic response to a transfer.

The closest purely bandit-theoretic relatives are models with corrupted actions, compliance noise, or bandits with "intermediate" decision makers. However, in those models the noise is usually exogenous, whereas in our hierarchy deviations are both systematic (driven by utility gaps) and learnable (shrinking as transfers are estimated). This distinction motivates our regret decomposition into action-learning, payment, and deviation terms: it aligns the statistical analysis with the economic reasons an induced reward sample may be biased.

Steering no-regret learners and incentive shaping. A growing literature asks how to influence an adaptive agent who runs a no-regret algorithm, through reward shaping, subsidies, or information design. Examples include "teaching" a learner, steering dynamics in repeated games, and manipulating feedback to induce desirable equilibria **??**. In many cases, the designer can modify the learner's loss function or observation model, and the technical goal is to bound the cost of steering while ensuring convergence to a target action. Our framework is aligned with this goal but differs in the available control channel: we assume the principal can only offer action-contingent transfers and observe actions (not internal gradients or losses). This restriction reflects deployed systems where an upstream orchestrator can provide credits or penalties conditional on logged behavior, but cannot reliably inspect or override a downstream model's internal updates. The assumption that each agent satisfies a high-probability action-regret condition after warmup is also consistent with practice: many learners are monitored to ensure stable no-regret behavior (or are wrapped by conservative exploration controllers) before being placed in a critical delegation loop.

Delegated ML, toolchains, and hierarchical control. In systems and applied ML, hierarchical delegation appears as hierarchical reinforcement learning, modular policy stacks, tool-use agents, and multi-agent pipelines **??**. A central theme is compositionality: higher-level modules set subgoals or choose tools, while lower-level modules execute. Our model shares this architecture but emphasizes incentive alignment rather than purely algorithmic decomposition. In many modern deployments, contracts are not monetary but take the form of compute budgets, latency allowances, access permissions, or reputation scores. These are naturally modeled as transfers because they are scarce resources that can be conditioned on compliance. Action observability corresponds to audit logs and tool-use traces, which are increasingly standard for safety and debugging; our analysis therefore highlights a practical lesson: without reliable observability, the principal must infer compliance indirectly, which fundamentally changes the identification problem for incentives.

Comparison to the tree bandit-with-transfers baseline. Our work is directly inspired by recent "bandit with transfers" models on trees, where a principal can incentivize children via minimal payments derived from continuation values, and separability yields a decomposition that avoids enumerating K^{B+1} joint actions. Relative to that baseline, we make three substantive extensions. First, we introduce *explicit contexts* $c \in \mathcal{C}$ and require transfers $\tau^{\star}_{w}(c, b)$ to be learned and applied per context; this brings the theory closer to systems in which the same action can have different safety or performance implications across regimes. Second, we provide a transfer-estimation view that uses only the principal's observable signals—$(c_t, \tau_t(w), A^w_t)$—and exploits repeated occurrences of the same context to perform threshold-style refinement. Third, we connect the economic structure to learning guarantees through a contextual regret decomposition that cleanly separates decision error, overpayment, and noncompliance. This decomposition is not only a proof device; it also offers operational diagnostics for practitioners (e.g., whether inefficiency is driven by exploration, miscalibrated subsidies, or residual deviations).

Limitations and adjacent directions. Finally, we note two gaps where existing literatures suggest natural next steps. First, finite contexts are analytically convenient but restrictive; extending our transfer estimation and decomposition to rich context classes would connect to representation learning in contextual bandits and to empirically relevant continuous covariates. Second, action observability is strong; relaxing it would bring our model closer to classical moral hazard and to partial-monitoring bandits, but would require new identification arguments because compliance could no longer be directly inferred from actions. We view these as fruitful directions precisely because they force a tighter integration of economic observability constraints with online learning guarantees.

3 Model

We study a hierarchical delegation environment in which decision making and learning occur locally, but outcomes are coupled through a principal–agent tree. Formally, players are the nodes of a rooted tree $\mathcal{T} = (V, E)$ with depth D and branching factor B. For a node $v \in V$, let $P(v)$ denote its parent (undefined for the root) and $C(v)$ its set of children; leaves satisfy $C(v) = \emptyset$. We index depth so that leaves sit at depth 1 and the root at depth D. Each player v has a finite action set A_v of size $|A_v| = K$ (we allow heterogeneity but keep K for notational clarity).

Contexts. Interaction unfolds over rounds $t = 1, \dots, T$. At each round, a context $c_t \in \mathcal{C}$ is realized and (in our baseline) observed by all players before any contracts or actions are chosen. We take $\mathcal{C}$ to be a finite set of size $M = |\mathcal{C}|$, and we assume contexts are drawn i.i.d. from an unknown distribution ρ satisfying a uniform support condition

$$\min_{c \in \mathcal{C}} \rho(c) \; \geq \; p_{\min} \; > \; 0.$$

This "no rare contexts" condition is not merely technical: because both action learning and transfer calibration are context-indexed, a context that appears only $o(\log T)$ times cannot support uniform high-probability guarantees. In deployed delegated systems, $\mathcal{C}$ can be interpreted as a discrete set of regimes (task type, risk tier, jurisdictional setting, or customer class) for which one is willing to maintain separate incentive and action policies.

Timing and contracts. Each round t proceeds as follows. After observing c_t, principals move from the root downward: each node v offers to each child $w \in C(v)$ a contract consisting of (i) a recommended action $B_t^w \in A_w$ and (ii) a nonnegative transfer $\tau_t(w)$ that is paid *only if* the child plays the recommendation. After observing its parent's contract, each node v chooses its action $A_t^v \in A_v$. Finally, rewards are realized, parents observe their children's actions, and transfers are settled. This is a deliberately narrow contract form—action-contingent transfers with observable actions—because it matches settings with audit logs or tool-use traces, and because it admits a clean "gap payment" characterization in hindsight (developed in Section 4) that we can then learn online.

Rewards and local coupling. Each node v receives a stochastic bandit reward $X_t^v \in [0, 1]$ with conditional mean

$$\mathbb{E}\Big[X_t^v \mid c_t, A_t^v, A_t^{C(v)}\Big] \; = \; \theta_v\Big(c_t, A_t^v, A_t^{C(v)}\Big) \, ,$$

where $A_t^{C(v)} = (A_t^w)_{w \in C(v)}$. We assume the realized reward decomposes as

$$X_t^v \; = \; \theta_v(c_t, A_t^v, A_t^{C(v)}) \; + \; z_t^v, \qquad \mathbb{E}[z_t^v \mid \mathcal{F}_{t-1}] = 0,$$

with z_t^v conditionally sub-Gaussian (and uniformly boundedness of X_t^v can be viewed as a convenient normalization). Importantly, v's reward depends only on its own action and its children's actions: siblings do not directly interact. This is the local-dependence structure implied by the tree and is what makes backward, layer-wise analysis feasible.

Additive separability across children. A central modeling assumption is that θ_v is additively separable across children:

$$\theta_v(c, a_v, a_{C(v)}) \;=\; g_v(c, a_v) \;+ \sum_{w \in C(v)} h_{v,w}(c, a_v, a_w). \tag{1}$$

The term $g_v(c, a_v)$ captures the component of v's performance attributable to its own action in context c, while each $h_{v,w}(c, a_v, a_w)$ captures the incremental effect of child w's action on v, potentially moderated by v's action. We emphasize what separability does and does not assume. It does not require that children are independent learners, nor that their rewards are independent, nor that $h_{v,w}$ is small; it only rules out *interaction terms among multiple children* in the parent's reward. This restriction is exactly what prevents the parent's contract choice from becoming combinatorial in B: without (1), even a myopically optimal set of recommendations would generally require evaluating K^B joint child action profiles for each parent action a_v, which is prohibitive in large branching systems.

Utilities with transfers. Transfers shift incentives but do not create or destroy intrinsic task reward. We model each node's per-round utility as its realized reward plus any transfer it receives from its parent minus any transfers it pays to its children:

$$U_t^v \;=\; X_t^v \;+ \mathbf{1}\{A_t^v = B_t^v\}\, \tau_t(v) \;- \sum_{w \in C(v)} \mathbf{1}\{A_t^w = B_t^w\}\, \tau_t(w), \tag{2}$$

where the term $\tau_t(v)$ is absent for the root (which has no parent), and $\mathbf{1}\{\cdot\}$ denotes the indicator function. This accounting mirrors practical delegation pipelines in which resources (compute credits, access privileges, queue priority) can be passed down conditional on compliance. Equation (2) also makes clear why we distinguish between *social welfare* and *private utility*: transfers are internal to the hierarchy, so they reallocate utility across nodes but, in the baseline, they do not enter the welfare objective.

Observability and feedback. The informational assumptions combine two features. First, *action observability*: each parent observes each child's realized action A_t^w. This is the key that allows contracts of the form "play b and receive τ." Second, *bandit feedback*: each node observes only its own realized reward X_t^v, not the rewards of others. Thus, while a parent can audit compliance (actions), it cannot directly infer a child's continuation value from observed rewards, because the child's reward process is private. Learning must therefore proceed through local reward samples and through observed compliance as a function of offered transfers.

Welfare benchmark. Given a context c and a joint action profile $a = (a_v)_{v \in V}$, define the (mean) welfare as

$$W(c,a) \; := \; \sum_{v \in V} \theta_v\big(c, a_v, a_{C(v)}\big) \, .$$

Transfers do not appear in W because they cancel in the aggregate when summed over nodes (every payment is a receipt), so welfare captures the intrinsic quality of the delegated action profile. Over T rounds, the realized welfare is $\sum_{t=1}^{T} \sum_{v \in V} X_t^v$, and our social benchmark is the clairvoyant contextual planner that, for each realized context c_t, selects the welfare-maximizing joint action profile $a^\star(c_t) \in \arg\max_a W(c_t, a)$. We measure social-welfare regret as

$$SWReg(T) \; := \; \sum_{t=1}^{T} \Big(\max_{a \in \prod_{v \in V} A_v} W(c_t, a) \; - \; W(c_t, A_t) \Big),$$

where $A_t = (A_t^v)_{v \in V}$ is the realized action profile induced by contracts and strategic responses.

Individual performance and the role of induced objectives. Each node v is itself a strategic learner: it chooses actions to maximize its own cumulative utility $\sum_{t=1}^{T} U_t^v$ given its incoming contract (if any), its outgoing contracts (if any), and its reward feedback. From the perspective of learning guarantees, we evaluate each node by a contextual pseudo-regret $R_v(T)$ against the best context-dependent decision rule available to v *under the contracting protocol.* The subtlety is that a principal's effective payoff from an action is mediated by whether children comply, and compliance depends on the transfers offered. This motivates introducing, in Section 4, a recursive induced-utility object $\mu_v^\star(c, a_v)$ that internalizes optimal downstream recommendations and the minimal transfers needed to implement them in context c. Once $\mu_v^\star$ is defined, the principal's learning problem becomes a contextual bandit on a shifted reward scale, while the transfer-learning problem becomes an estimation task for the opportunity-cost gaps that make recommendations incentive compatible.

Why the model is algorithmically tractable. The tree structure limits payoff externalities to parent–child edges, and separability (1) removes cross-child complementarities in a parent's reward. Together, these two features ensure that an upstream node never needs to enumerate K^{B+1} joint action profiles to decide what to recommend. Instead, once we express incentives in terms of continuation values, each child can be handled via an independent maximization and a corresponding gap payment. This is the economic source of our polynomial dependence on (K, B, D, M): rather than solving a global mechanism design problem, we exploit the tree to compute and learn contracts locally, and we use the repeated occurrence of contexts to calibrate those local contracts context by context. Section 4 formalizes this logic by defining continuation utilities recursively and showing that optimal contextual incentives admit a closed form in hindsight, together with a welfare telescoping identity that links decentralized induced objectives to the planner's benchmark.

4 Optimal contextual incentives in hindsight

We now characterize, for a fixed context c, what a principal would like to recommend to its children and how much it must minimally pay to implement those recommendations when all continuation values are known. This "in-hindsight" analysis plays two roles. Economically, it isolates the exact object that transfers must compensate: a child's opportunity cost of complying in that context. Algorithmically, it provides the target that our online procedure in Section 5 will estimate and then plug into a standard contextual bandit routine on appropriately shifted rewards.

Continuation utilities and induced objectives. Because contracts are offered and accepted (via action choice) locally along edges, the relevant object for node v is not its raw mean reward θ_v, but its *continuation utility* after optimally contracting with its descendants. We define these objects pointwise in the context c, working backward from the leaves.

For a leaf ℓ (so $C(\ell) = \emptyset$), there are no downstream contracts, hence the continuation utility coincides with its mean reward:

$$\mu_\ell^\star(c, a_\ell) \;:=\; \theta_\ell(c, a_\ell).$$

For an internal node v, fix an own action $a_v \in A_v$ and a vector of recommendations $b_{C(v)} = (b_w)_{w \in C(v)}$. If each child w complies with b_w, then v's expected reward is $\theta_v(c, a_v, b_{C(v)})$. To induce compliance, v must offer transfers along each edge (v, w). Let $\tau_w(c, b_w) \geq 0$ denote the transfer offered to child w conditional on playing b_w. Given these offers, child w chooses its action to maximize its own continuation utility plus any transfer received, i.e.,

$$a_w \in \arg\max_{a \in A_w} \Big\{ \mu_w^\star(c, a) \;+\; \mathbf{1}\{a = b_w\}\, \tau_w(c, b_w) \Big\}.$$

This formulation makes explicit why we define $\mu_w^\star$ *excluding* incoming transfers: transfers are contract-dependent and are added only for the recommended action. It is precisely this separation that will allow us to identify transfers as "gap payments."

Given the child's best-response condition, the principal's objective is naturally expressed in terms of an *induced utility* that subtracts the transfers needed for implementation. Define

$$\mu_v(c, a_v, b_{C(v)}) \;:=\; \theta_v(c, a_v, b_{C(v)}) \;-\; \sum_{w \in C(v)} \tau^\star_w(c, b_w), \tag{3}$$

where $\tau^\star_w(c, b_w)$ is the minimal transfer that makes b_w optimal for w in context c, given w's continuation utilities. Node v's best induced utility for action a_v is then

$$\mu^\star_v(c, a_v) \;:=\; \max_{b_{C(v)} \in \prod_{w \in C(v)} A_w} \;\mu_v(c, a_v, b_{C(v)}). \tag{4}$$

By construction, $\mu^\star_v(c, a_v)$ is the relevant benchmark for v's action learning: it is the best payoff v can secure from choosing a_v when it can optimally recommend and minimally incentivize its children.

Minimal inducing transfers are opportunity-cost gaps. Under action observability, inducing a child to take a particular action is equivalent to making that action a best response under a simple action-contingent bonus. The next characterization is therefore intuitive: to implement b, the parent must compensate the child for the utility it forgoes relative to its best alternative in that context.

Fix a child w, context c, and target action $b \in A_w$. The incentive-compatibility inequalities for inducing b are

$$\mu^\star_w(c, b) \;+\; \tau_w(c, b) \;\geq\; \mu^\star_w(c, a) \qquad \text{for all } a \in A_w,$$

which are satisfied by the minimal transfer

$$\tau^\star_w(c, b) \;:=\; \max_{a \in A_w} \mu^\star_w(c, a) \;-\; \mu^\star_w(c, b). \tag{5}$$

Two remarks are worth emphasizing for later learning. First, $\tau^\star_w(c, b)$ is always nonnegative and equals zero if and only if b is itself a best action for w under $\mu^\star_w(c, \cdot)$. Second, (5) depends on c: in practice, a recommendation that is cheap to implement in one regime (say, a low-risk task type) may be expensive in another (high-risk), because the child's continuation values shift with the context.

Separability yields a decomposable parent objective. The definition (4) is general, but computing $\mu^\star_v(c, a_v)$ naively would require maximizing over $K^{|C(v)|}$ recommendation profiles. This is exactly where our separability assumption becomes economically and computationally meaningful. With

$$\theta_v(c, a_v, a_{C(v)}) \;=\; g_v(c, a_v) \;+\; \sum_{w \in C(v)} h_{v,w}(c, a_v, a_w),$$

the induced utility (3) becomes

$$\mu_v(c, a_v, b_{C(v)}) \;=\; g_v(c, a_v) \;+\; \sum_{w \in C(v)} \Big(h_{v,w}(c, a_v, b_w) \;-\; \tau^\star_w(c, b_w) \Big),$$

and therefore the maximization over $b_{C(v)}$ decomposes across children:

$$\mu^\star_v(c, a_v) \;=\; g_v(c, a_v) \;+\; \sum_{w \in C(v)} \max_{b \in A_w} \Big\{ h_{v,w}(c, a_v, b) \;-\; \tau^\star_w(c, b) \Big\}. \tag{6}$$

This is the key tractability statement: rather than solving a combinatorial contract problem, the parent solves B independent K-ary problems (one per child) for each candidate a_v. In delegated systems with many submodules, (6) is the difference between a local procedure that can run at the edge and a centralized optimization over joint profiles that is operationally infeasible.

A welfare telescoping identity (context by context). The preceding objects may appear "private"—they are defined from each agent's continuation values and involve transfers—yet they line up exactly with the social planner's welfare benchmark. The alignment is clearest when expressed as a telescoping identity that holds separately for each context c.

Let $W(c, a) = \sum_{v \in V} \theta_v(c, a_v, a_{C(v)})$ denote mean welfare. Then, for every fixed context c,

$$\sum_{v \in V} \max_{a \in A_v} \mu^\star_v(c, a) \;=\; \max_{a \in \prod_{v \in V} A_v} W(c, a). \tag{7}$$

The economic content of (7) is that the gap payments $\tau^\star_w(c, \cdot)$ act like shadow prices: they decentralize the planner's problem down the tree without distorting the welfare objective, because transfers are internal and the opportunity-cost terms cancel when aggregated appropriately.

A compact way to see the cancellation is to interpret $\max_{a \in A_v} \mu^\star_v(c, a)$ as the *incremental welfare* created at node v relative to giving each child its own best attainable continuation value. Formally, define the planner's optimal welfare for a subtree rooted at v,

$$\mathrm{OPT}_v(c) \;:=\; \max_{(a_u)_{u \in \mathrm{subtree}(v)}} \sum_{u \in \mathrm{subtree}(v)} \theta_u(c, a_u, a_{C(u)}),$$

with $\mathrm{OPT}_\ell(c) = \max_{a_\ell} \theta_\ell(c, a_\ell)$ at leaves. One can verify by backward induction that

$$\max_{a_v \in A_v} \mu^\star_v(c, a_v) \;=\; \mathrm{OPT}_v(c) \;-\; \sum_{w \in C(v)} \mathrm{OPT}_w(c),$$

because $\tau^\star_w(c, b)$ subtracts exactly the gap between child w's best attainable continuation value $\mathrm{OPT}_w(c)$ and the value under the implemented action.

Summing this equality over all $v \in V$ telescopes: every $\mathrm{OPT}_w(c)$ appears once with a plus sign (as $\mathrm{OPT}_v(c)$ when $v = w$) and once with a minus sign (inside its parent's subtraction), leaving only $\mathrm{OPT}_{\mathrm{root}}(c)$, which equals $\max_a W(c, a)$. This establishes (7).

What this buys us, and what it does not. Equations (5)–(7) pin down the *right* targets for learning: minimal context-dependent transfers are opportunity-cost gaps, and induced objectives align with welfare when aggregated. At the same time, these are hindsight characterizations: $\mu_w^\star(c, \cdot)$ is not directly observed by the parent, and the parent only sees compliance (via actions) and its own bandit rewards. Section 5 addresses precisely this gap by showing how repeated contexts and local exploration allow each principal to estimate $\tau_w^\star(c, b)$ from acceptance behavior, and then to run a contextual bandit algorithm on the shifted reward signals implied by (6).

5 Contextual-MAIL: warm-up, context-indexed incentive estimation, and learning on shifted rewards

Our online problem is to approximate, using only local bandit feedback and observed compliance, the in-hindsight objects characterized in Section 4: the context-dependent gap transfers $\tau_w^\star(c, b)$ and the induced continuation utilities $\mu_v^\star(c, \cdot)$. Contextual-MAIL is the decentralized procedure that accomplishes this by modularizing learning along edges (to stabilize incentives) and at nodes (to learn context-dependent actions under the stabilized incentive system). The guiding design principle is practical: each principal should be able to operate using only information it naturally has in organizational and delegated-AI deployments—its own realized reward X_t^v, the realized context c_t, and the observed actions of its children.

A bounded search domain for transfers. Because each $X_t^v \in [0, 1]$ and the tree has finite depth D, continuation utilities are uniformly bounded: for every node v, context c, and action a, we have $\mu_v^\star(c, a) \in [0, D]$ under the backward definition. Consequently, every opportunity-cost gap satisfies

$$0 \;\le\; \tau_w^\star(c, b) \;=\; \max_a \mu_w^\star(c, a) - \mu_w^\star(c, b) \;\le\; D.$$

This elementary bound is operationally useful: it lets each principal run a finite, context-indexed threshold search over a known interval $[0, D]$ without any global calibration.

Warm-up: making compliance statistically meaningful. A basic tension appears immediately. To estimate $\tau_w^\star(c, b)$ from behavior, the parent must interpret whether the child accepted the proposed contract, i.e., whether $A_t^w = B_t^w$. But early in time the child is itself learning and may deviate for exploration even when the transfer would be sufficient in hindsight. We therefore include a warm-up period W_v for each node v (with W_v increasing with depth) during which contracts are deliberately simple and generous. Concretely, each parent plays an exploration routine over its own actions and (temporarily) offers high transfers to children for a small set of recommended actions, ensuring that deviations are rare enough that observed noncompliance is informative rather than dominated by the child's experimentation. This is the only role of warm-up: it creates a regime in which acceptance can be treated as an approximately thresholded response to the offered transfer, which is exactly the behavioral structure exploited by our incentive estimator. In deployments, this corresponds to an initial ramp-up period with conservative budgets and standardized playbooks before performance-based fine-tuning.

Edge module: context-indexed estimation of minimal inducing transfers. Fix an edge (v, w), a context c, and a candidate recommended action $b \in A_w$. The parent maintains an interval

$$\big[\underline{\tau}_w(c,b),\, \overline{\tau}_w(c,b)\big] \subseteq [0, D], \qquad \underline{\tau}_w(c,b) = 0,\ \overline{\tau}_w(c,b) = D \text{ initially,}$$

interpreted as a confidence bracket for $\tau_w^\star(c, b)$. When context c occurs, the parent occasionally schedules a *test* for (c, b): it recommends b and offers transfer λ (typically the midpoint $\lambda = (\underline{\tau} + \overline{\tau})/2$). The observation is binary and local: whether the child complied, $\mathbf{1}\{A_t^w = b\}$, which under action observability is directly observed by v.

Because the child's behavior may still have residual stochasticity, tests are batched. In a batch of size m consisting of rounds in which the same (c, b, λ) is offered, we define acceptance if compliance occurs on at least a $(1 - \eta)$-fraction of those rounds, where $\eta \in (0, 1/2)$ is a tolerance level. If accepted, we set $\overline{\tau}_w(c, b) \leftarrow \lambda$; if rejected, we set $\underline{\tau}_w(c, b) \leftarrow \lambda$. Iterating yields a batched binary search, run separately for each (c, b). The output used by the algorithm is a conservative estimate

$$\widehat{\tau}_w(c,b) \ :=\ \overline{\tau}_w(c,b),$$

so that whenever the bracket is valid, $\widehat{\tau}_w(c, b) \geq \tau_w^\star(c, b)$ and the offered transfer is sufficient to induce b (up to the child's residual learning error). The need for a one-sided estimate is not merely technical: from the parent's standpoint, underestimation risks noncompliance and propagating regret up the hierarchy, whereas overestimation only burns subsidy budget and can be controlled quantitatively in the regret decomposition of Section 6.

The module is fully local. It requires no knowledge of $\mu_w^\star$, no reports from the child, and no observation of downstream transfers; it uses only (c_t, λ) and the realized action A_t^w. The key statistical precondition is repeated contexts. With $|\mathcal{C}| = M$ and $\rho(c) \geq p_{\min}$, each context appears $\Theta(Tp_{\min})$ times, enabling each (c, b) threshold to be refined to accuracy ε_T by $O(\log(D/\varepsilon_T))$ batches, provided the batches are scheduled sparsely enough not to dominate the parent's action learning.

Node module: learning actions using shifted rewards. Once transfers are estimated well enough that compliance is likely, the parent can treat the induced utility $\mu_v(c, \cdot)$ as an unknown contextual reward function and apply a standard contextual bandit routine. Operationally, we implement this by learning on *shifted rewards* that subtract estimated incentive costs. On a round t with context c_t, after choosing own action A_t^v and issuing contracts $\{(B_t^w, \tau_t(w))\}_{w \in C(v)}$, node v observes its realized reward X_t^v and settles transfers. We define the shifted observation

$$\widetilde{X}_t^v \;:=\; X_t^v \;-\; \sum_{w \in C(v)} \mathbf{1}\{A_t^w = B_t^w\}\, \widehat{\tau}_w(c_t, B_t^w),$$

which is precisely the parent's realized payoff under the conservative payments $\widehat{\tau}$ (up to the fact that it may pay $\widehat{\tau}$ even when the true minimal transfer is lower). In expectation, when compliance holds, $\widetilde{X}_t^v$ is an unbiased sample of the induced utility with $\tau_w^\star$ replaced by $\widehat{\tau}_w$, so maximizing expected shifted reward approximates maximizing the in-hindsight objective.

For finite $\mathcal{C}$, a simple and effective implementation is to run M independent bandit learners, one for each context c, each producing an action-selection rule over A_v. Because the recommendation problem is separable across children, the per-round computation does not require enumerating K^{B+1} joint profiles. In particular, we parameterize the parent's unknown mean reward in context c as additive across children and treat each pair (a_v, b_w) as a primitive choice affecting one component of the sum. A convenient implementation uses a linear model per context: define a feature vector $x(c, a_v, a_{C(v)})$ that contains a one-hot coordinate for (c, a_v) and one-hot coordinates for each (c, w, a_v, a_w). Then

$$\mathbb{E}[X_t^v \mid c_t = c, A_t^v = a_v, A_t^{C(v)} = a_{C(v)}] \;=\; \langle \beta_{v,c},\, x(c, a_v, a_{C(v)}) \rangle$$

for an unknown $\beta_{v,c} \in \mathbb{R}^{K+BK^2}$. Using ridge regression (or any standard linear contextual bandit routine), node v forms an optimistic estimate of the induced payoff

$$\mathrm{UCB}_t^v(c, a_v, b_{C(v)}) \;:=\; \langle \widehat{\beta}_{v,c,t},\, x(c, a_v, b_{C(v)}) \rangle + \alpha_t \big\| x(c, a_v, b_{C(v)}) \big\|_{V_{v,c,t}^{-1}} - \sum_{w \in C(v)} \widehat{\tau}_w(c, b_w),$$

and chooses $(A_t^v, B_t^{C(v)})$ to maximize UCB_t^v. Crucially, due to separability of both the mean reward and the transfer term, this maximization decomposes:

for each candidate a_v, the best b_w can be selected independently across w, and the overall best a_v is found by scanning K options. Thus the per-round time at node v is $O(BK^2)$ (dominated by evaluating K own actions and K child actions per child), and the memory is $O(M(K + BK^2))$ for the per-context sufficient statistics. This is polynomial in (M, B, K) and avoids the exponential K^{B+1} explosion.

Putting the modules together in a decentralized protocol. Contextual-MAIL is run simultaneously at all nodes. At each round, the root initiates contract announcements; each internal node v, upon receiving its parent's contract, (i) chooses its own action and its contracts to children using its node module and current $\widehat{\tau}$ values, (ii) schedules occasional edge tests for its children to refine $\widehat{\tau}$ in the realized context, and (iii) updates its bandit state using $\widetilde{X}_t^v$ after rewards realize. No messages about rewards or beliefs are exchanged; the only communication is the contract pair $(B_t^w, \tau_t(w))$ on each edge.

Limitations and implementation notes. The finite-context design is intentionally conservative: it trades function approximation for transparent, context-indexed accounting of incentives and samples. When M is large or contexts are continuous, a practical extension is to replace per-context tables by function classes (e.g., generalized linear models) for both $\widehat{\tau}_w(\cdot, b)$ and $\mu_v^\star(\cdot, a)$, at the cost of additional approximation error. More subtly, conservative overestimation of $\tau^\star$ is stabilizing but can be expensive early on; in budget-constrained settings one can cap transfers and accept a controlled deviation probability, which our theory in Section 6 will capture as an explicit deviation-regret term.

6 Theory: contextual regret decomposition and layer-wise propagation

We now provide the analytical backbone behind Contextual-MAIL. The core difficulty is that each node simultaneously (i) learns which actions are optimal *given* the incentive system it is implementing, and (ii) learns the incentive system itself through transfer estimation that relies on observed compliance. Our analysis therefore separates the sources of inefficiency into three interpretable components—wrong actions, excessive payments, and residual deviations—and then shows that these components can be controlled locally and propagated up the hierarchy without exploding in D.

Benchmark and pseudo-regret. Fix a node v. In each round t, node v observes context c_t and (possibly after receiving a contract from its parent) chooses its own action A_t^v and, if internal, recommendations $B_t^{C(v)}$ and transfers $\tau_t(\cdot)$ for its children. The natural contextual benchmark is the best induced continuation utility achievable by v in that context:

$$\max_{a \in A_v} \mu_v^\star(c_t, a),$$

where $\mu_v^\star$ is defined recursively using optimal (minimal) inducing transfers $\tau_w^\star$. We measure node-level pseudo-regret over horizon T as

$$R_v(T) \; := \; \sum_{t=1}^{T} \Big(\max_{a \in A_v} \mu_v^\star(c_t, a) \; - \; \mu_v^\star(c_t, A_t^v) \Big),$$

with the understanding that $\mu_v^\star(c_t, A_t^v)$ is the benchmark induced utility of the realized own action, evaluated under optimal downstream recommendations and minimal incentives. This isolates the learning problem faced by v: it should act as if it were maximizing $\mu_v^\star(c, \cdot)$ in each context.

A three-term decomposition: action, payment, deviation. The algorithm does not directly optimize $\mu_v^\star$, because it does not know $\tau^\star$ nor does it perfectly enforce compliance. The first step is an add-and-subtract argument that yields a decomposition aligned with implementation concerns:

$$R_v(T) \; \le \; R_v^{\mathrm{ac}}(T) \; + \; R_v^{\mathrm{pay}}(T) \; + \; R_v^{\mathrm{dev}}(T),$$

where the three terms correspond to (a) action selection error in the shifted bandit problem, (b) overpayment relative to minimal inducing transfers, and (c) welfare loss due to realized noncompliance.

Formally, let the algorithm's per-round shifted payoff be

$$\widetilde{X}_t^v \; = \; X_t^v \; - \sum_{w \in C(v)} \mathbf{1}\{A_t^w = B_t^w\} \, \widehat{\tau}_w(c_t, B_t^w),$$

and define the corresponding shifted mean (conditional on the realized recommendations) by

$$\widetilde{\theta}_v(c_t, A_t^v, B_t^{C(v)}) \; := \; \theta_v(c_t, A_t^v, B_t^{C(v)}) \; - \sum_{w \in C(v)} \widehat{\tau}_w(c_t, B_t^w).$$

On rounds where all children comply, $\widetilde{X}_t^v$ is an unbiased observation of $\widetilde{\theta}_v(\cdot)$ up to sub-Gaussian noise, so the node module can be analyzed as a standard contextual bandit on $\widetilde{\theta}_v$. The gap between $\widetilde{\theta}_v$ and the desired induced utility $\mu_v(\cdot)$ is precisely the payment error $\widehat{\tau} - \tau^\star$, while the gap between recommendations and realized actions appears only on deviation rounds. One convenient instantiation is:

$$\begin{aligned}
R_v^{\mathrm{ac}}(T) \; &:= \; \sum_{t=1}^{T} \Big(\max_{a \in A_v} \widetilde{\mu}_v^{\star}(c_t, a) \; - \; \widetilde{\mu}_v^{\star}(c_t, A_t^v) \Big), \\
R_v^{\mathrm{pay}}(T) \; &:= \; \sum_{t=1}^{T} \sum_{w \in C(v)} \Big(\widehat{\tau}_w(c_t, B_t^w) - \tau_w^{\star}(c_t, B_t^w) \Big), \\
R_v^{\mathrm{dev}}(T) \; &:= \; \sum_{t=1}^{T} \sum_{w \in C(v)} \mathbf{1}\{A_t^w \neq B_t^w\} \cdot \Delta_{v,w},
\end{aligned}$$

where $\widetilde{\mu}_v^{\star}$ is the induced utility computed using $\widehat{\tau}$ in place of $\tau^{\star}$, and $\Delta_{v,w}$ is a uniform upper bound on the one-step loss to v from child w's deviation (under bounded rewards, we may take $\Delta_{v,w} = O(1)$; with depth-D continuation values, $O(D)$ suffices). The decomposition is intentionally conservative: it treats any deviation as potentially worst-case, reflecting the operational reality that misalignment in a subteam can destroy the value of an upstream decision even if average performance remains high.

Uniform-in-time concentration and repeated contexts. To make the three terms simultaneously small, we require high-probability control that is uniform over time and over contexts. Two concentration steps are central.

First, because contexts are i.i.d. with $\rho(c) \geq p_{\min}$, standard multiplicative Chernoff bounds imply that with probability at least $1 - \delta$,

$$N_c(T) \; := \; \sum_{t=1}^{T} \mathbf{1}\{c_t = c\} \; \geq \; \tfrac{1}{2} T p_{\min} \qquad \text{for all } c \in \mathcal{C},$$

provided T exceeds a logarithmic threshold in $(M, 1/\delta)$. This event ensures that each context receives enough samples to support both transfer estimation and contextual action learning.

Second, conditional on sufficient repetitions of each context, we can analyze each per-context learner with standard self-normalized martingale tools (for linear models) or Hoeffding-style arguments (for finite-action bandits). In the linear instantiation described earlier, for each fixed context c, the design matrix $V_{v,c,t}$ concentrates, yielding a uniform confidence sequence:

$$\Big| \langle \widehat{\beta}_{v,c,t} - \beta_{v,c}, \, x(c, a_v, a_{C(v)}) \rangle \Big| \; \leq \; \alpha_t \| x(c, a_v, a_{C(v)}) \|_{V_{v,c,t}^{-1}}$$

simultaneously for all $t \geq 1$ and all action tuples, on an event of probability $1 - \delta$ after a union bound over $c \in \mathcal{C}$. This implies the usual $O(\sqrt{T})$-type cumulative action regret $\mathbb{E}[R_v^{\mathrm{ac}}(T)] = \tilde{O}(\sqrt{T})$, with the dependence on M entering through the number of parallel learners and the effective sample size $N_c(T)$.

For transfers, the acceptance/rejection batches yield one-sided confidence brackets that can be union-bounded over (w, c, b) and over the $O(\log(D/\varepsilon))$ binary-search stages. The key is that the algorithm outputs $\widehat{\tau} \geq \tau^{\star}$ on the high-probability event, turning compliance failures into a deviation term rather than contaminating the payment term with negative errors.

A propagation lemma across layers. The principal technical challenge is to prevent errors at depth $d-1$ from amplifying as they flow to depth d. We capture this with a layer-wise propagation inequality. Fix an internal node v. On the high-probability event where (i) the transfer estimates on edges (v, w) are accurate up to ε_T and one-sided, and (ii) each child $w \in C(v)$ satisfies its contextual action-regret condition after warm-up W_w, we can show:

$$\mathbb{E}\big[R_v(T)\big] \;\leq\; \mathbb{E}\big[R_v^{\mathrm{ac}}(T)\big] + O\left(\sum_{w \in C(v)} \varepsilon_T \, \mathbb{E}[N^{\mathrm{rec}}_{(v,w)}(T)]\right) + O\left(\sum_{w \in C(v)} \sum_{t=1}^{T} \mathbb{P}(A_t^w \neq B_t^w)\right),$$

where $N^{\mathrm{rec}}_{(v,w)}(T)$ is the number of times v recommends an action to w up to T. The first term is purely local (a contextual bandit regret bound for v under shifted rewards). The second term converts transfer estimation error into an additive payment regret proportional to recommendation frequency. The third term is the deviation channel: it depends on the child's behavior but only through the (observable) event of noncompliance. This structure is why warm-up and one-sided transfer estimates matter: they allow us to control the deviation probability by ensuring the recommended action is strictly optimal for the child up to its learning error.

Layer-wise no-regret and welfare no-regret. Combining the preceding ingredients yields our main performance guarantee. The proof proceeds by induction on depth: leaves have no incentive-learning burden and satisfy the assumed contextual action-regret condition; given the bound for all nodes below depth d, the propagation lemma plus concentration implies the bound for depth d. Summing the resulting node-level regrets and invoking the telescoping identity of Proposition 2 then gives social welfare regret.

In particular, under the stated assumptions (bounded rewards, action observability, separability, i.i.d. finite contexts with $p_{\min} > 0$, and high-probability contextual no-regret of agents after warm-up), if every node runs Contextual-MAIL with exploration schedules that allocate $O(\log T)$ tests per (c, b) while keeping the total testing mass $o(T)$, then for every node v,

$$\mathbb{E}[R_v(T)] \;=\; o(T),$$

and moreover the planner's contextual welfare regret satisfies

$$\mathbb{E}[SWReg(T)] := \mathbb{E}\left[\sum_{t=1}^{T}\left(\max_{a\in\prod_v A_v}\sum_{v\in V}\theta_v(c_t, a_v, a_{C(v)}) - \sum_{v\in V}\theta_v(c_t, A_t^v, A_t^{C(v)})\right)\right] = o(T).$$

The resulting rates scale polynomially in $(K, B, D, M, 1/p_{\min})$, reflecting three distinct scarcities: more actions increase both the child-gap search and the node bandit complexity, more children increase the number of edges to stabilize, and rarer contexts reduce the effective sample size for both modules.

What the theory does and does not capture. We emphasize two limitations that motivate the extensions in the next section. First, finite contexts and $p_{\min} > 0$ are strong but transparent: they guarantee repeated opportunities to "debug" incentives in each situation. Second, separability is not merely a convenience; it prevents the parent's recommendation problem from becoming a combinatorial contextual bandit over K^B joint child profiles. When separability is violated, the same decomposition remains conceptually valid, but the node module requires approximation or numerical optimization, and transfer learning may need to target joint deviations rather than single-agent thresholds.

7 Extensions and variants

The baseline analysis deliberately adopts two "debuggable" assumptions: a finite context space with $p_{\min} > 0$ ensuring repeated visits, and exact separability ensuring that each parent's recommendation problem decomposes into B independent subproblems. In deployments, neither assumption is sacrosanct. In this section we outline several extensions that preserve the economic logic of induced utilities and minimal transfers, while clarifying where we can still obtain clean guarantees and where numerical methods become unavoidable.

Continuous contexts via smoothness or linear structure. When $\mathcal{C}$ is large or continuous (e.g., feature vectors describing a task, a customer, or a local operating state), the per-context transfer estimation in Proposition 4 cannot be run independently for each c. A standard remedy is to assume regularity of the primitives in context. Two tractable regimes are:

(i) Lipschitz contexts. Assume $\mathcal{C} \subseteq \mathbb{R}^d$ and that, for each edge (v, w) and recommended action $b \in A_w$, the minimal inducing transfer $\tau_w^\star(c, b)$ is L-Lipschitz in c. Then we can replace context-indexed batches by adaptive partitioning (e.g., a zooming-style scheme) that refines regions of $\mathcal{C}$ where the algorithm frequently operates. Intuitively, rather than requiring $N_c(T) \gtrsim$

$Tp_{\min}$ for every context point, we require sufficient mass in each visited region and accept that rare regions will have coarser estimates and higher deviation risk. The economic interpretation remains: the transfer is still an opportunity cost, but now we borrow strength across similar contexts.

(ii) Linear (or generalized linear) contexts. Suppose $\tau_w^\star(c,b)$ is approximately linear in features $\phi(c)$, i.e.,

$$\tau_w^\star(c,b) \;\approx\; \langle \gamma_{w,b}, \phi(c) \rangle,$$

and similarly the induced utilities $\mu_v^\star(c,a)$ admit a linear representation. Then each node can run a contextual linear bandit for action selection on shifted rewards, while each principal estimates $\gamma_{w,b}$ from observed compliance. One convenient modeling choice is a *threshold response* for each child: conditional on context and the recommended action b, there exists a scalar threshold $\tau_w^\star(c,b)$ such that offering $\tau \geq \tau_w^\star(c,b)$ makes b optimal up to the child's learning error. Estimation can then proceed via one-sided regression or conservative upper-confidence bounds so that $\widehat{\tau}_w(c,b) \geq \tau_w^\star(c,b)$ holds with high probability, preserving the "overpay rather than under-incentivize" discipline that keeps deviation events interpretable.

In both regimes, the role of $p_{\min}$ is replaced by a *coverage* condition: we need that the realized contexts place enough probability mass in neighborhoods (Lipschitz) or excite features (linear) to identify both reward and transfer parameters. Practically, this maps to an operational requirement: if the environment never produces variation along a feature dimension, no contract-learning algorithm can reliably price it.

Approximate separability and structured interactions. Exact separability, $\theta_v(c, a_v, a_{C(v)}) = g_v(c, a_v) + \sum_{w \in C(v)} h_{v,w}(c, a_v, a_w)$, rules out complementarities among children (e.g., two subteams whose outputs must match). A more realistic model allows a residual interaction term:

$$\theta_v(c, a_v, a_{C(v)}) \;=\; g_v(c, a_v) \;+\; \sum_{w \in C(v)} h_{v,w}(c, a_v, a_w) \;+\; r_v(c, a_v, a_{C(v)}),$$

where r_v captures higher-order couplings. If $|r_v| \leq \eta$ uniformly, the induced-utility benchmark $\mu_v^\star$ is perturbed by at most $O(\eta)$ per round, so welfare regret bounds degrade additively by $O(\eta T)$ even if we continue to optimize the separable surrogate. This "graceful degradation" is often acceptable when separability is an approximation chosen for tractability.

However, once r_v is non-negligible, the parent's optimization over recommended actions becomes a contextual bandit over joint profiles $b_{C(v)} \in \prod_{w \in C(v)} A_w$, which is exponential in B. At that point, structure is decisive. If r_v has low treewidth when viewed as a factor graph over children actions, we can use message passing or dynamic programming to compute approximately optimal recommendations. If r_v is dense but smooth, one may resort to stochastic search (e.g., coordinate ascent over children, Monte Carlo tree search) embedded inside the node's decision rule. Economically, the interpretation of $\tau_w^\star$ as a marginal opportunity cost still holds, but marginal incentives may be insufficient to implement a globally optimal joint recommendation when complementarities create multiple local optima; numerical methods are then not a convenience but a necessity.

Limited transfer budgets and constrained incentive design. Many organizations face explicit constraints on incentive payments: a manager has a quarterly bonus pool, a public agency has appropriations, or a platform has a subsidy budget. To model this, we can impose either a per-round budget $\sum_{w\in C(v)} \tau_t(w) \le \mathcal{B}_v$ or a cumulative budget $\sum_{t\le T}\sum_{w\in C(v)} \tau_t(w) \le \mathcal{B}_v(T)$. The minimal-transfer formula remains informative, but it may no longer be feasible to induce the first-best action profile in every context.

A natural extension is a *primal–dual* variant of Contextual-MAIL: each principal maintains a Lagrange multiplier λ_v that prices budget consumption and chooses recommendations by maximizing a penalized induced utility,

$$\theta_v(c_t, a_v, b_{C(v)}) \;-\; \sum_{w\in C(v)} \widehat{\tau}_w(c_t, b_w) \;-\; \lambda_v \sum_{w\in C(v)} \widehat{\tau}_w(c_t, b_w),$$

updating λ_v online to satisfy the constraint. This makes the tradeoff explicit: we can guarantee no-regret relative to the best *budget-feasible* induced policy, but we should not expect vanishing welfare regret to the unconstrained planner benchmark unless budgets scale with T in a compatible way. From a policy standpoint, this extension clarifies how budget caps translate into predictable compliance shortfalls concentrated in contexts where opportunity costs are high.

Partial observability: hidden actions, noisy compliance, and audits. Action observability is strong: it assumes the parent directly sees whether a child complied. In many settings, only an outcome is observed (sales, latency, defect rates), which depends on unobserved effort and exogenous noise. Then a contract contingent on $A_t^w = B_t^w$ is infeasible, and our compliance indicator $\mathbf{1}\{A_t^w \ne B_t^w\}$ is not directly measurable.

One path is to move from *action-contingent* to *outcome-contingent* transfers, replacing B_t^w with a target statistic Y_t^w and paying based on a scoring rule. This enters the domain of moral hazard and requires assumptions on how actions map to observable outcomes. A second, operationally common path is *probabilistic auditing*: with small probability the principal observes a verifiable signal of action (a review, a log, a code diff), enabling a transfer scheme that is incentive compatible in expectation. Both variants preserve the recursive induced-utility viewpoint, but the transfer-learning problem becomes a partially observed control problem; the simple threshold estimation of Proposition 4 is replaced by either (i) identification of an outcome model, or (ii) joint optimization of audit rates and payments. In these regimes, numerical methods (e.g., Bayesian filtering for latent actions, POMDP solvers for audit policies, or simulation-based optimization) are typically required even in small trees.

Where numerical methods enter the picture. Across the extensions, a common dividing line is whether the parent's induced objective remains a sum of low-dimensional components with closed-form transfers. With continuous contexts, approximate separability, budgets, or partial observability, we often face a bilevel problem: a principal chooses (b, τ), anticipating a child's learned response under limited feedback. In practice we recommend a modular approach: keep the economic core (minimal-transfer logic, telescoping welfare identity, and regret decomposition) but allow the *node solver* to be numerical—for example, approximate dynamic programming for recommendation selection under interaction terms, convex programming for budgeted transfers, and likelihood-based estimation for compliance under noisy observation. The conceptual payoff is that even when computation becomes approximate, the decomposition into action, payment, and deviation channels remains a useful diagnostic: it tells us whether welfare losses arise from mislearning, mispricing, or noncompliance, and therefore which module should be improved before scaling to deeper hierarchies.

8 Experiments

Our theoretical results are deliberately modular—they separate (i) learning the induced action values, (ii) learning the inducing transfers, and (iii) controlling deviations so that errors do not amplify up the hierarchy. The experiments in this section are designed to stress-test exactly these three channels, and to make the comparative statics in $M = |\mathcal{C}|$, $p_{\min}$, and depth D visible in finite time.

Evaluation goals and logged quantities. In each run we log three families of outcomes. First, *welfare*: the realized per-round welfare $\sum_{v \in V} X_t^v$ and an empirical welfare regret computed against a context-by-context benchmark (estimated either by an oracle with access to the true θ_v, in synthetic experiments, or by an offline Monte Carlo estimator in the proxy environment). Second, *compliance*: deviation indicators $\mathbf{1}\{A_t^w \neq B_t^w\}$ for each edge, and their propagation to ancestors (since a single deviation can change the payoff distribution faced by multiple principals). Third, *payments*: the realized transfers $\tau_t(w)$, overpayment relative to the minimal inducing transfer $\tau_w^\star(c_t, B_t^w)$ when $\tau_w^\star$ is available (synthetic), and a conservative proxy when it is not (LLM toolchain). These logs allow us to empirically instantiate the decomposition in Proposition 3: when welfare deteriorates, we can ask whether the proximate cause is mislearning of actions, mispricing of transfers, or residual noncompliance.

Synthetic contextual hierarchies. We generate balanced B-ary trees of depth D with homogeneous action sets $|A_v| = K$. Contexts are drawn i.i.d. from a categorical distribution on $\mathcal{C}$ with controlled sparsity: we either use a near-uniform distribution (so $p_{\min} \approx 1/M$) or a skewed distribution in which a subset of contexts carries most mass while the remaining contexts appear rarely, exposing the failure mode suggested by the role of $p_{\min}$. Conditional on c, we draw separable mean rewards of the form

$$\theta_v(c, a_v, a_{C(v)}) \;=\; g_v(c, a_v) \;+\; \sum_{w \in C(v)} h_{v,w}(c, a_v, a_w),$$

with $g_v(c, a_v)$ and $h_{v,w}(c, a_v, a_w)$ sampled once at initialization and held fixed across time, and then clip and shift so that realized rewards lie in $[0, 1]$ after adding sub-Gaussian noise. This construction makes the true $\tau_w^{\star}(c, b)$ computable by backward induction, which is crucial for diagnosing transfer estimation rather than conflating it with action learning.

We compare Contextual-MAIL to three baselines that are economically meaningful. The first is *myopic learning without incentives*: each node runs a contextual bandit on its own realized rewards X_t^v and never pays transfers (so B_t^w is either absent or costless talk). This baseline isolates the value of incentive design when agents are strategic (or, more precisely, when local objectives are misaligned with upstream welfare). The second baseline is *static contracts*: each principal picks a fixed transfer schedule $\tau(w, b)$ independent of context (e.g., tuned on an initial batch), and then learns only which actions to recommend given those fixed prices. This captures the common operational practice of “set the bonus policy and move on.” The third baseline is a *centralized benchmark* that chooses the full action profile each round using knowledge of θ (an upper bound on attainable welfare rather than a feasible decentralized policy).

Across these synthetic instances we typically find two qualitative patterns. First, incentives matter primarily through compliance: myopic learning can converge to stable but inefficient local conventions, whereas Contextual-MAIL drives deviation rates down after a warmup, at which point the induced rewards behave like a standard contextual bandit with shifted payoffs. Second, the cost of decentralization appears mainly as payments: relative to the centralized benchmark, Contextual-MAIL attains comparable welfare only after paying the opportunity costs needed to align child behavior, and this payment burden is larger in contexts where the child’s action gaps are large.

Transfer estimation dynamics. A central diagnostic is whether $\widehat{\tau}_w(c, b)$ approaches $\tau_w^\star(c, b)$ from above (the conservative regime) quickly enough to stabilize behavior. In the synthetic setting we can plot, for each edge, the time path of

$$\max_{c \in \mathcal{C},\, b \in A_w} \big(\widehat{\tau}_w(c, b) - \tau_w^\star(c, b)\big) \quad \text{and} \quad \max_{c,b} \big(\tau_w^\star(c, b) - \widehat{\tau}_w(c, b)\big)_+,$$

alongside the realized deviation frequency. What matters operationally is not symmetric estimation error but *underpayment*: even a small negative error can trigger persistent deviations, which then corrupt the parent's reward observations and slow learning up the tree. Consistent with the theory's "overpay rather than under-incentivize" logic, conservative estimators typically deliver a sharp phase transition: once underpayment events become rare, deviation rates fall and the residual regret is dominated by standard exploration.

Ablations: payments versus compliance versus welfare. To connect outcomes to mechanism design choices, we run targeted ablations that selectively "turn off" components. (i) *Oracle transfers*: we give the algorithm access to $\tau^\star$ but still require it to learn which recommendations maximize induced utility. This isolates pure contextual action learning on shifted rewards and confirms that most early-round welfare loss in the full algorithm comes from transfer learning rather than action selection. (ii) *Optimistic transfer estimation*: we allow $\widehat{\tau}$ to be unbiased (or even slightly downward-biased) rather than conservative. This ablation is informative precisely because it tends to reduce payments but increase deviations; empirically, welfare often deteriorates because deviation-induced noise overwhelms the savings on transfers. (iii) *Frozen contracts after warmup*: we estimate $\widehat{\tau}$ for a fixed number of rounds and then stop updating transfers. This tends to stabilize compliance but produces persistent overpayment in contexts where the child's internal learner improves over time, illustrating that payment regret can remain nontrivial even when welfare regret is small. These ablations make concrete the practical tradeoff: platforms can buy compliance cheaply only if they accept a higher deviation rate and the downstream instability it causes.

Sensitivity to $|\mathcal{C}|$ and context frequency. We vary M holding T fixed to induce context sparsity. The core empirical regularity is unsurprising but important for deployment: when M grows, transfer estimation becomes the bottleneck, and the deviation rate remains elevated for longer because each (c, b) pair is visited less often. When contexts are skewed, performance becomes uneven: high-mass contexts quickly reach near-perfect compliance and low regret, while rare contexts remain effectively "unpriced," leading to sporadic but sharp welfare drops. This is the finite-sample manifestation of the role played by $p_{\min}$ in Proposition 4, and it motivates the continuous-context extensions in Section 7: without some form of generalization across contexts, rare states will remain fragile.

Sensitivity to depth and branching. We vary D and B to expose error propagation. Holding per-edge learning schedules fixed, deeper trees exhibit two effects: (a) longer time until the root experiences stable effective payoffs, because deviations in lower layers perturb the distribution of outcomes higher up; and (b) higher cumulative payments, because upstream principals must offer conservative transfers to insure against residual instability downstream. Increasing B has a different signature: even with separability, each principal must learn more (c, b) thresholds, and union-bound effects appear as a slower decline in the maximum underpayment probability across children. These observations align with the model's comparative statics and help calibrate exploration schedules in practice (e.g., allocating more warmup to deeper layers or to high-degree managers).

LLM toolchain proxy environment. Finally, we evaluate a proxy setting motivated by current platform practice: a principal (router/manager) assigns tasks to LLM-based agents (children) and can offer "credits" or "bonuses" that change the agent's private cost-benefit tradeoff of using expensive tools (retrieval, code execution, external APIs) versus cheap heuristics. Each round consists of a task with observable context c_t (e.g., domain, latency target, allowable external calls). Each child chooses an action a_w interpreted as a toolchain (from a small discrete set), which is observable from logs, satisfying action observability. Rewards combine task quality and resource costs, measured by automatic graders or held-out reference solutions plus penalties for latency and tool usage. Transfers are implemented as resource credits (e.g., additional tool-call budget, larger context window, or explicit monetary credits in a sandbox), and thus have a natural interpretation as opportunity-cost compensation.

In this proxy, Contextual-MAIL is not meant to claim behavioral realism about LLMs; rather, it serves as a systems-level check that (i) compliance can be instrumented, (ii) threshold-like responses can be induced by adjustable credits, and (iii) the same diagnostics—payment burn, deviation spikes, and welfare shortfalls—remain meaningful even when "agents" are tool-using models rather than humans. We find this environment particularly useful for ablations: by replaying the same task stream under different payment policies, we can directly visualize how conservative pricing trades off immediate credit expenditure against downstream stability in routing and quality.

Taken together, these experiments aim to make the model's economic logic operational: the tree structure creates a propagation problem, contexts create a data sparsity problem, and transfers are the control knob that converts local learning into global coordination.

9 Discussion: platform design guidance, limitations, and open problems

Our motivating use case is not an abstract hierarchy for its own sake, but a platform design problem: a top-level objective (quality, safety, cost, latency) must be implemented through layers of delegated decisions made by agents with their own local payoffs and learning dynamics. The model's contribution is to separate *what* a platform wants (a welfare benchmark defined context-by-context) from *how* it can get it under decentralization (recommendations plus transfers learned from interaction data), and to clarify where the engineering effort must go: stabilizing compliance so that upstream learning faces a stationary-enough problem.

Credits and bonuses as implementable transfers. In many platforms, transfers are not literal cash. They are *allocations of scarce resources* that agents privately value: tool-call budget, context-window budget, priority scheduling, API rate limits, or even reputational credits that affect future task assignment. Our equilibrium logic treats these as transferable utility units; what matters is that (i) the principal can condition them on an observable action (e.g., which toolchain was used), and (ii) the agent optimizes a local objective in which these credits enter additively. Under these conditions, the "minimal inducing transfer" $\tau_w^\star(c, b)$ can be read operationally as the *smallest bonus/credit* that makes action b the agent's best response in context c. The policy implication is that platforms can often avoid brittle instruction-following by instead designing *measurable incentives*: rather than exhorting an agent to "use retrieval on medical questions," one can price the retrieval tool in credits and adjust the credit subsidy by context.

Routing incentives and "pay-for-compliance" is a feature, not a bug. A recurring concern in deployments is that incentive schemes "waste budget" on payments that do not directly improve outcomes. Our framework highlights when such payments are in fact necessary: if a child's privately optimal action differs from the action that maximizes the parent's induced objective, the gap must be compensated somewhere. In other words, the platform is not buying performance directly; it is buying *alignment of best responses*. This distinction matters for diagnostics. If welfare is low because the induced objective is mislearned, more payments will not help. If welfare is low because agents are deviating (or mixing) due to under-incentivization, then additional payments can be the most cost-effective intervention because they restore the stationarity required for learning to work up the tree.

Practical guidance: conservative pricing, explicit budgets, and rare-context handling. Three design recommendations follow immediately from the theory and the deployment-style diagnostics we emphasized. First, platforms should bias transfer learning toward *conservative (weakly overpaying) estimates* of $\tau^\star$ during early phases. Underpaying even occasionally can induce deviations that corrupt upstream reward observations, creating a negative feedback loop; overpaying is costly but tends to preserve compliance and hence learnability. Second, bonus policies should be integrated with *budget constraints* explicitly. While our baseline analysis focuses on regret, product teams typically face a hard cap on credits per day or per user. This suggests a natural extension: solve a constrained optimization in which $\tau_t(w)$ is chosen to keep deviation probabilities below a target while satisfying a budget, possibly by prioritizing payments in high-impact contexts and tolerating higher deviation in low-impact regions. Third, rare contexts require dedicated handling. When $p_{\min}$ is small, uniform guarantees are unattainable without generalization; platforms should therefore either (i) coarsen the context taxonomy so that each bucket appears often enough to price, (ii) impose a safe default policy in rare contexts (e.g., require a high-compliance toolchain), or (iii) introduce cross-context structure (shared representations) so that transfer estimates can borrow strength.

Governance: auditability, fairness, and strategic externalities. One advantage of action-contingent credits is auditability: the platform can log recommended actions, realized actions, and transfers, and thus compute compliance rates and effective prices. This enables operational governance: if an incentive is causing unintended behavior (e.g., excessive tool use that harms latency), one can detect it as a shift in compliance or payments before it manifests as downstream quality failures. At the same time, incentives raise fairness and manipulation concerns. Context-dependent bonuses can differentially reward agents assigned to certain task types; if assignments correlate with protected characteristics or job roles, this may create disparate impact. Moreover, agents may “game” contexts if they can influence how tasks are classified. Our baseline assumes contexts are exogenous; in practice, platforms should treat context definition as part of mechanism design, with robust logging, anomaly detection, and possibly cryptographic attestation in high-stakes settings.

Limitations of the baseline model. Several assumptions are doing real work. The finite-context condition $|\mathcal{C}| = M$ and i.i.d. sampling are primarily analytical conveniences; real systems face nonstationarity (concept drift, changing user mix) and high-dimensional contexts. Action observability is also central: the parent must be able to verify whether the child took the recommended action to condition transfers. Many important choices are only partially observable (e.g., degree of effort, prompt quality, internal chain-of-thought), which pushes the problem toward moral hazard. Separability, likewise, is a tractability assumption: it rules out strong complementarities among siblings' actions. In multi-agent toolchains, such complementarities can be first-order (e.g., two agents coordinating on a shared intermediate artifact). Finally, we have modeled each node as a coherent learner optimizing a stable objective; human teams, heterogeneous models, and mixed incentives can violate this abstraction, making $\tau^\star$ itself drift over time.

Open problem 1: beyond trees (DAGs, coalitions, and overlapping principals). Modern organizations and platforms rarely form strict trees. Agents may report to multiple principals; components may be reused across workflows; incentives may interact through shared resource constraints. Formally, the principal–agent graph becomes a DAG (or even a cyclic graph with repeated interactions). The telescoping logic behind welfare alignment becomes more delicate: transfers that cancel along a tree edge may no longer cancel uniquely when nodes have multiple parents, and budget balance may require additional accounting variables (e.g., shadow prices for shared constraints). A promising direction is to characterize classes of graphs (series-parallel DAGs, arborescences with cross-links) where a generalized "potential function" still exists and decentralized learning remains stable.

Open problem 2: sequential tasks (MDPs, state, and long-run incentives). Many applications are not one-shot contextual rounds but multi-step processes with state: customer support threads, software development pipelines, or iterative planning-and-execution loops. In such settings, today's action changes tomorrow's context distribution, violating i.i.d. and introducing strategic delay or exploration incentives. Technically, the induced objective becomes an MDP and transfers must account for continuation values. One can imagine defining $\mu_v^\star$ as an *optimal value function* and $\tau^\star$ as compensating for Q-value gaps, but learning these quantities in a decentralized hierarchy raises new propagation issues: deviations early in an episode can change the entire downstream trajectory, amplifying the deviation-regret channel.

Open problem 3: moral hazard and unverifiable actions. When the key choice is effort or internal computation that cannot be verified, contracts cannot condition directly on actions. Platforms then resort to outcome-based bonuses, peer prediction, or audits. Outcome-based incentives reintroduce the classic tradeoff between risk-sharing and incentives, and in learning systems they interact with exploration: noisy outcome bonuses can inadvertently discourage experimentation. A useful target is a hybrid design in which observable proxies (tool usage, latency footprints) are combined with occasional audits, yielding a partially observable mechanism that approximates the action-contingent benchmark while remaining implementable.

Stepping back, our view is that the main value of this line of work is conceptual: it provides a *diagnostic lens* for when decentralization fails and which lever—learning, pricing, or compliance—is responsible. For platforms, the actionable message is not "always pay bonuses," but rather: if you want hierarchical learners to behave like a single planner, you must either align objectives intrinsically or pay the opportunity costs that alignment requires, and you must do so in a way that preserves the statistical conditions needed for learning to converge.