Power and Limitations of Aggregation in Compound AI Systems

Nivasini Ananthakrishnan[1] and Meena Jagadeesan[2]

[1]UC Berkeley
[2]Stanford University

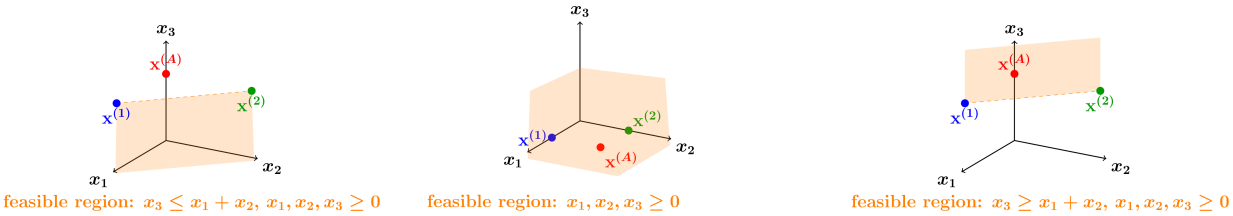
Abstract

When designing compound AI systems, a common approach is to query multiple copies of the same model and aggregate the responses to produce a synthesized output. Given the homogeneity of these models, this raises the question of whether aggregation unlocks access to a greater set of outputs than querying a single model. In this work, we investigate the power and limitations of aggregation within a stylized principal-agent framework. This framework models how the system designer can partially steer each agent's output through its reward function specification, but still faces limitations due to prompt engineering ability and model capabilities. Our analysis uncovers three natural mechanisms—*feasibility expansion*, *support expansion*, and *binding set contraction*—through which aggregation expands the set of outputs that are elicitable by the system designer. We prove that any aggregation operation must implement one of these mechanisms in order to be elicitability-expanding, and that strengthened versions of these mechanisms provide necessary and sufficient conditions that fully characterize elicitability-expansion. Finally, we provide an empirical illustration of our findings for LLMs deployed in a toy reference-generation task. Altogether, our results take a step towards characterizing when compound AI systems can overcome limitations in model capabilities and in prompt engineering.

1 Introduction

Compound AI systems—which leverage multiple AI components, rather than a single model in isolation—present a powerful paradigm to tackle complex tasks [BAIR Research Blog, 2024]. In the context of large language models (LLMs), one common approach is to create many copies of the same model, give these models different prompts or access to different tools, and aggregate the outputs of these models at test-time. This approach has proven fruitful in multi-agent research systems [Anthropic, 2025] where a lead LLM agent delegates subtasks to different specialized agents and aggregates their outputs, in multi-agent debate protocols where different LLM agents seek consensus [Du et al., 2024] or argue for different answers [Khan et al., 2024], and in prompt ensembling approaches where the outputs from different prompts are combined [Arora et al., 2022].

Given the empirical success of these compound LLM systems, this raises the question of when aggregating across multiple copies of the same model unlocks greater performance than querying a single model. At first glance, aggregation may seem redundant when the model copies are homogeneous. However, one source of improved performance is at the prompting level: complex prompt engineering approaches for a single model may be replaceable by simple but diverse prompting strategies for a set of models [Arora et al., 2022], illustrating how aggregation across models can overcome limitations in prompt engineering ability. Another source of improved performance is at the output level: aggregating multiple LLM agents over multiple interactions can help correct errors such as hallucinations [Du et al., 2024], illustrating how aggregation can overcome limitations in model capabilities as well. This suggests that the extent to which aggregation overcomes these limitations in prompt engineering and model capabilities fundamentally impacts the power of compound AI systems.



(a) Feasibility expansion (b) Support expansion (c) Binding set contraction

Figure 1: Three mechanisms by which the aggregation operation $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ expands the set of outputs that the system designer can elicit. Feasibility expansion captures when two feasible vectors are aggregated into an infeasible vector (left; Definition 3.1). Support expansion captures when two vectors are aggregated into a vector with richer support (middle; Definition 3.3). Binding set contraction captures when two vectors on the boundary of the feasible set are aggregated into a vector in the interior (right; Definition 3.5). Any aggregation operation must implement one of these mechanisms to offer power to the system designer (Theorem 3.7), and strengthened versions of these mechanisms characterize when aggregation adds power (Theorem 4.4, Theorem 4.3). See Figure 3 for an empirical illustration of these mechanisms for LLMs in a reference-generation task.

In this work, we study the power and limitations of aggregation from a theoretical perspective, building on a classical principal-agent framework [Kleinberg and Raghavan, 2020]. Our focus is on compound AI systems where a system designer passes reward functions (e.g., via prompts) to many copies of the same model and then aggregates their outputs. For intuition, consider a reference-generation task where the goal is to produce a list of paper references; the system designer may benefit from prompting different copies of an LLM to focus on different topics, and then taking the union or intersection of the resulting reference lists.

In our stylized principal-agent framework (Section 2), the system designer (i.e., the principal) designs reward functions to elicit M-dimensional outputs from each agent (i.e., each model), and aggregates these outputs to produce a synthesized output. Each agent generates an output in its feasible set that maximizes the reward function. The system designer co-designs the reward functions across models to try to produce a specific output. We capture prompt engineering limitations as the reward functions operating over a coarser N-dimensional feature space rather than directly over M-dimensional outputs, and model capability limitations as conic constraints on each agent's feasible set of outputs.

Using our theoretical framework, we formalize three natural mechanisms by which aggregating across multiple agents enables the system designer to elicit a greater set of outputs than relying on a single model (Figure 1). The first mechanism is *feasibility expansion*, where aggregation produces outputs outside of any agent's feasibility set. The second is *support expansion*, where aggregation combines outputs with smaller supports into an output with a richer support. The third is *binding set contraction*, where aggregation combines outputs that are binding with respect to constraints into an output that falls within the interior.

We then formally connect these mechanisms to elicitability-expansion. Specifically, we find that the power of aggregation fundamentally relies on at least one of these mechanisms being implemented: if none are implemented, then aggregation does not expand elicitability on any problem instance (Theorem 3.7). This illustrates a strong form of limitation where the aggregation operation offers no power to the system designer regardless of the level of prompt engineering limitations. However, we also show that these mechanisms do not provide sufficient conditions for elicitability-expansion. To close this gap, we prove that strengthened versions of the mechanisms fully characterize elicitability-expansion (Theorem 4.3 and Theorem 4.4).

Finally, to connect our findings to LLM aggregation, we empirically illustrate the three mechanisms for LLMs (GPT-5.4, GPT-5-mini, GPT-4o-mini) deployed in a toy reference-generation task (Figure 3; Table 1). The system designer aims to generate a list that reflects a specific balance of topics, but can only prompt the model using higher-level topics. Within this task, we empirically construct aggregation operations that implement each mechanism (feasibility expansion, support expansion, and binding-set contraction), and that

also expand the set of outputs that are elicitable to the system designer. This empirical analysis illustrates the robustness of our findings beyond the assumptions of our theoretical framework.

Altogether, our work uncovers key mechanisms that underpin the power and limitations of aggregation in compound AI systems. Our results fully characterize when aggregation expands the set of elicitable outputs. Our results offer conceptual insights about aggregation for system designers (Section 4.5), and our mechanisms connect to empirical phenomena observed for language models (Section 6). More broadly, our work takes a step towards understanding when aggregation of multiple copies of the same model provides benefits to system designers.

1.1 Related Work

Aggregation across multiple models. Aggregating outputs from multiple copies of an LLM is a common strategy for tackling complex tasks. One common approach is resampling the same model or reasoning trace and then selecting outputs via reward models [Christiano et al., 2017], self-consistency [Wang et al., 2023b], or synthesis [Zhang et al., 2025]; coverage is an important property for inference-time computations [Huang et al., 2025]. Closest to our setting are systems with multiple copies of the same model under different reward function specifications, as in LLM debate [Du et al., 2024], consensus games between generators and discriminators [Jacob et al., 2023], prompt ensembling [Arora et al., 2022], and multi-agent research frameworks [Anthropic, 2025]. Another related approach is to combine many different LLMs, for example by routing queries across different LLMs [Chen et al., 2023] or adversarially combining models to generate unsafe outcomes [Jones et al., 2025a]. Motivated by these systems, we study the fundamental conceptual question of when aggregating multiple models elicits strictly more outputs than querying a single model.

The conceptual question that we study is inspired by a rich literature on aggregation in a variety of other domains. This includes ensembling [Dietterich, 2000], voting [Ladha, 1992], distributed algorithms [Lynch, 1996], and multi-agent reinforcement learning [Tan, 1993], among other domains. For example, some works (e.g., [Gentzkow and Kamenica, 2017, Collina et al., 2025, Bansal et al., 2021, Donahue et al., 2022]) demonstrate the benefits of aggregating heterogeneous agents in achieving alignment. However, these works focus on a prespecified set of distinct agents, whereas we focus on eliciting different behaviors from the same agent through designing different rewards. Turning to market-level aggregation coming from users choosing between different models, another line of work studies when model-providers are incentivized to train heterogeneous models [Ben-Porat and Tennenholtz, 2019, Jagadeesan et al., 2023, Raghavan, 2024, Xu et al., 2025, Einav and Rosenfeld, 2025, Donahue and Raghavan, 2026]. As another example, a growing line of work on alignment under plurality uses principles from voting rules and social choice to design methods for aggregating diverse human preferences and training models to optimize this aggregated objective (e.g., [Conitzer et al., 2024, Mishra, 2023, Dai and Fleisig, 2024, Chidambaram et al., 2025, Shirali et al., 2025, Gölz et al., 2025]).

Principal-Agent Models and Reward Design. Our model extends the principal-agent framework introduced by Kleinberg and Raghavan [2020]. This model was originally developed in a strategic classification setting where an agent takes strategic efforts to improve their classification outcome. Within this framework, they characterize the type of effort that classifiers induce in agents, identifying when an agent is incentivized to exert efforts that genuinely improve their outcomes rather than merely gaming the classifier. We extend their model in two ways to study how aggregating outputs from models can overcome prompting and output limitations. First, the principal interacts with multiple agents at once and aggregates the outputs produced by different agents. Second, agents face conic constraints on their outputs, which captures output limitations. Our goal also differs: we introduce and characterize a novel notion of elicitability-expansion to capture the power afforded by aggregation for some level of prompting limitation.[1] We note that Alon et al. [2020]

[1]The main technical lemma (Lemma 4.7) underlying our characterization of elicitability-expansion extends Theorem 3 of their paper—which characterizes the elicitability of a given effort profile—to the setting with additional conic constraints on agent outputs. Their characterization reveals a single-agent limitation: effort profiles with larger supports are difficult to elicit (Lemma 10 of Kleinberg and Raghavan [2020]). This limitation underlies support expansion, one of the three mechanisms through which we show aggregation overcomes single-agent limitations.

also generalizes Kleinberg and Raghavan [2020] to multiple agents, but their model and motivation differ significantly from ours. In their model, different agents have different mappings from features to outputs but share the same reward function specification, and they aim to design a single reward function specification that incentivizes agents to all choose an output vector with a specific structure.

These models fall under the broader principal-agent framework [Holmström, 1979, Grossman and Hart, 1983, Laffont and Martimort, 2002, Campbell et al., 2007], which captures the challenge of designing rewards based on imperfect proxies. Zhuang and Hadfield-Menell [2020] use this framework to study misalignment between the AI agent's reward and the human's reward, focusing on the impact of underspecification. In addition to the effects of imperfect reward functions arising from limitations in reward design, principal-agent frameworks also capture agent limitations, often in the form of costs for actions. Similar to the interdependence of output dimensions induced by our conic constraints, multi-task principal-agent settings [Holmstrom and Milgrom, 1991, Slade, 1996, Bond and Gomes, 2009] study the effects of cost dependencies between tasks, which lead to phenomena such as substitutability and complementarity. Substitutability occurs when effort on one task increases the marginal cost of effort on another, while complementarity occurs when effort on one task decreases it. These dependencies among tasks are similar to the interactions among output dimensions captured by conic constraints in our model. Principal–agent theory has also considered multiple agents [Holmstrom, 1982, Lazear and Rosen, 1981, Dasaratha et al., 2024], focusing mainly on the joint design of rewards, but our work differs in considering the use of aggregation to synthesize new outputs. Finally, in a related but complementary literature on strategic classification, a handful of papers [Hossain et al., 2024, Liu et al., 2022] study strategic interactions between multiple agents.

2 Model

We extend the principal-agent framework in Kleinberg and Raghavan [2020] to capture a compound AI system with K agents (who represent LLMs) and a single principal (the system designer) who aggregates the outputs of the agents. The outputs are represented in M-dimensions, and we capture model capability limitations as conic constraints $\boldsymbol{C}$ over the output space (Section 2.1). The system designer designs reward function specifications and budget levels in order to elicit outputs from each agent (Section 2.2). The reward function operates over a coarser feature space defined by $\boldsymbol{\alpha}$, which captures prompt engineering limitations. Within this framework, we formalize when aggregation is elicitability-expanding (i.e., when it expands the set of outputs that the system designer can elicit), which captures the power of aggregation (Section 2.3). As a case study, we instantiate our framework in a reference-generation task (Section 2.4) which we will revisit in our empirical analysis in Section 5. We defer a discussion of the limitations of our framework to Section 6.

2.1 Output space

We embed outputs of agents into M-dimensional vectors with non-negative coordinates. We view each output dimension as capturing a different characteristic of the output. The vector representation $\boldsymbol{x}$ quantifies the degree to which the output captures each characteristic. We note that some dimensions may capture undesirable characteristics (e.g., hallucinations). The system designer seeks a specific output $\boldsymbol{x}^{(A)} \in \mathbb{R}^M_{\geq 0}$.

Model capability limitations as conic constraints. Our framework can capture restrictions on the outputs agents produce, for example due to capability limitations. We study restrictions requiring output vectors to satisfy conic constraints. These conic constraints capture the types of outputs that the agent can produce: for example, some agents may not be able to avoid producing hallucinations without facing capability degradation along other characteristics.

We let L denote the number of conic constraints, and we let $\boldsymbol{C} \in \mathbb{R}^{L \times M}$ denote the conic constraints themselves. Let $\boldsymbol{C}_i \in \mathbb{R}^M$ denote the ith row of $\boldsymbol{C}$ for $i \in [L]$, and let $\boldsymbol{C}_V \in \mathbb{R}^{|V| \times M}$ denote the set of rows corresponding to indices $V \subseteq [L]$. We denote by $\boldsymbol{C}_\emptyset$ the zero-vector, to capture how $\{\boldsymbol{d} : C_\emptyset \boldsymbol{d} \leq 0, \boldsymbol{d} \geq 0\} =$

$\mathbb{R}^M_{\geq 0}$. This restriction defines a feasible set $\mathcal{B}^{\text{feasible}}(\boldsymbol{C})$ of output vectors:

$$\mathcal{B}^{\text{feasible}}(\boldsymbol{C}) := \left\{\boldsymbol{x} \in \mathbb{R}^M_{\geq 0} \mid \boldsymbol{C}\boldsymbol{x} \leq \boldsymbol{0}\right\}.$$

We assume that membership in $\mathcal{B}^{\text{feasible}}(\boldsymbol{C})$ does not implicitly require any output dimension to always be zero. This assumption on $\boldsymbol{C}$ is stated below.

Assumption 2.1. We assume that given any output dimension $i \in [M]$, there exists an output vector $\boldsymbol{x} \in \mathbb{R}^M_{\geq 0}$ satisfying $x_i > 0$ and $\boldsymbol{C}\boldsymbol{x} \leq 0$.

2.2 Reward and Budget Specification

The system designer designs a reward function specification $R^{(k)}$ and a budget level $E^{(k)}$ for each agent $k \in [K]$. The reward function specification represents the reward function implicit in the prompt given to the agent, and the budget level represents the level of test-time compute that the agent is allowed to use.

Prompt engineering limitations as coarser features. To capture prompt engineering limitations, we model the reward function specification as operating over a coarser N-dimensional feature space than the outputs. We adopt the same form of how output vectors map to features as in Kleinberg and Raghavan [2020]. Here, the features $\boldsymbol{F}(\boldsymbol{x}) = [F_1(\boldsymbol{x}), \ldots, F_N(\boldsymbol{x})]$ take the form

$$F_j(\boldsymbol{x}) = f_j\left(\sum_{i=1}^{M} \alpha_{ji}\boldsymbol{x}_i\right).$$

We call the α_{ji}'s *feature weights*. We will denote by $\boldsymbol{\alpha} \in \mathbb{R}^{N\times M}_{\geq 0}$ the matrix with entries α_{ji} and call this the *feature weights matrix*. We borrow the following assumptions on the feature mapping functions F_j, $j \in [N]$ from Kleinberg and Raghavan [2020].

Assumption 2.2 (Assumptions on feature mapping)**.** We assume that each $f_j(\cdot)$ for $j \in [N]$ is strictly increasing, nonnegative, smooth, and weakly concave (i.e., diminishing returns from increasing quality on this dimension). We also assume each $\alpha_{ji} \geq 0$, for $i \in [M], j \in [N]$. Finally, we assume that $\boldsymbol{\alpha}$ has no zero rows, which rules out trivial features.

Reward functions. We consider reward functions $R^{(1)}, \ldots, R^{(K)} : \mathbb{R}^N \to \mathbb{R}$ which operate on the features $(F_j)_{j=1}^N$. Following prior work [Kleinberg and Raghavan, 2020], we restrict to *monotone* reward functions R satisfying the notion of monotonicity stated below.

Assumption 2.3 (Monotonicity of reward functions)**.** We assume reward functions R are monotone. That is, R does not decrease if all features are weakly increased i.e., if $F_j(\boldsymbol{x}') \geq F_j(\boldsymbol{x})$ for every $j \in [N]$, then $R(x') \geq R(x)$. Additionally, there exists a feature F_j such that increasing the value of F_j, keeping all other features fixed, strictly increases the value of R.

Agent optimization program. Given a monotone reward function $R^{(k)}$ and a positive budget level $E^{(k)} > 0$, each agent k produces an output that maximizes its reward $R^{(k)}$, meets its budget constraint, and is within the feasible set $\mathcal{B}^{\text{feasible}}(\boldsymbol{C})$: that is,

$$\boldsymbol{x} \in \mathbf{X}^*(R^{(k)}, E^{(k)}; \boldsymbol{C}, \boldsymbol{F}) := \text{argmax}_{\boldsymbol{x}\in\mathcal{B}^{\text{feasible}}(\boldsymbol{C}), \|\boldsymbol{x}\|_1 \leq E^{(k)}} R^{(k)}(\boldsymbol{F}(\boldsymbol{x})).$$

This captures how even though agents are homogeneous and solve the same optimization program, they can be given different reward functions and thus produce different outputs.

2.3 Elicitability

Given constraints $\boldsymbol{C}$ and features $\boldsymbol{F}$, we say that an output $\boldsymbol{x}$ is elicitable by a single agent if there exist a monotone reward function R and a positive budget level E such that $\boldsymbol{x} \in \mathbf{X}^*(R, E; \boldsymbol{C}, \boldsymbol{F})$. As shown in prior work [Kleinberg and Raghavan, 2020] and illustrated in Section 3.1, some output vectors $\boldsymbol{x} \in \mathcal{B}^{\text{feasible}}(\boldsymbol{C})$ are not elicitable by any reward function R and budget level E. We note that the reward function R determines whether the direction $\boldsymbol{x}/\|\boldsymbol{x}\|_1$ is elicitable, and the specified budget E determines the ℓ_1 norm $\|\boldsymbol{x}\|_1$ of elicitable outputs. Specifically, increasing the budget scales up the ℓ_1 norm of the elicitable outputs, but it does not change the set of elicitable directions (Corollary B.5).

As we will show later in our analysis, the condition for whether $\boldsymbol{x}$ is elicitable only depends on $\boldsymbol{x}$ through the following sufficient statistic $(\mathcal{S}(\boldsymbol{x}), \mathcal{V}(\boldsymbol{x}))$. The first component $\mathcal{S}(\boldsymbol{x}) = \{i \in [M] : x_i > 0\}$ denotes the support of x. The second component $\mathcal{V}(\boldsymbol{x}) = \{l \in [L] : C_l \boldsymbol{x} = 0\}$ denotes the set of indices of conic constraints that are binding at $\boldsymbol{x}$.

Elicitability-expansion. When the system designer can aggregate the outputs of different agents, this may expand the set of elicitable outputs. That is, outputs that are not directly elicitable by specifying reward functions and budgets may be obtained by aggregating a set of outputs that are directly elicitable. The following definition captures when aggregation expands the set of elicitable outputs.

Definition 2.4 (Elicitability-expansion)**.** We call $\boldsymbol{x}^{(1)} \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ an **elicitability-expanding operation** *relative to constraints* $\boldsymbol{C}$ *and features* $\boldsymbol{F}$ if

- There exist monotone reward functions $R^{(1)}, \ldots, R^{(K)}$ and positive budget levels $E^{(1)}, \ldots, E^{(K)}$ such that $\boldsymbol{x}^{(k)} \in X^*(R^{(k)}, E^{(k)}; \boldsymbol{C}, \boldsymbol{F})$ for all $k \in [K]$.
- There does not exist a monotone reward function R and budget level $E > 0$ such that $x^{(A)} \in X^*(R, E; \boldsymbol{C}, \boldsymbol{F})$.

We say that $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is an elicitability-expanding operation relative to constraints $\boldsymbol{C}$ if there *exist* features $\boldsymbol{F}$ such that $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding relative to $\boldsymbol{C}$ and $\boldsymbol{F}$.

We only consider aggregation operations $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$, for $k \in [K]$ is feasible. Other operations are clearly not useful to the system designer. Intuitively, if an aggregation operation is elicitability-expanding, then there is a prompt engineering limitation under which this operation offers power. Namely, this operation allows the system designer to query the model using multiple distinct reward and budget specifications and put together the resulting output vectors to obtain a new output that is not directly elicitable by specifying a single reward and budget to the model. When an aggregation operation is not elicitability-expanding it is not useful for *any* form of prompt engineering limitation. Our results (Lemma 4.7) show that the elicitable set under features $\boldsymbol{F}$ depends on $\boldsymbol{F}$ only through the feature weights matrix $\boldsymbol{\alpha}$. So we will denote the elicitability set and conditions for elicitability-expansion in terms of $\boldsymbol{\alpha}$ instead of features $\boldsymbol{F}$.

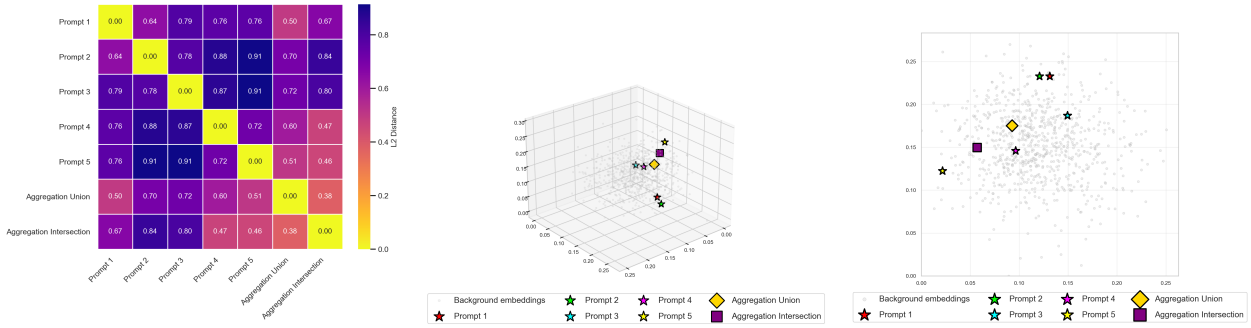
Aggregation rules. An aggregation rule is a mapping from a list of output vectors $(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)})$ to an aggregated output vector $\boldsymbol{x}^{(A)}$. There are two natural aggregation rules we will often consider in our work. Although our results apply to more general aggregation rules, we will often use these natural aggregation rules to provide examples.

The first is *intersection aggregation*, which is defined to be the coordinate-wise minimum of the vectors:

$$\mathcal{A}_{\text{intersect}}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}) = \boldsymbol{x}^{(1)} \wedge \ldots \wedge \boldsymbol{x}^{(K)}. \tag{1}$$

This aggregation rule combines outputs based on commonality among different output vectors, which is conceptually similar to debate protocols [Du et al., 2024] that aim to create agreement or inference scaling methods that aim to filter out incorrect information [Zhang et al., 2025]. The second is *addition aggregation*, which takes a weighted sum of the vectors. For a weight vector $\boldsymbol{w} \in \mathbb{R}^K_{\geq 0}$, the rule is given by

$$\mathcal{A}_{\text{add}}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}; \boldsymbol{w}) = \sum_{k=1}^{K} \boldsymbol{w}_k \boldsymbol{x}^{(k)}. \tag{2}$$



(a) ℓ_2 distances (b) Projection onto top 3 dims (c) Projection onto top 2 dims

Figure 2: Visualization of output vectors for a reference-generation task (Section 2.4). Output vectors are computed using the $M = 768$-dimensional embeddings from all-mpnet-base-v2, shifted to be in the nonnegative orthant. Embeddings are shown for GPT-4o-mini outputs from five different prompts, and as well as two different aggregated outputs based on additional-style and intersection-style aggregation rules. The ℓ_2-distances (left) and projections onto the top 3 highest-variance (middle) and top 2 highest-variance dimensions (right), are shown. The plots show that the five prompts produce semantically different outputs, and each aggregation operation results in a combination of the five outputs that does not resemble any output in isolation.

Addition aggregation interpolates among different output directions. This rule conceptually captures how system designers synthesize multiple outputs to delegate specialized subtasks to each agent and synthesize the outputs of these subtasks [BAIR Research Blog, 2024, Anthropic, 2025].

2.4 Illustrative Example: Reference-Generation Task

To illustrate how our framework captures LLM aggregation, we consider a natural task: generating a list of papers on a given topic [Wang et al., 2023a, Press et al., 2024]. We will also revisit this task in our empirical analysis in Section 5. At a high-level, the system designer may benefit from asking many copies of the same LLM to produce a reference list, and then aggregating these lists into a final list. For example, to increase the breadth of the final list, the system designer may prompt each LLM to focus on topics from a specific subarea and then take a union of the lists. Alternatively, to reduce hallucinations or remove irrelevant references, the system designer may prompt each LLM to focus on different measures of paper quality, and then take an intersection of the lists.

Below, we show how different instantiations of our framework capture different aspects of this task.

Output vector space. We specify two different instantiations of the output vector space $\mathbb{R}^M_{\geq 0}$ in our framework, depending on the level of specificity of the output which the system designer aims to elicit.

1. Suppose that the system designer aims to elicit a list of papers that overall reflects a specific balance of different topics (e.g., machine learning, economics, etc.). To capture this, let each dimension $i \in [M]$ of the output capture a different topic, so the value x_i captures the fraction of papers on the list that reflect the corresponding topic. We build on this instantiation in our empirical analysis in Section 5.

2. Suppose that the system designer aims to elicit a specific list of references. To capture this, we take the output vector to be high-dimensional embedding coming from a text embedding model. We design a simple experiment using LLMs to illustrate this: we prompt GPT-4o-mini in different ways, perform aggregation also using GPT-4o-mini, and use a sentence-transformers model to produce 768-dimensional output vectors (see Section C for details of the empirical setup).[2] Figure 2 shows the embeddings for

[2]Code is available at https://github.com/nivasini/aggregation_compound_ai.

outputs to the five prompts, as well as the outputs produced by the intersection-style and addition-style aggregation rules. The five prompt outputs vary substantially, and the aggregated outputs differ markedly from both the originals and from each other, demonstrating how different prompts and aggregation rules can shape the embedding-space representation.

Reward function specification limitations. In our framework, recall that the reward function operates over coarsenings of the output dimensions (as captured by the feature weights matrix $\boldsymbol{\alpha}$) rather than directly on the output dimensions. This captures two types of prompt engineering limitations that we describe below.

1. First, the system designer may struggle to precisely express what they truly want in the prompt, leading to underspecified prompts omitting some of the system designer's requirements [Yang et al., 2025]. For example, the system designer may prompt the model to focus on combinations of high-level topics, rather than fully specifying the fine-grained topic that they wish to see. We build on this instantiation in our empirical analysis in Section 5.

2. Second, the model may not correctly interpret the system designer's prompt. For example, the model might map two dissimilar words in the prompt to the same word [Jones et al., 2025b]. In the citation task, this could surface as the model interpreting "papers with high attribute 'X"' similarly for many different attributes "X".

3 Natural Mechanisms for Elicitability-Expansion

In this section, we formalize natural mechanisms by which aggregation expands elicitability (Figure 1). First, we define the mechanisms, and we show how they expand elicitability via examples (Section 3.1). Then, we show that these mechanisms are necessary for elicitability-expansion in general (Section 3.2). While these mechanisms are not sufficient for elicitability-expansion in general, strengthened versions of these mechanisms turn out to fully characterize elicitability-expansion, as we will show in Section 4.

3.1 Formalizing the Mechanisms and Motivating Examples

We formalize three natural mechanisms and demonstrate their ability to expand elicitability through examples. While our mechanisms are general, our examples focus on a 3-dimensional output space ($M = 3$) with 2-dimensional features ($N = 2$). We focus on feature weights matrices $\boldsymbol{\alpha}$ of the form $\boldsymbol{\alpha}(q) := \begin{bmatrix} 1 & 0 & q \\ 0 & 1 & q \end{bmatrix}$. Each of the output dimensions x_1, x_2 specialize to features F_1, F_2, respectively. That is, increasing the first output dimension x_1 only increases the first feature F_1, and increasing the second output dimension x_2 only increases the second feature F_2. Increasing the third output dimension x_3 increases both features, though the contribution is weighted by a factor of q. The parameter q captures the extent to which it is possible to simultaneously maximize both features.

Our examples use the intersection and addition aggregation rules that we previously introduced.

Let us now formalize the mechanisms and demonstrate their ability to expand elicitability.

Mechanism 1: Feasibility Expansion. Aggregation can help overcome the output limitations due to the conic constraints faced by each agent, producing outputs that are outside of the feasible set $\mathcal{B}^{\text{feasible}}(\boldsymbol{C})$. We formalize this through the following mechanism.

Definition 3.1 (Feasibility Expansion)**.** Given a constraint matrix $\boldsymbol{C}$, an aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements **feasibility expansion** if $\boldsymbol{x}^{(A)}$ is infeasible i.e. $\boldsymbol{C}\boldsymbol{x}^{(A)} \not\leq 0$.

The following example, depicted in Figure 1a, illustrates how aggregation operations which implement feasibility expansion can in turn expand elicitability.

Example 3.2. *Let the feature map be $\boldsymbol{\alpha} = \boldsymbol{\alpha}(2)$, so that increasing the third output dimension contributes significantly to both features. We view the first two output dimensions as corresponding to two types of "undesirable" behavior, while dimension 3 corresponds to "desirable" behavior. Let $\boldsymbol{C}$ be a single constraint of the form $x_3 \leq x_1 + x_2$. The constraint captures how the model cannot produce the desirable dimension without also producing some of the undesirable dimension(s).*

Consider the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 1], \boldsymbol{x}^{(2)} = [0, 1, 1] \to \mathcal{A}_{intersect}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) = [0, 0, 1]$, which implements feasibility expansion. The output $[0, 0, 1]$ is outside the feasibility set since it has only desirable dimensions and hence is not elicitable with any reward function specification and budget level. The outputs $[1, 0, 1]$ and $[0, 1, 1]$ turn out to be elicitable, so the aggregation operation is elicitability-expanding (Proposition A.4 in Appendix A.2.1).

Mechanisms 2-3: Overcoming Reward Function Specification Limitations. Even when an output is in the feasible set, the limitations of reward function specification still restrict which outputs are elicitable. Aggregation can overcome the reward function specification limitations faced by the system designer, as the next two mechanisms formalize.

Mechanism 2: Support Expansion. Reward function specification limitations can make it impossible to elicit outputs with a large support, as our results will later show.[3] The following mechanism formalizes how aggregation can combine outputs with smaller supports to produce an output with a richer support.

Definition 3.3 (Support expansion)**.** An aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements **support expansion relative to** k if $\mathcal{S}(\boldsymbol{x}^{(A)}) \not\subseteq \mathcal{S}(\boldsymbol{x}^{(k)})$.

Aggregation operations which implement support expansion can in turn expand elicitability, by producing outputs with richer supports than that are elicitable by a single agent. We illustrate this in the following example, which is depicted in Figure 1b.

Example 3.4. *Let the feature map be $\boldsymbol{\alpha} = \boldsymbol{\alpha}(0.6)$. Suppose that there are no conic constraints $\boldsymbol{C} = \emptyset$, so elicitability challenges entirely stem from reward function specification limitations. We will think of the first two dimensions as two aspects we would like our output to simultaneously capture.*

Consider the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 0], \boldsymbol{x}^{(2)} = [0, 1, 0] \to \mathcal{A}_{add}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}; [1/2, 1/2]) = [1/2, 1/2, 0]$ which implements support expansion. The output $[1/2, 1/2, 0]$ cannot be elicited directly, because reward functions that equally reward both features make dimension 3 strictly preferred over the combination of dimensions 1 and 2. In contrast, outputs supported on just one of these two dimensions $j \in \{1, 2\}$ can be elicited by only rewarding feature F_j. As a result, the aggregation operation is elicitability-expanding (Prop A.5 in Appendix A.2.2).

Mechanism 3: Binding Set Contraction. This next mechanism overcomes reward function specification limitations by taking advantage of the output limitations of the agent. Perhaps counterintuitively, the constraints on the output space can make it easier to elicit an output through a single reward. When a constraint is binding for an output vector, some reward-increasing directions become inaccessible to the agent, as these directions will lead to violation of the binding constraint. There are thus fewer directions of change that need to be disincentivized. Aggregation can combine outputs with binding constraints into an output with fewer binding constraints.

Definition 3.5 (Binding set contraction)**.** An aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements **binding set contraction relative to** k if $\mathcal{V}(\boldsymbol{x}^{(A)}) \not\supseteq \mathcal{V}(\boldsymbol{x}^{(k)})$.

Aggregation operations which implement binding set contraction can expand elicitability. We illustrate this in the following example, which is depicted in Figure 1c.

[3] Kleinberg and Raghavan [2020] showed this in single-agent environments without constraints.

Example 3.6. *Let the feature map be $\boldsymbol{\alpha} = \boldsymbol{\alpha}(0.2)$. As in the first example, we will think of x_3 to be a "desirable" dimension and x_1, x_2 to be "undesirable" dimensions. Let $\mathbf{C}$ be a single constraint of the form $x_1 + x_2 \leq x_3$. This constraint captures how the model cannot produce the bad dimension(s) without also producing some of the good dimension.*

Consider the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 1], \boldsymbol{x}^{(2)} = [0, 1, 1] \to \mathcal{A}_{intersect}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) = [0, 0, 1]$, which implements binding-set contraction. The value of $q = 0.2$ is small, leading to dimension 3 being inelicitable without the constraint. The constraint allows us to elicit a vector with some amount of x_3, but not a vector that has only x_3. The aggregation operation can allow us to get dimension x_3 without the additional dimensions x_1 or x_2, and is thus elicitability-expanding (Prop A.6 in Appendix A.2.3).

3.2 Connections between Elicitability-Expansion and Mechanisms

Moving beyond the examples in Section 3.1, we more generally study how these mechanisms connect to elicitability-expansion. We will show that implementing one of these mechanisms is necessary for elicitability-expansion but is not sufficient.

Necessity of these mechanisms. First, we show that if an aggregation operation expands elicitability, it must implement at least one of the three mechanisms. Specifically, Theorem 3.7 shows that either the operation must implement feasibility-expansion or it must implement at least one of support expansion or binding set contraction for every output $x^{(k)}$.

Theorem 3.7. *Fix conic constraints $\boldsymbol{C}$, and any aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$ is feasible (i.e., $\boldsymbol{C}\boldsymbol{x}^{(k)} \leq 0$, for every $k \in [K]$). If $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding relative to constraints $\boldsymbol{C}$, then at least one of the following conditions holds:*

- *$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is feasibility-expanding relative to $\boldsymbol{C}$ (Definition 3.1).*
- *For each $k \in [K]$, $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is either support-expanding relative to k (Definition 3.3) or binding set-contracting relative to k (Definition 3.5).*

Theorem 3.7 reveals a strong form of limitation for aggregation operations that do not implement at least one of the mechanisms (Definition 3.1, 3.3, and 3.5). Specifically, the result illustrates that if an operation does not implement the mechanisms according to the conditions in Theorem 3.7, then aggregation is not elicitability-expanding regardless of the feature weights matrix $\boldsymbol{\alpha}$. This result illustrates conditions under which aggregation adds no power to compound AI systems regardless of the level of prompt engineering limitations.

Proof ideas. The full proof of Theorem 3.7 is presented in Section A.3.1. The proof builds on the technical tools we develop in Section 4 (i.e., Theorem 4.4). The main idea is that when the condition of Theorem 3.7 is violated by not implementing any of the natural mechanisms, the aggregation operation does not expand the set of viable directions of change. The notion of viable directions is formalized in Definition 4.1. At a high level, these are directions in which we can move a small amount from an output vector while retaining non-negative values on all coordinates and continuing to satisfy the conic feasibility constraints and budget constraints. A vector with a larger set of viable directions is harder to elicit since this vector must be preferable compared to a larger set of feasible vectors. An aggregation operation that expands the set of viable directions produces vectors that are hard to elicit through direct prompting.

These mechanisms are not sufficient. While implementing one of these three natural mechanisms (implemented in the way provided by conditions of Theorem 3.7) is necessary for aggregation to have power, it is not necessarily sufficient to guarantee power. Even if an aggregation operation implements support expansion for every $k \in [K]$, the aggregation operation still may not be elicitability-expanding as shown in Proposition A.1 in Section A.1. This is a strong limitation since it means that the aggregation operation offers no power regardless of the feature weights matrix. Proposition A.2 in Section A.1 shows a similar

insufficiency for binding set contraction. On the other hand, feasibility expansion on its own guarantees elicitability-expansion as shown in Proposition A.3 in Section A.1.

Summary. While the conditions based on the three natural mechanisms stated in Theorem 3.7 are necessary for aggregation to have power, they are not sufficient for aggregation to have power. In the next section (Section 4), we will show that implementation of a strengthened version of these mechanisms serves as a necessary and sufficient conditions for power of aggregation, more precisely capturing the power and limitations of aggregation.

4 Characterizing Elicitability-Expansion

In this section, we will provide a characterizing condition that is both necessary and sufficient for an aggregation operation to have power (i.e., expand elicitability under some feature map). This characterizing condition is based on *strengthened* versions of the natural mechanisms we introduced in Section 3.1.

4.1 Necessary and Sufficient conditions for elicitability-expansion

Before stating our characterizing condition we will introduce an object that our characterizing condition depends on. This is the set of feasible and budget-reducing directions of the output vectors $\boldsymbol{x}$ in the aggregation operation. We first define this set and how it relates to the elicitability of output vectors.

Definition 4.1 (Feasible, budget-reducing directions)**.** The set of feasible, budget-reducing directions for an output vector $\boldsymbol{x}$ depends on the sufficient statistics of the support and binding conic constraints indices $(\mathcal{S}(\boldsymbol{x}), \mathcal{V}(\boldsymbol{x}))$ of $\boldsymbol{x}$. It is defined as

$$\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} = \underbrace{\{\boldsymbol{d} \in \mathbb{R}^M : \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x})}\boldsymbol{d} \leq 0\}}_{(1)} \cap \underbrace{\{\boldsymbol{d} \in \mathbb{R}^M : d_j \geq 0 \forall j \in (\mathcal{S}(\boldsymbol{x}))^c\}}_{(2)} \cap \underbrace{\{\mathbb{1}^t\boldsymbol{d} < 0\}}_{(3)}.$$

The set $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$ captures the set of viable directions to move a small amount from $\boldsymbol{x}$ while maintaining all constraints on output vectors (i.e., non-negative coordinates, feasibility according to conic constraints, and satisfying budget constraints). Note that the directions in $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$ are defined relative to only binding conic and non-negativity constraints rather than the set of all constraints. This is because we can move a small amount in a direction violating non-binding constraints while not violating those constraints.

To determine if $\boldsymbol{x}$ is elicitable with a specified reward function and budget, we search for reward-improving directions within $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$. If such a reward-improving direction exists, this certifies that $\boldsymbol{x}$ is not elicitable by providing a feasible output vector that the agent strictly prefers over $\boldsymbol{x}$. On the other hand if no reward-improving direction exists within $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$, then $\boldsymbol{x}$ is elicitable.

Power-characterizing condition. The power-characterizing condition for an aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is stated below based on directions in $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$, $k \in [K]$.

Definition 4.2. [Power-characterizing condition] Fix conic constraints $\boldsymbol{C}$. We say that the **power-characterizing condition** is satisfied for $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ if and only if one of the following two conditions hold:

1. *Feasibility expansion:* $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements feasibility-expansion for $\boldsymbol{C}$ or
2. *Strengthened support expansion or strengthened binding set contraction:* There exists $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ with $\boldsymbol{d} \not\leq 0$ such that for every $k \in [K]$, there exist $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}(\boldsymbol{x}^{(k)})|}$ and $\boldsymbol{\lambda}^{(k)} \in \mathbb{R}_{\geq 0}^{|\mathcal{S}(\boldsymbol{x}^{(k)})^c|}$ such that the following inequality holds:

$$\boldsymbol{w}^{(k)}\boldsymbol{d} \;+\; \left|\mathbb{1}^\top \boldsymbol{d}\right| \cdot \min_{j \in [M]} \min\left(\boldsymbol{w}_j^{(k)}, 0\right) \;>\; 0$$

where $\boldsymbol{w}^{(k)} := \left(\boldsymbol{\gamma}^{(k)}\right)^\top C_{\mathcal{V}(\boldsymbol{x}^{(k)})} \;-\; \left(\boldsymbol{\lambda}^{(k)}\right)^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}$.

This condition requires the aggregation operation is either feasibility-expanding or for every $k \in [K]$, it either implements a strengthened version of support expansion or a strengthened version of binding set contraction. We discuss this in more detail in Section 4.2, but briefly discuss the intuition here. The second condition implies that either $(\boldsymbol{\gamma}^{(k)})^\top C_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} > 0$ or $(\boldsymbol{\lambda}^{(k)})^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c} < 0$. The first case turns out to imply binding-set contraction, while the second case turns out to imply support expansion. Moreover, the condition strengthens the mechanisms in two ways. First, it requires the mechanisms to jointly be implemented across all of the agents, rather than implemented for each agent individually. Second, it requires violation of (weighted) conic and non-negativity constraints by a minimum margin, whereas the original mechanisms have no margin requirements.

Characterization results. Our main theorems state that the power-characterizing condition provides a full characterization of when an aggregation operation can expand elicitability under some feature map. For the first part of this characterization, we show that the power-characterization is sufficient for elicitability-expansion for some feature weights matrix.

Theorem 4.3 (Sufficient)**.** *Fix constraints $\boldsymbol{C}$ and aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$ is feasible (i.e., $\boldsymbol{C}\boldsymbol{x}^{(k)} \leq 0$, for every $k \in [K]$). If the power-characterizing condition is satisfied for $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$, then $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding relative to $\boldsymbol{C}$.*

To complete the characterization, our next main theorem states that power-characterizing condition is also necessary for an aggregation to have power. That is, if this condition does not hold, the aggregation operation faces a strong limitation of not offering elicitability-expansion for any feature weights matrix.

Theorem 4.4 (Necessary)**.** *Fix constraints $\boldsymbol{C}$ and aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$ is feasible (i.e., $\boldsymbol{C}\boldsymbol{x}^{(k)} \leq 0$, for every $k \in [K]$). If the power-characterizing condition is not satisfied, then $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is not elicitability-expanding relative to $\boldsymbol{C}$.*

Key proof ideas. Our main technical tool (Lemma 4.7 in Section 4.3) provides a geometric characterization of elicitability-expansion under a fixed feature map, extending Kleinberg and Raghavan [2020] to handle conic feasibility constraints and aggregated outputs. This lemma reduces the problem to checking whether two sets have empty intersection: the set of feasible, budget-reducing directions for vectors in the aggregation, and the cone of *feature-improving directions* $\{\boldsymbol{d} \in \mathbb{R}^M \mid \boldsymbol{\alpha}\boldsymbol{d} \geq \boldsymbol{0}\}$.

To move from elicitability-expansion under a *fixed* feature map to elicitability-expansion under *any* feature map, we consider how to construct feature weights $\boldsymbol{\alpha}$ that make a set $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}$ elicitable while rendering $\boldsymbol{x}^{(A)}$ inelicitable. The key subtlety is that *feature-improving directions* form a cone. So non-negative scaling and translations of improving directions remain improving. As a result, we cannot simply find a single separating direction for $\boldsymbol{x}^{(A)}$—we need a robust geometric condition guaranteeing separation that survives all such transformations. We capture this robustness through a minimax argument, showing it is equivalent to the existence of a worst-case distribution over translations that certifies inelicitability. We elaborate on these ideas in Section 4.4.

4.2 Connection of the power-characterizing condition to mechanisms

The power-characterizing condition requires one of two cases to hold. The first is implementation of feasibility expansion. We can interpret the second case as unifying a strengthening of support expansion and a strengthening of binding set contraction into a single inequality. To demonstrate the connection between the power-characterizing condition and the mechanisms, we will first show how the power-characterizing condition implies implementation of one of the mechanisms. We will then show how the power-characterizing condition strengthens the mechanisms.

Power-characterization condition implies the mechanisms. The following result shows that the power-characterizing condition implies that at least one of feasibility-expansion, support expansion, or binding

set contraction is implemented. This result immediately implies Theorem 3.7 (i.e., that implementing at least one of these mechanisms is necessary for elicitability-expansion).

Proposition 4.5. *Fix conic constraints $\boldsymbol{C}$, and any aggregation operation $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$ is feasible i.e., $\boldsymbol{C}\boldsymbol{x}^{(k)} \leq 0$, for every $k \in [K]$. If $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ satisfies the power-characterizing condition (Definition 4.2), then one of the following conditions holds:*

- *$\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is feasibility-expanding relative to $\boldsymbol{C}$ (Definition 3.1).*
- *For each $k \in [K]$, $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is either support-expanding relative to k (Definition 3.3) or binding set-contracting relative to k (Definition 3.5).*

Proof. The first case for the power-characterizing condition to hold is feasibility expansion. Suppose feasibility expansion does not hold, i.e., all vectors in the aggregation are feasible. Then the second case of Definition 4.2 holds: there exists $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ with $\boldsymbol{d} \not\leq 0$ such that for every $k \in [K]$, there exist $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}(\boldsymbol{x}^{(k)})|}$ and $\boldsymbol{\lambda}^{(k)} \in \mathbb{R}_{\geq 0}^{|\mathcal{S}(\boldsymbol{x}^{(k)})^c|}$ with

$$\boldsymbol{w}^{(k)}\boldsymbol{d} \;+\; |\mathbb{1}^\top \boldsymbol{d}| \cdot \min_{j \in [M]} \min\big(\boldsymbol{w}_j^{(k)},\, 0\big) \;>\; 0,$$

where $\boldsymbol{w}^{(k)} := \big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})} - \big(\boldsymbol{\lambda}^{(k)}\big)^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}$. Since $\min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0) \leq 0$ and $|\mathbb{1}^\top \boldsymbol{d}| \geq 0$, the second term on the left-hand side is non-positive, so the inequality implies $\boldsymbol{w}^{(k)}\boldsymbol{d} > 0$, i.e.,

$$\big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} \;>\; \big(\boldsymbol{\lambda}^{(k)}\big)^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}\boldsymbol{d}.$$

We will show that for each $k \in [K]$, this strict inequality implies either binding set contraction relative to k or support expansion relative to k.

Case 1: $\big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} > 0$ implies binding set contraction relative to k. Since $\boldsymbol{\gamma}^{(k)} \geq 0$ and $\big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} = \sum_{\ell \in \mathcal{V}(\boldsymbol{x}^{(k)})} \boldsymbol{\gamma}_\ell^{(k)} \boldsymbol{C}_\ell \boldsymbol{d} > 0$, there exists $\ell \in \mathcal{V}(\boldsymbol{x}^{(k)})$ with $\boldsymbol{C}_\ell \boldsymbol{d} > 0$. However, $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ requires $\boldsymbol{C}_\ell \boldsymbol{d} \leq 0$ for every $\ell \in \mathcal{V}(\boldsymbol{x}^{(A)})$. So $\ell \in \mathcal{V}(\boldsymbol{x}^{(k)}) \setminus \mathcal{V}(\boldsymbol{x}^{(A)})$, which is evidence that $\mathcal{V}(\boldsymbol{x}^{(A)}) \not\supseteq \mathcal{V}(\boldsymbol{x}^{(k)})$, i.e., binding set contraction relative to k.

Case 2: $\big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} \leq 0$ implies support expansion relative to k. In this case, the strict inequality forces $\big(\boldsymbol{\lambda}^{(k)}\big)^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}\boldsymbol{d} < 0$, i.e., $\sum_{j \in \mathcal{S}(\boldsymbol{x}^{(k)})^c} \boldsymbol{\lambda}_j^{(k)} \boldsymbol{d}_j < 0$. So there exists $j \in \mathcal{S}(\boldsymbol{x}^{(k)})^c$ with $\boldsymbol{d}_j < 0$. Since $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ requires $\boldsymbol{d}_j \geq 0$ for every $j \in \mathcal{S}(\boldsymbol{x}^{(A)})^c$, we have $j \notin \mathcal{S}(\boldsymbol{x}^{(A)})^c$, i.e., $j \in \mathcal{S}(\boldsymbol{x}^{(A)}) \setminus \mathcal{S}(\boldsymbol{x}^{(k)})$. This is evidence that $\mathcal{S}(\boldsymbol{x}^{(A)}) \not\subseteq \mathcal{S}(\boldsymbol{x}^{(k)})$, i.e., support expansion relative to k. □

How the power-characterizing condition strengthens the mechanisms. While the power-characterizing condition implies the implementation of one of the mechanisms, it is a strictly stronger condition. This is evidenced by the fact that the power-characterizing condition is necessary and sufficient for elicitability-expansion while implementing one of the mechanisms is not sufficient for elicitability-expansion as shown by propositions in Section A.1.

The power-characterizing condition, specifically the second case of Definition 4.2, strengthens the mechanisms in two ways.

- First, we require a mechanism to jointly be implemented across all of the agents, rather than implemented for each agent individually. That is, we place a *joint* requirement across all $k \in [K]$, requiring that the same $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ witnesses the violation of binding constraints for every $k \in [K]$.
- Second, we strengthen the mechanism for each agent individually, requiring there to be a sufficient gap between the original outputs $\boldsymbol{x}^{(k)}$ and the aggregated output $\boldsymbol{x}^{(A)}$. Implementing support expansion or binding set contraction only requires that $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ violate some binding non-negativity or conic constraint of each $\boldsymbol{x}^{(k)}$, i.e., that $\boldsymbol{w}^{(k)}\boldsymbol{d} > 0$ for some non-negative weighting

$\boldsymbol{w}^{(k)} = (\boldsymbol{\gamma}^{(k)})^\top \boldsymbol{C}_{\mathcal{V}(\boldsymbol{x}^{(k)})} - (\boldsymbol{\lambda}^{(k)})^\top I_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}$ of these binding constraints. Such a violation can hold with $\boldsymbol{x}^{(A)}$ arbitrarily close to any of the $\boldsymbol{x}^{(k)}$ vectors. In contrast, the power-characterizing condition requires this violation to hold by a minimum margin: rearranging the inequality in Definition 4.2, it asks that

$$\boldsymbol{w}^{(k)}\boldsymbol{d} \;>\; \left|\mathbb{1}^\top \boldsymbol{d}\right| \cdot \left|\min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0)\right|,$$

which is a strictly positive margin whenever $\boldsymbol{w}^{(k)}$ has a negative coordinate. For example, taking $\boldsymbol{\gamma}^{(k)} = \mathbf{0}$ and $\boldsymbol{\lambda}^{(k)}$ supported on a single index $j \in \mathcal{S}(\boldsymbol{x}^{(k)})^c$ gives $\boldsymbol{w}^{(k)}\boldsymbol{d} = -\boldsymbol{d}_j$ and recovers the requirement $-\boldsymbol{d}_j > |\mathbb{1}^\top \boldsymbol{d}|$, so $\boldsymbol{d}$ must violate that binding non-negativity constraint by a margin of $|\mathbb{1}^\top \boldsymbol{d}|$. This margin forces $\boldsymbol{x}^{(A)}$ to be sufficiently distinct from each $\boldsymbol{x}^{(k)}$.

The power-characterizing condition is in general a strictly stronger condition than implementing one of the mechanisms, as reflected by counterexamples in Section A.1. However, in some special cases, the power-characterizing condition corresponds exactly to implementing one of the mechanisms, instead of implementing a strengthening. One special case is when no vector in the aggregation operation has any binding conic constraints. This holds when there are no conic constraints. In this special case, even the regular, non-strengthened form of binding set contraction cannot kick in. We can show that in this special case, the power-characterizing condition reduces to either feasibility-expansion or the usual, non-strengthened support expansion as long as a particular edge case does not occur (the proof is deferred to Section B.3).

Corollary 4.6. *Fix conic constraints $\boldsymbol{C}$, and any $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$. Suppose that $\mathcal{V}(x^{(A)}) = \mathcal{V}(x^{(1)}) = \ldots = \mathcal{V}(x^{(K)}) = \emptyset$. Suppose also that either $\boldsymbol{x}^{(A)}$ is not full support (i.e., $\mathcal{S}(\boldsymbol{x}^{(A)}) \neq [M]$) or there exists $j \in [M]$ such that no $\boldsymbol{x}^{(k)}$ has support $[M]\setminus\{j\}$. Then $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding relative to $\boldsymbol{C}$ if and only if at least one of the following two conditions holds:*

- *$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is feasibility-expanding.*
- *$\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is support-expanding relative to every $k \in [K]$.*

Note that it appears harder to show an analogous result for binding-set contraction. This is due to the joint geometry of the constraints that appears in the power-characterizing condition.

4.3 Main lemma: Elicitability-expansion for a given feature map

In this section, we will build tools to prove our main characterization result. The main technical lemma is the characterization of elicitability-expansion under a given feature map $\boldsymbol{\alpha}$. This lemma extends the characterization of previous work Kleinberg and Raghavan [2020] for elicitability of an output vector under a feature map α in a setting without conic constraints on the agent's output vector. We extend this characterization to allow for output limitations (i.e., nontrivial constraints $\boldsymbol{C}$) and aggregation of multiple outputs.

This characterization depends on the set $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$ of feasible, budget-reducing directions (Definition 4.1) of the vectors $\boldsymbol{x}$ in the aggregation operation. The condition checks for intersection between these sets and the set of feature-improving directions $\left\{\boldsymbol{d} \in \mathbb{R}^M \mid \boldsymbol{\alpha}\boldsymbol{d} \geq \mathbf{0}\right\}$ consisting of directions that weakly increase *all* feature values.

Lemma 4.7. *Fix conic constraints $\boldsymbol{C}$, feature weights matrix $\boldsymbol{\alpha}$, and aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ where each $\boldsymbol{x}^{(k)}$ is feasible (i.e., $\boldsymbol{C}\boldsymbol{x}^{(k)} \leq 0$, for every $k \in [K]$). The aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding relative to $\boldsymbol{C}$ and $\mathbf{F}$ if and only if both of the following conditions hold:*

- *For every $k \in [K]$, $\mathcal{B}_{\mathcal{S}(x^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})} \cap \left\{\boldsymbol{d} \in \mathbb{R}^M \mid \boldsymbol{\alpha}\boldsymbol{d} \geq \mathbf{0}\right\} = \emptyset$.*
- *$\mathcal{B}_{\mathcal{S}(x^{(A)}),\mathcal{V}(x^{(A)})} \cap \left\{\boldsymbol{d} \in \mathbb{R}^M \mid \boldsymbol{\alpha}\boldsymbol{d} \geq \mathbf{0}\right\} \neq \emptyset$ or $\boldsymbol{C}\boldsymbol{x}^{(A)} \not\leq 0$.*

Lemma 4.7 characterizes the power of aggregation for a given feature weights matrix $\boldsymbol{\alpha}$. As a result, the characterizing condition depends on both the reward function specification limitation (which reflects prompt engineering limitations) via $\boldsymbol{\alpha}$ and the output limitation (which reflect model capability limitations) via the conic constraints $\mathbf{C}$. This result illustrates how both forms of limitations influence the power of aggregation.

Proof ideas. The full proof of Lemma 4.7 appears in Section B.2. The key idea is that elicitability of an output vector $\boldsymbol{x}$ is characterized by whether the set of feasible perturbation directions, $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$, intersects the set $\{\boldsymbol{d} \in \mathbb{R}^M : \boldsymbol{\alpha d} \geq \boldsymbol{0}\}$. Lemmas B.3 and B.4 establish the necessity and sufficiency of this condition. This characterization yields conditions under which each individual output $\boldsymbol{x}^{(k)}$ for $k \in [K]$ is elicitable, while the aggregate output $\boldsymbol{x}^{(A)}$ is not, which is precisely the notion of elicitability expansion.

To prove the lemma, one direction is straightforward: if the intersection is nonempty, moving in any direction $\boldsymbol{d}$ in the intersection maintains feasibility and strictly increases every monotone reward over the features, resulting in a feasible vector strictly preferred over $\boldsymbol{x}$, providing a certificate that $\boldsymbol{x}$ cannot be elicitable.

The other direction of the characterization is more involved and shows that whenever the intersection is empty, there is a reward function that elicits $\boldsymbol{x}$. We can write the empty intersection as infeasibility of a system of linear inequalities. We certify emptiness via duality, and the dual certificate directly yields a reward function that elicits the desired input vectors. Specifically, using Motzkin's transposition, we can equivalently write this as a certificate in terms of dual variables. The dual variables from this certificate along with a linear reward function constructed based on these variables demonstrate optimality of $\boldsymbol{x}$ for the reward-maximization program by satisfying the KKT conditions of this program.

4.4 Proof ideas for main theorems

In this section, we will provide proof sketches for our main theorems: Theorem 4.4 and Theorem 4.3 that provide the characterization of elicitability-expansion. We will build on the technical lemma (Lemma 4.7), but move beyond specific feature weights matrices $\boldsymbol{\alpha}$ to reason about elicitability-expansion under *some* feature weights matrix $\boldsymbol{\alpha}$.

Based on Lemma 4.7, one might expect that elicitability-expansion occurs exactly when $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ either implements feasibility-expansion or if there exists a direction $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ that does not exist in $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$ for any $k \in [K]$. We might hope that the existence of such a $\boldsymbol{d}$ allows us to construct feature weights matrix $\boldsymbol{\alpha}$ where $\boldsymbol{d}$ is the only feature-improving direction, which would make $\boldsymbol{x}^{(A)}$ not elicitable while each $\boldsymbol{x}^{(k)}$ is elicitable.

However, it is not possible to construct $\boldsymbol{\alpha}$ with $\boldsymbol{d}$ being the only feature-improving direction. If $\boldsymbol{d}$ is a feature-improving direction (i.e., $\boldsymbol{\alpha d} \geq 0$), then every direction $\boldsymbol{u} + \lambda \boldsymbol{d}$ for $\boldsymbol{u}, \lambda > 0$ is also a feature-improving direction. Due to this property, a stronger empty intersection condition than each $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$ and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ having empty intersection turns out to be necessary for elicitability expansion. This stronger condition is stated below and we will show that this is actually equivalent to the power-characterizing condition.

Definition 4.8. Fix constraints $\boldsymbol{C}$ and aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$. We say that the **alternate power-characterizing condition** is satisfied for $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ if (1) $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements feasibility-expansion, or (2) there exists $\boldsymbol{d}^{(A)} \in \mathcal{B}_{\mathcal{S}(x^{(A)}),\mathcal{V}(x^{(A)})}$ with $\boldsymbol{d}^{(A)} \not\leqq 0$ such that:

$$\left\{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)} \mid \boldsymbol{u} \in \mathbb{R}^M_{\geq 0}, \lambda \geq 0\right\} \bigcap \left(\bigcup_{k \in [K]} \mathcal{B}_{\mathcal{S}(x^{(k)}),\mathcal{V}(x^{(k)})} \right) = \varnothing.$$

Equivalence of both forms of power-characterizing condition. The alternate power-characterizing condition turns out to be exactly equivalent to the power-characterizing condition in Definition 4.2. as stated in Proposition B.6.

We provide a brief proof sketch of Proposition B.6 here. The full proof is in Section B.4. Note that Definition 4.2 is stated in terms of a $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}\boldsymbol{x}^{(A)},\mathcal{V}\boldsymbol{x}^{(A)}}$ violating constraints by a margin, whereas Definition 4.8 is in terms of all non-negative scaled translations $\boldsymbol{u} + \lambda \boldsymbol{d}$ violating constraints. Let us now describe how violation by a margin relates to violation after non-negative translation. For simplicity, let us assume there is a single constraint. That is $\boldsymbol{C}$ has a single row.

To prevent against any budget-reducing $\boldsymbol{u} + \lambda \boldsymbol{d}$ satisfying the constraint, we need to prevent against the optimal choice of $\boldsymbol{u}$ picked for constraint satisfaction. This is the $\boldsymbol{u}$ that minimizes $\boldsymbol{C}\boldsymbol{u}$ subject to the budget-decreasing constraint, which is that $\|\boldsymbol{u}\|_1 \leq \lambda |\mathbb{1}^t d|$. The minimizing $\boldsymbol{u}$ is the zero vector if $\boldsymbol{C}$ has all positive coordinates. Otherwise, it places all of its weight on the most negative coordinate of $\boldsymbol{C}$. Ensuring that the minimum value of $\boldsymbol{C}\boldsymbol{u}$, which is $\lambda |\mathbb{1}^t \boldsymbol{d}| \min_{j \in [n]}(0, \boldsymbol{C}_j)$, is lower than $\lambda \boldsymbol{C}\boldsymbol{d}$ is exactly the violation by a margin condition from the power-characterization condition.

Extending to multiple constraints requires ensuring that no single translation $\boldsymbol{u}$ can satisfy all constraints simultaneously. Rather than maximizing satisfaction of a single constraint, we must now consider the max-min problem: choosing $\boldsymbol{u}$ to maximize the minimum level of satisfaction across all constraints. By the minimax theorem, this max-min value equals the min-max value obtained by first choosing a weighting over constraints and then choosing $\boldsymbol{u}$ to maximize satisfaction of the resulting weighted constraint. This duality reduces the multi-constraint case to the single-constraint analysis above, applied to the worst-case weighted combination.

Proof sketch of Theorem 4.4. The full proof is provided in Section B.5. It uses the ideas from the proof of Theorem 3.7 which shows the need for a feasible, budget-reducing direction of $\boldsymbol{x}^{(A)}$ that is not a feasible, budget-reducing direction for any other $\boldsymbol{x}^{(k)}$ for $k \in [K]$. Only then can there be a feature-improving direction that is viable for $\boldsymbol{x}^{(A)}$ but not any of the $\boldsymbol{x}^{(k)}$'s to result in elicitability of each $\boldsymbol{x}^{(k)}$ but not of $\boldsymbol{x}^{(A)}$. However, since all non-negative scaling and translations of feature-improving directions continue to be feature-improving, we further require positive scaling and translation of some direction $\boldsymbol{d}^{(A)} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}), \mathcal{V}(\boldsymbol{x}^{(A)})}$ to not be present in any $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}), \mathcal{V}(\boldsymbol{x}^{(k)})}$, $k \in [K]$. This is precisely the condition of the alternate power-characterizing condition.

Proof sketch of Theorem 4.3. The full proof is provided in Section B.6. We have shown why a stronger condition is still necessary for elicitability-expansion. Now let us show that this stronger condition is also sufficient. We show this by explicitly constructing a feature map $\boldsymbol{\alpha}$ for which the set $\{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)} \mid \boldsymbol{u} \in \mathbb{R}^M_{\geq 0}, \lambda \geq 0\}$ contains all feature-improving directions i.e., the set $\{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$. In the construction, feature improving directions can have any value on the positive coordinates of $\boldsymbol{d}^{(A)}$. But their negative coordinates cannot be too large compared to the positive coordinates.

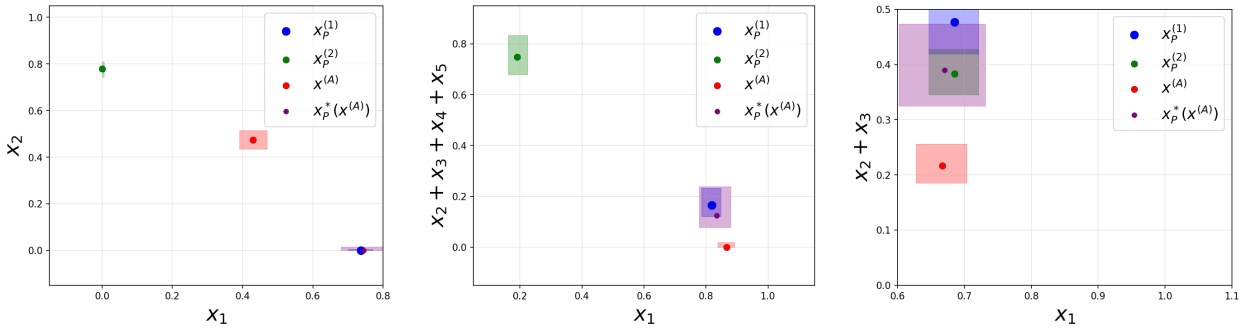
4.5 Conceptual insights for system designers

Our characterization results offer conceptual insights into when system designers benefit from specific aggregation operations in compound AI systems.

General takeaways. Our results characterize how the interplay between prompt engineering limitations and model capability limitations affects which types of aggregation operations are useful. Specifically, aggregation not only overcomes model capability limitations (feasibility expansion), but also overcomes prompt engineering limitations through combining multiple output characteristics (support expansion) and through taking advantage of output-level limitations (binding set-contraction). Notably, even as model capabilities continue to improve, the latter two mechanisms mean that aggregation can still be useful to system designers. On the flip side, our results illustrate how aggregation operations that do not take advantage of these mechanisms offer no power, regardless of whether the system designer employs sophisticated or unsophisticated prompt engineering practices.

Illustrative example. As an illustrative example, consider intersection aggregation $\mathcal{A}_{\text{intersect}}$ and addition aggregation $\mathcal{A}_{\text{add}}$.

- **Intersection aggregation:** Intersection aggregation combines outputs based on commonality among different output vectors, which is conceptually similar to debate protocols [Du et al., 2024] that aim to create agreement or inference scaling methods that aim to filter out incorrect information [Zhang et al., 2025]. Intersection aggregation can implement feasibility expansion and binding-set contraction, but not support expansion (Table 2 in Section B.7). However, feasibility expansion and binding-set contraction



(a) Support expansion (b) Binding set contraction (c) Feasibility expansion

Figure 3: Empirical illustration of the three mechanisms in a toy reference-generation task with GPT-4o-mini at temperature 0.7 (Section 5). Output dimensions capture paper topics, prompt dimensions capture (potentially higher-level) topics, and aggregation takes a union or an intersection of the lists. The plots show output vectors generated by the individual prompts ($\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$) and by the aggregation operation ($\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$), along with the output vector closest to $\boldsymbol{x}^{(A)}$ that is elicitable by a single model with prompt topics ($\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$). The shaded regions show confidence sets. Each plot shows an aggregation operation that implements one of the mechanisms—support expansion (left), binding-set contraction (middle), and feasibility expansion (right). The plots show that these aggregation operations are all elicitability-expanding. The numerical values are shown in Table 1. This plot is an empirical analogue of Figure 1.

fundamentally rely on model capability limitations. If models are very capable (i.e., if they face no conic constraints), Theorem 4.4 demonstrates that intersection aggregation never adds power, regardless of whether the system designer employs sophisticated or unsophisticated prompt engineering practices.

- **Addition aggregation:** Addition aggregation interpolates among different output directions, which conceptually captures how system designers synthesize multiple outputs to delegate specialized subtasks to each agent and synthesize the outputs of these subtasks [BAIR Research Blog, 2024, Anthropic, 2025]. Addition aggregation can implement support expansion (Table 2 in Section B.7) and can actually implement the strengthened form of support expansion as well (Example 3.4). Theorem 4.3 shows that addition aggregation adds power even when models face no capability limitations (i.e, no conic constraints), at least for some level of prompt engineering limitations.

5 Empirical Illustration using LLMs

In this section, we provide empirical support using LLMs for each of these three mechanisms and show they give power to the system designer.[4] Specifically, we aim to construct an aggregation operation for each operation that implements each mechanism and expands elicitability. Like in Section 2.4, we focus on a reference generation task. These experiments illustrate the robustness of our findings beyond the assumptions of our theoretical framework (e.g., reward-maximization and conic constraints).

Instantiation of our model. The task is to generate a list of titles of papers that reflects specific topics. We instantiate this task to capture key aspects of our model such as reward functions operating over coarser features, but we depart from some of the specifics such as linearity of the features and the monotonicity assumption on the reward functions. Each problem instance is defined by output dimension topics $\mathcal{T}^O$, prompt topics $\mathcal{T}^P$, and an aggregation operation $\mathcal{A}^\cup$ or $\mathcal{A}^\cap$ as we describe below.

[4]Code is available at https://github.com/nivasini/aggregation_compound_ai.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.70, 0.77]$	$[0.00, 0.00]$	$[0.04, 0.08]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.74, 0.81]$	$[0.03, 0.06]$
$\boldsymbol{x}^{(A)}$	$[0.39, 0.47]$	$[0.43, 0.51]$	$[0.03, 0.06]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.68, 0.80]$	$[0.00, 0.01]$	$[0.05, 0.12]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.65, 0.72]$	$[0.36, 0.44]$	$[0.06, 0.10]$
$\boldsymbol{x}^{(2)}$	$[0.65, 0.72]$	$[0.00, 0.00]$	$[0.35, 0.42]$
$\boldsymbol{x}^{(A)}$	$[0.63, 0.70]$	$[0.00, 0.00]$	$[0.19, 0.25]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.60, 0.73]$	$[0.00, 0.01]$	$[0.33, 0.46]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.79, 0.85]$	$[0.09, 0.14]$	$[0.00, 0.02]$	$[0.00, 0.02]$	$[0.03, 0.06]$
$\boldsymbol{x}^{(2)}$	$[0.16, 0.22]$	$[0.00, 0.01]$	$[0.61, 0.69]$	$[0.04, 0.08]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(A)}$	$[0.84, 0.89]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.78, 0.88]$	$[0.07, 0.16]$	$[0.00, 0.01]$	$[0.00, 0.03]$	$[0.00, 0.04]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.63, 1.02]
Feasibility expansion	[0.07, 0.39]
Binding-set contraction	[0.07, 0.35]

(d) ℓ_1 distance between $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$

Table 1: Empirical illustration of the three mechanisms in a toy reference-generation task with GPT-4o-mini at temperature 0.7 (Section 5). We show the exact values from Figure 3 for support expansion, feasibility expansion, and binding-set contraction. We also show the ℓ_1 distance between $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$. All of our quantities are reported as 95% confidence intervals.

- *Outputs:* Given a reference list L, we let the output vector $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^M$ capture the extent to which the list is perceived to reflect different topics $\mathcal{T}^O := \left\{\mathcal{T}_i^O\right\}_{i\in[M]}$. That is, each output dimension $i \in [M]$ corresponds to a topic T_i^O, and x_i captures the fraction of papers in L that are perceived to belong to the paper topic x_i.
- *Reward function specification:* Prompting[5] is also based on paper topics $\mathcal{T}^P := \left\{\mathcal{T}_j^P\right\}_{j\in[N]}$, which tend to be broader in scope than the output dimension topics $\left\{\mathcal{T}_i^O\right\}_{i\in[M]}$ in many of our setups. Each prompt is specified by a set of topics to include $G^{\text{inc}} \subseteq \mathcal{T}^P$, a connector operation $\text{op}^{\text{inc}} \in \{\text{and}, \text{or}\}$ between the included topics, a disjoint set of topics $G^{\text{exc}} \subseteq \mathcal{T}^P$ (where $G^{\text{inc}} \cap G^{\text{exc}} = \emptyset$) to explicitly exclude, and a connector operation $\text{op}^{\text{exc}} \in \{\text{and}, \text{or}\}$ between the excluded topics. If an inclusion or exclusion set is empty or size one, we omit the corresponding connector operation.
- *Aggregation:* We consider two types of aggregation rules: $\mathcal{A}^{\cup}$ takes a union of the lists (based on $\mathcal{A}_{\text{add}}$), and $\mathcal{A}^{\cap}$ takes a intersection of the lists (based on $\mathcal{A}_{\text{intersect}}$).

We build on our model to capture elicitability, but relax these concepts to allow for closeness in output vectors rather than necessarily requiring an exact match. We define the closeness of two output vectors $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^M$ and $\boldsymbol{x}' \in \mathbb{R}_{\geq 0}^M$ in terms of their ℓ_1 norm i.e., $\|\boldsymbol{x} - \boldsymbol{x}'\|_1$. We capture elicitability using prompting over the prompt topics $\mathcal{T}^P$. An output vector $\boldsymbol{x}$ is *elicitable* by a single model if there exists a prompt specification $(G^{\text{inc}} \subseteq \mathcal{T}^P, \text{op}^{\text{inc}}, G^{\text{exc}} \subseteq \mathcal{T}^P, \text{op}^{\text{exc}})$ over prompt topics that produces an output vector close to $\boldsymbol{x}$. Given a target vector $\boldsymbol{x}_{\text{target}}$, let $\boldsymbol{x}_P^*(\boldsymbol{x}_{\text{target}})$ denote the output vector closest to $\boldsymbol{x}_{\text{target}}$ that can be produced with some prompt specification over prompt topics.

Empirical setup. All of our experiments use GPT-4o-mini at temperature 0.7. For prompting, we convert a specification $(G^{\text{inc}}, \text{op}^{\text{inc}}, G^{\text{exc}}, \text{op}^{\text{exc}})$ into natural language by connecting the included topics using op^{inc}, and then saying "excluding" followed by connecting the excluded topics using op^{exc}. We use an LLM-as-a-judge (GPT-4o-mini at temperature 0) to classify whether each paper title falls within the topic, and then we

[5]For simplicity, in this setup we do not consider budget specification.

compute the fraction of papers that are classified as positive.[6] To study elicitability, we do a brute force search over the space of prompt specifications. We average over 30 trials to determine the output vector $\boldsymbol{x} \in \mathbb{R}^M_{\geq 0}$ associated with a given prompt specification. We defer other details of our setup to Appendix D.

In Appendix E, we show that our empirical results readily generalize to different model configurations. Specifically, we consider different models (GPT-5.4, GPT-5-mini), different temperatures, and aggregation of heterogeneous models. Our experiments provide empirical support for all three mechanisms in almost all of the experimental variations. This robustness illustrates that our findings hold beyond the specific assumptions of our theoretical framework.

For each mechanism, we aim to construct an aggregation operation $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ that implements the mechanism and also is elicitability-expanding.

5.1 Results

We show our results in Figure 3 and Table 1, and we explain them in detail below.

Support Expansion. We construct a setup that resembles Example 3.4 shown in Figure 1b. Let there be $M = 3$ output dimensions and $N = 2$ prompt features. The output dimension topics are:

- $\mathcal{T}_1^O$: complexity theory (computational complexity, P vs NP, complexity classes, hardness results, reductions, circuit complexity, space/time complexity bounds)
- $\mathcal{T}_2^O$: macroeconomics (GDP, inflation, monetary policy, fiscal policy, business cycles, economic growth, unemployment, central banking, aggregate demand/supply)
- $\mathcal{T}_3^O$: mechanism design (auction design, incentive compatibility, social choice, matching markets, market design, algorithmic game theory, incentive mechanisms)

The prompt topics are $\mathcal{T}_1^P$: computer science theory and $\mathcal{T}_2^P$: economics. We take the aggregation rule to be $\mathcal{A}^{\cup}$. We let $\boldsymbol{x}^{(1)}$ capture the output from prompt specification $(G_1^{\text{inc}} = \left\{\mathcal{T}_1^P\right\}, G_1^{\text{exc}} = \emptyset)$, and we let $\boldsymbol{x}^{(2)}$ capture the output from prompt specification $(G_2^{\text{inc}} = \left\{\mathcal{T}_2^P\right\}, G_1^{\text{exc}} = \emptyset)$.

The results are shown in Figure 3a and Table 1a. We find on average that $\boldsymbol{x}^{(A)} = [0.43, 0.47, 0.04]$, that $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) = [0.74, 0.00, 0.07]$. We obtain a confidence bound of $[0.63, 1.02]$ for ℓ_1 distance between $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$. The vector $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is obtained via brute-force search over prompt specifications.

These results suggest that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ implements support expansion and is elicitability-expanding. Specifically, $\boldsymbol{x}^{(1)}$ has most of its nonzero weight on topic $\mathcal{T}_1^O$, $\boldsymbol{x}^{(2)}$ has most of its nonzero weight on topic $\mathcal{T}_2^O$, and $\boldsymbol{x}^{(A)}$ reflects both topics $\mathcal{T}_1^O$ and $\mathcal{T}_2^O$. $\boldsymbol{x}^{(A)}$ is not elicitable by a single model, since $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ misses out on $\mathcal{T}_2^O$. This illustrates that no prompt is able to simultaneously activate topics $\mathcal{T}_1^O$ and $\mathcal{T}_2^O$, but aggregation is able to overcomes this limitation by separately considering $\mathcal{T}_1^O$ and $\mathcal{T}_2^O$. On the other hand, if we prompt using output dimension topics $\mathcal{T}^O$ as opposed to the prompt topics $\mathcal{T}^P$, it is possible to simultaneously activate both topics: we find on average that the output vector equals $[0.52, 0.51, 0.00]$ if we use the prompt specification $(G^{inc} = \{\mathcal{T}_1^O, \mathcal{T}_2^O\}, \text{op}^{\text{inc}} = \text{or}, G^{exc} = \emptyset)$.

Binding set contraction. We construct an example that resembles Example 3.6 shown in Figure 1c, but with a greater number of output dimensions. Let there be $M = 5$ output dimension topics and $N = 2$ prompt

[6] Note this reflects the perception of the paper's category given the title, rather than whether the paper truly belongs to the category at hand.

dimension topics. The output dimension topics are:

$\mathcal{T}_1^O$: deep learning (transformers, attention mechanisms, deep neural networks, modern architectures like BERT, GPT, ViT)

$\mathcal{T}_2^O$: non-transformer NLP methods (RNN, LSTM, GRU, word2vec, seq2seq without attention, traditional NLP)

$\mathcal{T}_3^O$: non-transformer CV methods (CNN, ConvNets, ResNet, VGG, pooling, convolutional architectures)

$\mathcal{T}_4^O$: multimodal methods bridging multiple modalities (combining text and images, vision-language models, image captioning, VQA)

$\mathcal{T}_5^O$: statistical machine learning (non-neural methods like SVM, random forests, logistic regression, Bayesian methods, traditional ML))

The prompt dimension topics are $\mathcal{T}_1^P$: natural language processing (NLP) and $\mathcal{T}_2^P$: computer vision (CV). We take the aggregation rule to be $\mathcal{A}^{\cap}$. We let $\boldsymbol{x}^{(1)}$ be the output from prompt specification ($G_1^{inc} = \{\mathcal{T}_1^P\}, G_1^{exc} = \{\mathcal{T}_2^P\}$), and we let $\boldsymbol{x}^{(2)}$ be the output from prompt specification ($G_2^{inc} = \{\mathcal{T}_2^P\}, G_2^{exc} = \{\mathcal{T}_1^P\}$).

The results are shown in Figure 3b and Table 1c. We find on average that $\boldsymbol{x}^{(A)} = [0.87, 0.00, 0.00, 0.00, 0.00]$, that $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) = [0.83, 0.11, 0.00, 0.01, 0.01]$. We get a confidence bound of $[0.07, 0.35]$ for ℓ_1 distance between $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$. The vector $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is obtained via brute-force search over prompt specifications.

The results suggest that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ implements binding-set contraction and it is elicitability-expanding. Specifically, since $\mathcal{T}_1^O$: deep learning is an umbrella topic covering many papers in topics $\mathcal{T}_2^O, \mathcal{T}_3^O, \mathcal{T}_4^O$, we expect that there is a semantic constraint capturing how lists of papers which include topics $\mathcal{T}_2^O$, $\mathcal{T}_3^O$ or $\mathcal{T}_4^O$ also tend to include topic $\mathcal{T}_1^O$. Both $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ appear to be binding relative to this constraint, while $\boldsymbol{x}^{(A)}$ is in the interior. $\boldsymbol{x}^{(A)}$ is not elicitable by a single model: while $\boldsymbol{x}^{(A)}$ predominantly reflects topic $\mathcal{T}_1^O$, the output vector $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ also reflects some topics outside of $\mathcal{T}_1^O$. On the other hand, if we prompt using output dimension topics $\mathcal{T}^O$ rather than the prompt topics $\mathcal{T}^P$, it is possible to slightly reduce the fraction of papers reflecting topics outside of $\mathcal{T}_1^O$: we find on average that the output vector equals $[0.96, 0.00, 0.01, 0.01, 0.00]$ if we use the prompt specification ($G^{inc} = \{\mathcal{T}_1^O\}, G^{exc} = \{\mathcal{T}_2^O, \mathcal{T}_3^O, \mathcal{T}_4^O, \mathcal{T}_5^O\}, \text{op}^{\text{exc}} = \text{or}$).

Feasibility Expansion. We construct a setup that resembles Example 3.2 shown in Figure 1a, but without the reward specification limitations. Let there be $M = 3$ output dimensions, corresponding to topics:

$\mathcal{T}_1^O$: blockchain(consensus protocols, smart contracts, decentralized ledgers, proof-of-work, proof-of-stake)

$\mathcal{T}_2^O$: cryptography (encryption, zero-knowledge proofs, hash functions, digital signatures, key exchange)

$\mathcal{T}_3^O$: distributed systems (consensus algorithms, fault tolerance, replication, distributed databases, CAP theorem)

We take $\mathcal{T}^P = \mathcal{T}^O$, which means that $M = N = 3$ and also that feasibility and elicitability are equivalent. We take the aggregation rule to be $\mathcal{A}^{\cap}$. We let $\boldsymbol{x}^{(1)}$ capture the output from prompt specification ($G_1^{\text{inc}} = \{\mathcal{T}_1^O, \mathcal{T}_2^O\}, \text{op}^{\text{inc}} = \text{or}, G_1^{\text{exc}} = \{\mathcal{T}_3^O\}$), and we let $\boldsymbol{x}^{(2)}$ capture the output from prompt specification ($G_2^{\text{inc}} = \{\mathcal{T}_1^O, \mathcal{T}_3^O\}, \text{op}^{\text{inc}} = \text{or}, G_2^{\text{exc}} = \{\mathcal{T}_2^O\}$).

The results are shown in Figure 3c and Table 1b. We find on average that $\boldsymbol{x}^{(A)} = [0.67, 0.00, 0.22]$, that $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) = [0.67, 0.00, 0.39]$. We get a confidence bound of $[0.07, 0.39]$ for the ℓ_1 distance between $\boldsymbol{x}^{(A)}$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$, which notably does not contain 0, illustrating a clear gap. The vector $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is obtained via brute-force search over prompt specifications.

The results suggest that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ implements feasibility expansion and is elicitability-expanding. Intuitively, we expect there to be the following constraint faced by agents (whether humans or LLMs) in this task: many blockchain papers ($\mathcal{T}_1^O$) are related to cryptography ($\mathcal{T}_2^O$) or to distributed systems ($\mathcal{T}_3^O$). Both $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ reflect this constraint, but $\boldsymbol{x}^{(A)}$ circumvents this constraint through aggregation. $\boldsymbol{x}^{(A)}$ is not

elicitable by a single model: $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ has substantially more weight on $\mathcal{T}_3^O$ (distributed systems) than $\boldsymbol{x}^{(A)}$ does, while matching on $\mathcal{T}_1^O$. The aggregation operation produces a vector whose distributed-systems content is markedly lower than what any single prompt can achieve.

6 Discussion

In this work, we theoretically study how aggregating multiple copies of the same model gives access to a greater set of outputs than using only a single model. Building on a principal-agent framework, our results show how aggregation must implement one of three mechanisms—feasibility-expansion, support expansion, and binding set contraction—in order to expand the set of elicitable outputs. Although these mechanisms are not sufficient to ensure that aggregation adds power, we show a more precise condition formed from strengthening the mechanisms is sufficient. Finally, we empirically illustrate these mechanisms by deploying LLMs in a toy reference-generation task, demonstrating the robustness of our findings beyond the assumptions of our stylized model.

Connecting our mechanisms to empirical phenomena. To complement our empirical support for the three mechanisms in the toy reference-generation task, we discuss how the mechanisms connect with more broadly observed empirical phenomena, which we hope inspire future empirical work. Since aggregation is only powerful when individual models are limited on their own, we begin by outlining the single-model limitations underlying each mechanism and the empirical phenomena supporting them.

- The power of feasibility expansion traces back to limitations in the types of outputs that individual models can generate: specifically, when models can't exhibit certain (desirable) dimensions without exhibiting other (undesirable) dimensions as a side effect. This side effect has been empirically observed for safety versus overrefusal, where models which refuse a larger fraction of toxic outputs tend to refuse a larger fraction of safe outputs as a side effect [Cui et al., 2024]. Similar side effects have been observed for alignment and hedging [Ouyang et al., 2022], and theoretically studied for creativity and factuality [Sinha et al., 2023].
- The power of support expansion traces back to challenges with eliciting outputs that perform along multiple dimensions at once in single-agent settings. This limitation has been empirically observed in cases where each dimension corresponds to a distinct user requirement. For example, prompts are often underspecified, since users may not include all of the requirements that they care about in the prompt [Yang et al., 2025]. Moreover, even when users specify all their requirements, LLMs struggle to satisfy many requirements simultaneously [Wen et al., 2024, Liu et al., 2025].

We leave pinpointing empirical phenomena which support binding set contraction—whose emergence depends on the interaction between prompt-engineering and model limitations—to future work. More broadly, since our results identify when aggregation enables these mechanisms, an important direction is to connect them to practice by testing whether real aggregation methods (e.g., debate [Du et al., 2024], prompt ensembling [Arora et al., 2022]) exhibit them. The single-model limitations discussed above suggest promising empirical settings where aggregation should add power.

Model limitations and extensions. Our stylized model, which builds on a classical principal-agent framework [Kleinberg and Raghavan, 2020], makes simplifying assumptions for tractability. First, we assume for simplicity that the agent chooses an output deterministically. Nonetheless, our empirical findings robustly generalize to LLMs with nonzero temperature which exhibit stochasticity (Section 5). It would be interesting to extend our theoretical results to capture this randomness and to allow the system designer to choose a temperature for each agent. Moreover, while our analysis allows for nonlinear rewards R, we restrict the output constraints (i.e., model limitations) and the feature map (i.e., prompt engineering limitations) to linear functional forms. Extending our model to allow for nonlinear limitations, which would complicate the structure of the agent's optimization program, is an interesting direction for future work. Furthermore, we also assume each agent's reward depends only on its own outputs, though richer interdependencies may arise in repeated, multi-turn interactions [Du et al., 2024]. Finally, while our analysis focuses on steering agents

through reward design, it would be interesting to incorporate other choices, such as tool use and fine-tuning, that enable specialization in compound AI systems [BAIR Research Blog, 2024].

7 Acknowledgments

We would like to thank Kate Donahue, Nika Haghtalab, Tatsu Hashimoto, Michael I. Jordan, and Manish Raghavan for useful feedback and discussions on the paper. This work was partially supported by a Stanford AI Lab postdoctoral fellowship.

References

Tal Alon, Magdalen Dobson, Ariel Procaccia, Inbal Talgam-Cohen, and Jamie Tucker-Foltz. Multiagent evaluation mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1774–1781, 2020.

Anthropic. How we built our multi-agent research system. https://www.anthropic.com/engineering/multi-agent-research-system, 2025. Blog post.

Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, Frederic Sala, and Christopher Ré. Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*, 2022.

BAIR Research Blog. Compound ai systems. https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/, 2024. Blog post.

Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–16, 2021.

Omer Ben-Porat and Moshe Tennenholtz. Regression equilibrium. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 173–191, 2019.

Philip Bond and Armando Gomes. Multitask principal–agent problems: Optimal contracts, fragility, and effort misallocation. *Journal of Economic Theory*, 144(1):175–211, 2009.

Arthur Campbell, Moshe Cohen, Florian Ederer, and Johannes Spinnewijn. *Solutions Manual to Accompany Contract Theory*, volume 1. The MIT Press, 2007.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Keertana Chidambaram, Karthik Vinary Seetharaman, and Vasilis Syrgkanis. Direct preference optimization with unobserved preference heterogeneity: The necessity of ternary preferences. *arXiv preprint arXiv:2510.15716*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, volume 30, 2017.

Natalie Collina, Surbhi Goel, Aaron Roth, Emily Ryu, and Mirah Shi. Emergent alignment via competition. *arXiv preprint arXiv:2509.15090*, 2025.

Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H Holliday, Bob M Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, et al. Social choice should guide ai alignment in dealing with diverse human feedback. *arXiv preprint arXiv:2404.10271*, 2024.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024.

Jessica Dai and Eve Fleisig. Mapping social choice theory to rlhf. *arXiv preprint arXiv:2404.13038*, 2024.

Krishna Dasaratha, Benjamin Golub, and Anant Shah. Incentive design with spillovers. *arXiv preprint arXiv:2411.08026*, 2024.

Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

Kate Donahue and Manish Raghavan. Impacts of aggregation on model diversity and consumer utility. Preprint, 2026.

Kate Donahue, Alexandra Chouldechova, and Krishnaram Kenthapadi. Human-algorithm collaboration: Achieving complementarity and avoiding unfairness. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1639–1656, 2022.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, pages 11733–11763. PMLR, 2024.

Ohad Einav and Nir Rosenfeld. A market for accuracy: Classification under competition. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Matthew Gentzkow and Emir Kamenica. Bayesian persuasion with multiple senders and rich signal spaces. *Games and Economic Behavior*, 104:411–429, 2017.

Paul Gölz, Nika Haghtalab, and Kunhe Yang. Distortion of ai alignment: Does preference optimization optimize for preferences? *arXiv preprint arXiv:2505.23749*, 2025.

Sanford J Grossman and Oliver D Hart. An analysis of the principal-agent problem. *Econometrica*, 51(1): 7–46, 1983.

Bengt Holmström. Moral hazard and observability. *The Bell journal of economics*, pages 74–91, 1979.

Bengt Holmstrom. Moral hazard in teams. *The Bell journal of economics*, pages 324–340, 1982.

Bengt Holmstrom and Paul Milgrom. Multitask principal–agent analyses: Incentive contracts, asset ownership, and job design. *The Journal of Law, Economics, and Organization*, 7(special_issue):24–52, 1991.

Safwan Hossain, Evi Micha, Yiling Chen, and Ariel Procaccia. Strategic classification with externalities. *arXiv preprint arXiv:2410.08032*, 2024.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. In *Forty-second International Conference on Machine Learning*, 2025.

Athul Paul Jacob, Yikang Shen, Gabriele Farina, and Jacob Andreas. The consensus game: Language model generation via equilibrium search. *arXiv preprint arXiv:2310.09139*, 2023.

Meena Jagadeesan, Michael Jordan, Jacob Steinhardt, and Nika Haghtalab. Improved bayes risk can yield reduced social welfare under competition. *Advances in Neural Information Processing Systems*, 36: 66940–66952, 2023.

Erik Jones, Anca Dragan, and Jacob Steinhardt. Adversaries can misuse combinations of safe models. In *Forty-second International Conference on Machine Learning*, 2025a.

Erik Jones, Arjun Patrawala, and Jacob Steinhardt. Uncovering gaps in how humans and llms interpret subjective language. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. In *International Conference on Machine Learning*, pages 23662–23733. PMLR, 2024.

Jon Kleinberg and Manish Raghavan. How do classifiers induce agents to invest effort strategically? *ACM Transactions on Economics and Computation (TEAC)*, 8(4):1–23, 2020.

Krishna K Ladha. The condorcet jury theorem, free speech, and correlated votes. *American Journal of Political Science*, pages 617–634, 1992.

Jean-Jacques Laffont and David Martimort. *The Theory of Incentives: The Principal-Agent Model*. Princeton University Press, Princeton, NJ, 2002.

Edward P Lazear and Sherwin Rosen. Rank-order tournaments as optimum labor contracts. *Journal of political Economy*, 89(5):841–864, 1981.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

Lydia T Liu, Nikhil Garg, and Christian Borgs. Strategic ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 2489–2518. PMLR, 2022.

Wenhao Liu, Zhengkang Guo, Mingchen Xie, Jingwen Xu, Zisu Huang, Muzhao Tian, Jianhan Xu, Muling Wu, Xiaohua Wang, Changze Lv, et al. Recast: Strengthening llms' complex instruction following with constraint-verifiable data. *arXiv preprint arXiv:2505.19030*, 2025.

Nancy A Lynch. *Distributed algorithms*. Elsevier, 1996.

Abhilash Mishra. Ai alignment and social choice: Fundamental limitations and policy implications. *arXiv preprint arXiv:2310.16048*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ori Press, Andreas Hochlehnert, Ameya Prabhu, Vishaal Udandarao, Ofir Press, and Matthias Bethge. Citeme: Can language models accurately cite scientific claims? *Advances in Neural Information Processing Systems*, 37:7847–7877, 2024.

Manish Raghavan. Competition and diversity in generative ai. *arXiv preprint arXiv:2412.08610*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Ali Shirali, Arash Nasr-Esfahany, Abdullah Alomar, Parsa Mirtaheri, Rediet Abebe, and Ariel Procaccia. Direct alignment with heterogeneous preferences. *arXiv preprint arXiv:2502.16320*, 2025.

Ritwik Sinha, Zhao Song, and Tianyi Zhou. A mathematical abstraction for balancing the trade-off between creativity and reality in large language models. *arXiv preprint arXiv:2306.02295*, 2023.

Margaret E Slade. Multitask agency and contract choice: An empirical exploration. *International Economic Review*, pages 465–486, 1996.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pages 330–337, 1993.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023a.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxing Xu, et al. Benchmarking complex instruction-following with multiple constraints composition. *Advances in Neural Information Processing Systems*, 37:137610–137645, 2024.

Renzhe Xu, Kang Wang, and Bo Li. Heterogeneous data game: Characterizing the model competition across multiple data sources. *arXiv preprint arXiv:2505.07688*, 2025.

Chenyang Yang, Yike Shi, Qianou Ma, Michael Xieyang Liu, Christian Kästner, and Tongshuang Wu. What prompts don't say: Understanding and managing underspecification in llm prompts. *arXiv preprint arXiv:2505.13360*, 2025.

Bohan Zhang, Xiaokang Zhang, Jing Zhang, Jifan Yu, Sijia Luo, and Jie Tang. Cot-based synthesizer: Enhancing llm performance through answer synthesis, 2025.

Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned ai. *Advances in Neural Information Processing Systems*, 33:15763–15773, 2020.

Appendix

Table of Contents

A Additional details and proofs for Section 3

A.1 Insufficiency of natural mechanisms implementation for elicitability-expansion

While implementing one of feasibility-expansion, support expansion, or binding set contraction is necessary for an aggregation operation to be elicitability-expanding as shown by Theorem 3.7, in this section, we will show that it is not sufficient.

Support expansion on its own or binding set contraction on its own is not sufficient for power. The following proposition states the insufficiency of support expansion relative to every input vector of the aggregation operation for elicitability-expansion.

Proposition A.1. *Fix $\boldsymbol{C} = \emptyset$. There exists an aggregation operation $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)} \to \boldsymbol{x}^{(A)}$ such that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)} \to \boldsymbol{x}^{(A)}$ implements support expansion relative to i for $i \in \{1, 2\}$. However, $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)} \to \boldsymbol{x}^{(A)}$ is not elicitability-expanding.*

Proof. This follows mainly from Corollary 4.6. This Corollary shows that an additional condition beyond support expansion is required for an operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ to be elicitability-expanding when $\mathcal{V}(\boldsymbol{x}^{(1)}) = \ldots = \mathcal{V}(\boldsymbol{x}^{(K)}) = \mathcal{V}(\boldsymbol{x}^{(A)})$. This is the condition that if $\boldsymbol{x}^{(A)}$ has full support, then there is a $j \in [M]$ and $k \in [K]$ such that $[M] \setminus \{j\} \not\subseteq \mathcal{S}(\boldsymbol{x}^{(k)})$.

We will now construct a problem instance with three output dimensions and an aggregation operation that is support expanding but fails the additional condition. Additionally this problem instance has no conic constraints making the binding constraint set the empty set for all vectors. Consider the aggregation operation $\boldsymbol{x}^{(1)} = [0, 1, 1], \boldsymbol{x}^{(2)} = [1, 0, 1], \boldsymbol{x}^{(3)} = [1, 1, 0] \to \boldsymbol{x}^{(A)} = [1, 1, 1]$. This aggregation operation fails the necessary condition for elicitability-expansion stated in Corollary 4.6. □

Similarly, binding set contraction relative to every input vector of the aggregation operation also does not guarantee that aggregation has power. The next proposition formalizes this.

Proposition A.2. *There exists an aggregation operation $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ and a set of conic constraints $\boldsymbol{C}$ such that $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements binding set contraction relative to i for $i \in \{1, 2\}$. However, $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ is not elicitability-expanding.*

Proof. Consider a problem with two output dimensions having the following two constraints: 1) $c_1 : x_1 - x_2 \leq 0$, 2) $c_2 : -2x_1 + x_2 \leq 0$. Consider an aggregation operation $\boldsymbol{x}^{(1)} = [1, 1], \boldsymbol{x}^{(2)} = [1, 2] \to \boldsymbol{x}^{(A)} = [5, 7]$, where the binding constraints sets are $\mathcal{V}_{\boldsymbol{x}^{(i)}} = \{c_i\}$ for $i \in \{1, 2\}$ and $\mathcal{V}_{\boldsymbol{x}^{(A)}} = \emptyset$ and the supports are the entire set of output dimensions. Hence the feasibility-improving and budget-reducing directions are $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}), \mathcal{V}(\boldsymbol{x}^{(A)})} = \{\boldsymbol{d} : \mathbb{1}^\top \boldsymbol{d} < 0\}, \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}), \mathcal{V}(\boldsymbol{x}^{(1)})} = \{\boldsymbol{d} : \boldsymbol{d}_1 - \boldsymbol{d}_2 \leq 0, \mathbb{1}^\top \boldsymbol{d} < 0\}, \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(2)}), \mathcal{V}(\boldsymbol{x}^{(2)})} = \{\boldsymbol{d} : -2\boldsymbol{d}_1 + \boldsymbol{d}_2 \leq 0, \mathbb{1}^\top \boldsymbol{d} < 0\}$.

First note that this operation is not feasibility-expanding since $\boldsymbol{x}^{(A)}$ satisfies the conic constraints. We will show that there is no $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}), \mathcal{V}(\boldsymbol{x}^{(A)})}$ such that $\boldsymbol{d} \not\leq 0$ and $\boldsymbol{d}$ is in neither $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}), \mathcal{V}(\boldsymbol{x}^{(1)})}$ nor in $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(2)}), \mathcal{V}(\boldsymbol{x}^{(2)})}$. This along with the lack of feasibility-expansion implies violation of the alternate power-characterizing condition in Definition 4.8 which implies that the aggregation operation cannot be elicitability-expanding by Theorem 4.4 and Proposition B.6.

Any $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}), \mathcal{V}(\boldsymbol{x}^{(A)})}$ such that $\boldsymbol{d} \notin \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}), \mathcal{V}(\boldsymbol{x}^{(1)})} \cup \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(2)}), \mathcal{V}(\boldsymbol{x}^{(2)})}$ satisfies $\boldsymbol{d}_1 - \boldsymbol{d}_2 > 0$ and $-2\boldsymbol{d}_1 + \boldsymbol{d}_2 > 0$. These inequalities satisfy $\boldsymbol{d}_2 < \boldsymbol{d}_1 < 0$. Hence such a $\boldsymbol{d}$ cannot satisfy $\boldsymbol{d} \not\leq 0$.

□

On the other hand, feasibility expansion on its own guarantees power under some feature mapping as stated by the following proposition.

Proposition A.3. *Fix conic constraints $\boldsymbol{C}$. If an aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements feasibility-expansion, then it is elicitability-expanding.*

Proof. From the definition of the power-characterizing condition (Definition 4.2), this condition is satisfied if the aggregation operation implements feasibility-expansion. Theorem 4.3 showing the sufficiency of the power-characterizing condition for elicitability-expansion implies that implementing feasibility-expansion is sufficient for elicitability-expansion. □

A.2 Proofs for Section 3.1

Recall that the examples in this section use the feature weights matrix $\boldsymbol{\alpha}(q) := \begin{bmatrix} 1 & 0 & q \\ 0 & 1 & q \end{bmatrix}$.

A.2.1 Analysis of Example 3.2

Proposition A.4. *For the feature weights matrix $\boldsymbol{\alpha}_2$ and constraint matrix with the row $x_3 \leq x_1 + x_2$ in Example 3.2, the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 1], \boldsymbol{x}^{(2)} = [0, 1, 1] \to \boldsymbol{x}^{(A)} = [0, 0, 1]$ is elicitability-expanding.*

Proof. From the construction, it is easy to see that $x^{(1)}$ can be elicited with a linear reward function $[1, 0, 0]$ equal to the F_1 and budget level $E = 1$ and $x^{(2)}$ can be elicited with a linear reward function $[0, 1, 0]$ equal to the F_1 and budget level $E = 1$.

Let us use our characterization Lemma 4.7 to formally demonstrate elicitability-expansion.

The support sets of $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$ are $\mathcal{S}(\boldsymbol{x}^{(1)}) = \{1, 3\}$ and $\mathcal{S}(\boldsymbol{x}^{(2)}) = \{2, 3\}$ respectively. The single conic constraint is binding for both $x^{(1)}, x^{(2)}$. The set $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}),\mathcal{V}(x^{(1)})}$ is $\{\boldsymbol{d} : \boldsymbol{d}_3 \leq \boldsymbol{d}_1 + \boldsymbol{d}_2, \boldsymbol{d}_2 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$. For any $\boldsymbol{d}$ in this set, $\boldsymbol{d}_3 < 0$ and $\boldsymbol{d}_1 + \boldsymbol{d}_2 < -\boldsymbol{d}_3$. The set $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(2)}),\mathcal{V}(x^{(2)})}$ is $\{\boldsymbol{d} : \boldsymbol{d}_3 \leq \boldsymbol{d}_1 + \boldsymbol{d}_2, \boldsymbol{d}_1 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$. For any $\boldsymbol{d}$ in this set, $\boldsymbol{d}_3 < 0$ and $\boldsymbol{d}_1 + \boldsymbol{d}_2 < -\boldsymbol{d}_3$.

Now consider the set of feature-improving direction $\{\boldsymbol{d} : \boldsymbol{d}_1 + 2\boldsymbol{d}_3 \geq 0, \boldsymbol{d}_2 + 2\boldsymbol{d}_3 \geq 0\}$. For any $\boldsymbol{d}$ in this set, $\boldsymbol{d}_1 + \boldsymbol{d}_2 \geq -4d_3$.

All three conditions $\boldsymbol{d}_3 < 0$, $\boldsymbol{d}_1 + \boldsymbol{d}_2 < -\boldsymbol{d}_3$, and $\boldsymbol{d}_1 + \boldsymbol{d}_2 \geq -4d_3$ cannot be satisfied since for $\boldsymbol{d}_3 < 0$, $-\boldsymbol{d}_3 < -4\boldsymbol{d}_3$. Hence there is no intersection between feasibility improving directions and features improving directions and $x^{(1)}$ is elicitable. Similarly, $x^{(2)}$ is also elicitable.

$\boldsymbol{x}^{(A)}$ is not feasible and hence not elicitable. This shows that $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding by implementing feasibility-expansion. □

A.2.2 Analysis of Example 3.4

Proposition A.5. *For the feature weights matrix $\boldsymbol{\alpha}(0.6)$ and null constraint matrix in Example 3.4, the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 0], \boldsymbol{x}^{(2)} = [0, 1, 0] \to \boldsymbol{x}^{(A)} = [1/2, 1/2, 0]$ is elicitability-expanding.*

Proof. The set of directions $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}),\mathcal{V}(\boldsymbol{x}^{(1)})} = \{\boldsymbol{d} : \boldsymbol{d}_2 \geq 0, \boldsymbol{d}_3 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$. And the set of feature-improving directions is $\mathcal{A} = \{\boldsymbol{d} : \boldsymbol{d}_1 + 0.6\boldsymbol{d}_3 \geq 0, \boldsymbol{d}_2 + 0.6\boldsymbol{d}_3 \geq 0\}$.

$\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}),\mathcal{V}(\boldsymbol{x}^{(1)})}$ means that $\boldsymbol{d}_1 < -(\boldsymbol{d}_2 + \boldsymbol{d}_3) < -\boldsymbol{d}_3$ and $\boldsymbol{d}_3 \geq 0$. $\boldsymbol{d} \in \mathcal{A}_1$ means that $\boldsymbol{d}_1 \geq -0.6\boldsymbol{d}_3$. These three conditions cannot be simultaneously showing that $\boldsymbol{x}^{(1)}$ is elicitable due to empty intersection of $\mathcal{A}$ and $\mathcal{B}_1$. Symmetrically, we can also show that $\boldsymbol{x}^{(2)}$ is also elicitable.

Now let us argue that $\boldsymbol{x}^{(A)} = [1/2, 1/2, 0]$ is not elicitable. $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})} = \{\boldsymbol{d} : \boldsymbol{d}_3 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$. Consider $\boldsymbol{d} = [-0.6, -0.6, 1]$. $\boldsymbol{d} \in \mathcal{A} \cap \mathcal{B}_A$. This shows that $\boldsymbol{x}^{(A)}$ is not elicitable.

□

A.2.3 Analysis of Example 3.6

Proposition A.6. *For the feature weights matrix $\boldsymbol{\alpha}(0.2)$ and conic constraint matrix with one constraint $x_1 + x_2 \leq x_3$ from Example 3.6, the aggregation operation $\boldsymbol{x}^{(1)} = [1, 0, 1], \boldsymbol{x}^{(2)} = [0, 1, 1] \to \boldsymbol{x}^{(A)} = [0, 0, 1]$ is elicitability-expanding.*

Proof of Proposition A.6. The feature-improving directions are the set $\mathcal{A} = \{\boldsymbol{d} : \boldsymbol{d}_1 + 0.2\boldsymbol{d}_3 \geq 0, \boldsymbol{d}_2 + 0.2\boldsymbol{d}_3 \geq 0\}$.

The constraint is binding at both $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$. The feasibility improving directions are $\mathcal{B}_{(\mathcal{S}(\boldsymbol{x}^{(1)}),\mathcal{V}(\boldsymbol{x}^{(1)})} = \{\boldsymbol{d} : \boldsymbol{d}_1 + \boldsymbol{d}_2 \leq \boldsymbol{d}_3, \boldsymbol{d}_2 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$.

If $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(1)}),\mathcal{V}(\boldsymbol{x}^{(1)})} \cap \mathcal{A}$, then $\boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0$, $\boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 \leq 2\boldsymbol{d}_3$, $\boldsymbol{d} \in \mathcal{A}$, $\boldsymbol{d}_1 \geq -0.2\boldsymbol{d}_3$, and $\boldsymbol{d}_2 \geq -0.2\boldsymbol{d}_3$. This implies that $\boldsymbol{d}_3 < 0$. If all the conditions are satisfied simultaneously, then $\boldsymbol{d}_1 > 0$ and $\boldsymbol{d}_2 > 0$. This contradicts $\boldsymbol{d}_1 + \boldsymbol{d}_2 \leq \boldsymbol{d}_3 < 0$.

The conic constraint is not binding at $\boldsymbol{x}^{(A)}$. Now consider the feasibility improving directions of $\boldsymbol{d}^{(A)}$: $\mathcal{B}_{(\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})} = \{\boldsymbol{d} : \boldsymbol{d}_1 \geq 0, \boldsymbol{d}_2 \geq 0, \boldsymbol{d}_1 + \boldsymbol{d}_2 + \boldsymbol{d}_3 < 0\}$. The vector $\boldsymbol{d} = (0.2, 0.2, -1) \in \mathcal{A} \cap \mathcal{B}_A$ demonstrating that $\boldsymbol{x}^{(A)}$ is not elicitable. □

A.3 Proofs in Section 3.2

A.3.1 Proof of Theorem 3.7

Proof of Theorem 3.7. We show this as a corollary of Theorem 4.4. We will prove this by showing that when both of the conditions in Theorem 3.7 are violated, the power-characterizing condition (Definition 4.2) is violated and hence $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ cannot be elicitability-expanding. The violation of the conditions in Theorem 3.7 correspond to lack of implementation of any of the natural mechanisms.

One way of satisfying the power-characterizing condition is through implementing feasibility-expansion. Violating the conditions in Theorem 3.7 means feasibility-expansion is not implemented. We will show that the other way of satisfying the power-characterizing condition also does not hold.

When the second condition of Theorem 3.7 theorem is violated, there exists $k \in [K]$ with respect to which $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is neither support-expanding nor binding-set contracting. That is, there is a k such that $\mathcal{V}(\boldsymbol{x}^{(k)}) \subseteq \mathcal{V}(\boldsymbol{x}^{(A)})$ and $\mathcal{S}(\boldsymbol{x}^{(k)}) \supseteq \mathcal{S}(\boldsymbol{x}^{(A)})$. As a result, the rows of $C_{\mathcal{V}(\boldsymbol{x}^{(k)})}$ are a subset of the rows in $C_{\mathcal{V}(\boldsymbol{x}^{(A)})}$ and similarly, the rows in $\boldsymbol{d}_{\mathcal{S}(\boldsymbol{x}^{(k)})^c}$ are a subset of the rows in $\boldsymbol{d}_{\mathcal{S}(\boldsymbol{x}^{(A)})^c}$. So every $\boldsymbol{d}$ in the feasibility-improving and budget-reducing directions set of $\boldsymbol{x}^{(A)}$ i.e., $\boldsymbol{d} \in \{C_{\mathcal{V}(\boldsymbol{x}^{(A)})}\boldsymbol{d} \leq 0, \boldsymbol{d}_{\mathcal{S}(\boldsymbol{x}^{(A)})^c} \geq 0, \mathbb{1}^\top \boldsymbol{d} = -1\}$ satisfies the inequalities $C_{\mathcal{V}(\boldsymbol{x}^{(k)})}(\boldsymbol{d}) \leq 0$ and $\boldsymbol{d}_{\mathcal{S}(\boldsymbol{x}^{(k)})^c} \geq 0$. Such a $\boldsymbol{d}$ also satisfies all weighted combinations of the inequalities. Hence for any $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}^{|\mathcal{V}_k|}_{\geq 0}$, $(\boldsymbol{\gamma}^{(k)\top})C_{\mathcal{V}(\boldsymbol{x}^{(k)})}\boldsymbol{d} - \|(\boldsymbol{\gamma}^{(k)\top}C_{\mathcal{V}(\boldsymbol{x}^{(k)})})_-\|_\infty \leq 0$. That is, the second condition of the power-characterizing condition is also violated. □

B Additional details and proofs for Section 4

B.1 Characterization of single output elicitation

A key technical tool for our characterization of elicitability through aggregation (provided by Lemma 4.7) is the characterization of direct elicitability of a vector $\boldsymbol{x}$ with a single agent, given a feature weights matrix $\boldsymbol{\alpha}$. This characterization is in terms of the intersection of feasible perturbation directions $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} = \{\boldsymbol{d} : C_{\mathcal{V}(\boldsymbol{x})}\boldsymbol{d} \leq 0\} \cap \{\boldsymbol{d} : \boldsymbol{d}_{\mathcal{S}(\boldsymbol{x})^c} \geq 0\} \cap \{\mathbb{1}^t\boldsymbol{d} < 0\}$ and feature-improving directions $d \in \mathbb{R}^M$: $\{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$. It is stated in the following proposition and it generalizes the characterization results in Kleinberg and Raghavan [2020] to allow for conic constraints $\boldsymbol{C}$.

Proposition B.1 (Single output elicitation characterization)**.** *Given a feature weights matrix $\boldsymbol{\alpha}$ and conic constraints $\boldsymbol{C}$, an output vector $\boldsymbol{x} \gneq 0$ is elicitable if and only if it is feasible i.e., $\boldsymbol{C}\boldsymbol{x} \leq 0$ is and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is empty.*

We will prove this proposition by proving the necessary and sufficient directions through Lemma B.3 and Lemma B.4 respectively.

To prove these lemmas we will make use of the monotonicity property we assume our reward functions satisfy. Recall that we assume that reward functions do not decrease if all features are weakly increased, and strictly increase if some feature is increased.

A consequence of this monotonicity of rewards is that the budget necessary to elicit a vector $\boldsymbol{x}$ is its ℓ_1 norm $\|\boldsymbol{x}\|_1$. This is shown in the following lemma.

Lemma B.2. *If a vector $\boldsymbol{x}$ is elicitable with budget E, then $\|\boldsymbol{x}\|_1 = E$.*

Proof. If $\boldsymbol{x}$ is elicitable, then it must be feasible. So $\|\boldsymbol{x}\|_1 \leq E$ due to the budget constraint and $\boldsymbol{C}\boldsymbol{x} \leq 0$. For any monotone reward function R, there is a feature F_j such that increasing the value of feature F_j strictly increases the reward R. Since we assume that $\boldsymbol{\alpha}$ has no zero rows (Assumption 2.2), there is a coordinate i in the j^{th} row of $\boldsymbol{\alpha}$ that is non-zero. Additionally, by Assumption 2.1, there is a vector $\boldsymbol{y} \geq 0$

with $\boldsymbol{y}_i > 0$ that satisfies $\boldsymbol{C}\boldsymbol{y} \leq 0$. Consider a scaled version $\boldsymbol{y}'$ of $\boldsymbol{y}$ with ℓ_1 norm less than $E - \|\boldsymbol{x}\|_1$. That is $\boldsymbol{y}' = (E - \|\boldsymbol{x}\|_1)\boldsymbol{y}/\|\boldsymbol{y}\|_1$. The vector $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{y}'$ has a strictly higher value of feature F_j and no lower values on other features. That is, $\boldsymbol{\alpha}\boldsymbol{x}' \geq \boldsymbol{\alpha}\boldsymbol{x}$ and $(\boldsymbol{\alpha}\boldsymbol{x}')_j > (\boldsymbol{\alpha}\boldsymbol{x})_j$. As a result, the reward function has a higher value on $\boldsymbol{x}'$ than on $\boldsymbol{x}$. $\boldsymbol{x}'$ satisfies the budget constraint since $\|\boldsymbol{x}\|_1 \leq \|\boldsymbol{x}\|_1 + \|\boldsymbol{y}'\|_1 \leq E$. It also satisfies conic constraints $\boldsymbol{C}$ since both $\boldsymbol{x}$ and $\boldsymbol{y}'$ satisfy $\boldsymbol{C}$. This shows that for any reward function, it is possible to construct another feasible vector with a higher reward if $\boldsymbol{x}$ has ℓ_1 norm less than the effort level E. Therefore, $\boldsymbol{x}$ is not elicitable if $\|\boldsymbol{x}\|_1 < E$. □

Lemma B.3 (Single output elicitation necessary)**.** *Given a feature weights matrix $\boldsymbol{\alpha}$ and conic constraints $\boldsymbol{C}$, an output vector $\boldsymbol{x} \gneq 0$ is elicitable only if $\boldsymbol{x}$ is feasible and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is empty.*

Proof of Lemma B.3. We will show that if $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is non-empty then $\boldsymbol{x}$ is not elicitable. If the intersection is non-empty, consider any $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$, we will use this $\boldsymbol{d}$ to construct a feasible output vector $\boldsymbol{y}$ with $\|\boldsymbol{y}\|_1 \leq \|\boldsymbol{x}\|_1$ and having strictly higher reward than $\boldsymbol{x}$ for every monotone reward function of the features. This vector $\boldsymbol{y}$ we construct is $\boldsymbol{y} = \|\boldsymbol{x}\|_1(\boldsymbol{x} + \lambda\boldsymbol{d})/\|\boldsymbol{x} + \lambda\boldsymbol{d}\|_1$ for an appropriate choice of λ that we will describe shortly.

First consider the vector $\boldsymbol{y}' = \boldsymbol{x} + \lambda\boldsymbol{d}$. Note that $\boldsymbol{y}'$ is feasible on all conic and non-negativity constraints that are binding at $\boldsymbol{x}$ due to $\boldsymbol{d}$'s membership in $\{\boldsymbol{d} : C_{\mathcal{V}(\boldsymbol{x})}\boldsymbol{d} \leq 0\} \cap \{\boldsymbol{d} : \boldsymbol{d}_{\mathcal{S}(\boldsymbol{x})^c} \geq 0\}$. We can choose λ to be small enough so that $\boldsymbol{y}'$ continues to meet all non-binding constraints. That is choose $\lambda < \min_{j \in \mathcal{V}(\boldsymbol{x})^c, \boldsymbol{C}_j\boldsymbol{d} > 0} -\boldsymbol{C}_j\boldsymbol{x}/\boldsymbol{C}_j\boldsymbol{d}$ and $\min_{i \in \mathcal{S}(\boldsymbol{x}), \boldsymbol{d}_i < 0} -\boldsymbol{x}_i/\boldsymbol{d}_i$. This establishes that we have a positive choice of λ making $\boldsymbol{y}'$ satisfy the nonnegativity and conic constraints. Additionally, we have that $\mathbb{1}^t\boldsymbol{y}' = \|\boldsymbol{y}'\|_1 = \|\boldsymbol{x}\|_1 + \lambda\mathbb{1}^t d < \|\boldsymbol{x}\|_1 = 1$. That is, $\boldsymbol{y}'$ satisfies any budget constraint that is satisfied by $\boldsymbol{x}$, and satisfies with a strictly larger margin. Hence $\boldsymbol{y}'$ is feasible relative to the conic constraints and budget constraints from any level of specified budget.

We also have that $\boldsymbol{\alpha}^t\boldsymbol{y}' = \boldsymbol{\alpha}^t(\boldsymbol{x} + \lambda\boldsymbol{d}) \geq \boldsymbol{\alpha}^t\boldsymbol{x}$ since $\boldsymbol{\alpha}^t\boldsymbol{d} \geq 0$. Hence $\boldsymbol{y}'$ satisfies feasibility constraints and has at least as high values on all features. By Lemma B.2, y' is not elicitable with budget $\|\boldsymbol{x}\|_1$ since $\|\boldsymbol{y}'\|_1 < \|\boldsymbol{x}\|_1$. This means that for any monotone reward function R, there is a feasible vector $\boldsymbol{x}'$ with $\|\boldsymbol{x}'\| \leq \|\boldsymbol{x}\|_1$ having $R(\boldsymbol{x}') > R(\boldsymbol{y}')$. Since $\boldsymbol{\alpha}\boldsymbol{y}' \geq \boldsymbol{\alpha}\boldsymbol{x}$, $R(\boldsymbol{y}') \geq R(\boldsymbol{x})$. In particular, $\boldsymbol{x}'$ has a strictly higher reward than $\boldsymbol{x}$. This means that $\boldsymbol{x}$ cannot be elicited with effort level $\|\boldsymbol{x}\|_1$ which is the only possible level at which $\boldsymbol{x}$ can be elicited by Lemma B.2. So, $\boldsymbol{x}$ cannot be elicited if $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{\alpha}\boldsymbol{d} \geq 0\} \neq \emptyset$. □

Lemma B.4 (Single output elicitation sufficient)**.** *Given feature weights matrix $\boldsymbol{\alpha}$ and conic constraints, $\boldsymbol{C}$, an output vector $\boldsymbol{x} \geq 0$ is elicitable if $\boldsymbol{x}$ satisfies the conic constraints i.e., $\boldsymbol{C}\boldsymbol{x} \leq 0$ and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is empty.*

Proof. Let us denote $S := \mathcal{S}(\boldsymbol{x})$ and $V := \mathcal{V}(\boldsymbol{x})$. And let $D_{\boldsymbol{\alpha}}$ be the set of feature-improving directions $\{\boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$. Under the condition that $\boldsymbol{x}$ is feasible and $\mathcal{B}_{S,V} \cap D_{\boldsymbol{\alpha}}$ is empty, we will show that $\boldsymbol{x}$ is elicitable by explicitly constructing a reward function that elicits $\boldsymbol{x}$ with budget $\|\boldsymbol{x}\|_1$.

Dual certificate of empty intersection. The condition of empty intersection of the sets $\mathcal{B}_{S,V}$ and $D_{\boldsymbol{\alpha}}$ can equivalently be written as the infeasibility of the system :

$$C_V d \leq 0, \qquad I_{S^c} d \geq 0, \qquad \alpha^\top d \geq 0, \qquad \mathbf{1}^\top d < 0, \tag{3}$$

where $I_{S^c} \in \mathbb{R}^{|S^c| \times M}$ is the subset of rows of the $M \times M$ identity matrix indexed by the set S^c.

By Motzkin's transposition theorem, infeasibility of (3) implies the existence of dual variables

$$\gamma \in \mathbb{R}^{|V|}_{\geq 0}, \quad \lambda \in \mathbb{R}^{|S^c|}_{\geq 0}, \quad \nu \in \mathbb{R}^{N}_{\geq 0}, \quad \tau > 0$$

such that

$$C_V^\top \gamma \; - \; I_{S^c}^\top \lambda \; + \; \tau\mathbf{1} \; - \; \boldsymbol{\alpha}\,\nu \; = \; 0 \tag{4}$$

Reward function construction. We will now define a reward function that is linear in the features to elicit a vector $\boldsymbol{x}$. We will later show how this reward function along with budget $\|\boldsymbol{x}\|_1$ elicits the vector $\boldsymbol{x}$ when the empty intersection condition is satisfied. The reward function is

$$R(\boldsymbol{u}) \;=\; \sum_{i=1}^{N} \beta_i f_i((\boldsymbol{\alpha}^\top \boldsymbol{u})_i) \qquad \text{with} \qquad \beta_i \;:=\; \frac{\nu_i}{f_i'\big((\boldsymbol{\alpha}^\top \boldsymbol{x})_i\big)},$$

which is well-defined since each f_i is strictly increasing, hence $f_i'((\boldsymbol{\alpha}^\top \boldsymbol{x})_i) > 0$. Because each f_i is concave and increasing, R is concave. Its gradient at x is

$$\nabla R(\boldsymbol{x}) \;=\; \sum_{i=1}^{N} \beta_i f_i'\big((\boldsymbol{\alpha}^\top \boldsymbol{x})_i\big)\, \alpha_{\cdot,i} \;=\; \alpha\, \nu,$$

where $\alpha_{\cdot,i}$ is the i-th column of α.

Elicitability. From Lemma B.2 we know that a vector $\boldsymbol{x}$ can only be elicited with a budget of $E = \|\boldsymbol{x}\|_1$. So let us consider the constrained reward maximization program with this budget E and a reward function R. We will show that when $\mathcal{B}_{S,V}$ and $D_{\boldsymbol{\alpha}}$ have empty intersection, the dual certificate of this empty intersection that we computed before along with $\boldsymbol{x}$ satisfies the KKT conditions of the reward-maximizing optimization program.

$$\max_{\boldsymbol{u} \in \mathbb{R}^M} \; R(\boldsymbol{f}(\boldsymbol{\alpha}^\top \boldsymbol{u})) \quad \text{s.t.} \quad \boldsymbol{C}\boldsymbol{u} \le 0, \;\; \boldsymbol{u} \ge 0, \;\; \mathbf{1}^\top \boldsymbol{u} \le E.$$

This is a concave program, and its Lagrangian is

$$\mathcal{L}(\boldsymbol{u}, \lambda_0, \boldsymbol{\mu}, \tilde{\boldsymbol{\gamma}}) \;=\; R(\boldsymbol{f}(\boldsymbol{\alpha}^\top \boldsymbol{u})) \;+\; \lambda_0 \,(E - \mathbf{1}^\top \boldsymbol{u}) \;+\; \boldsymbol{\mu}^\top \boldsymbol{u} \;-\; \tilde{\boldsymbol{\gamma}}^\top (\boldsymbol{C}\boldsymbol{u}),$$

with dual variables $\lambda_0 \ge 0$, $\boldsymbol{\mu} \ge 0$, $\tilde{\boldsymbol{\gamma}} \ge 0$. Evaluate the KKT conditions at $\boldsymbol{u} = \boldsymbol{x}$ with the choice of dual variables from the dual certificate of the empty intersection condition. That is, the dual variables are

$$\lambda_0 := \tau, \qquad \mu_S := 0, \;\; \mu_{S^c} := \lambda, \qquad \tilde{\gamma}_V := \gamma, \;\; \tilde{\gamma}_{V^c} := 0.$$

Primal feasibility holds due to feasibility of $\boldsymbol{x}$ by definition of S, V. Complementary slackness holds since $x_j = 0$ for $j \in S^c$ and $(Cx)_\ell = 0$ for $\ell \in V$, while $\mu_S = 0$. For stationarity,

$$\nabla R(x) \;-\; \lambda_0 \mathbf{1} \;+\; \mu \;-\; C^\top \tilde{\gamma} \;=\; \alpha\nu \;-\; \tau \mathbf{1} \;+\; I_{S^c}^\top \lambda \;-\; C_V^\top \gamma \;=\; 0$$

by (4).

Since R is concave and the constraints are linear, the KKT conditions are sufficient to certify optimality; hence x maximizes R over the feasible region and is therefore elicitable. □

Proof of Proposition B.1. This follows from Lemma B.3, Lemma B.4 showing necessity and sufficiency of the condition of non-emptiness of $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})} \cap \{\boldsymbol{\alpha}\boldsymbol{d} \ge 0\}$ for elicitability of $\boldsymbol{x}$. □

An immediate consequence of the condition characterizing elicitability of a single output vector from Proposition B.1 is that the role of the budget is only in scaling the ℓ_1 norm of elicitable output vectors.

Corollary B.5. *A vector $\boldsymbol{x}$ is elicitable with some budget E if and only $\boldsymbol{x}/\|\boldsymbol{x}\|_1$ is elicitable with budget* 1 *for the same reward function.*

Proof. The condition characterizing the elicitability of $\boldsymbol{x}$ given a feature weights matrix $\boldsymbol{\alpha}$ is whether or not the sets $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$ and $\{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \ge 0\}$ intersect (Lemma 4.7). Note that the sets $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}),\mathcal{V}(\boldsymbol{x})}$ and $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}/\|\boldsymbol{x}\|_1),\mathcal{V}(\boldsymbol{x}/\|\boldsymbol{x}\|_1)}$ are equal because the supports and binding sets of $\boldsymbol{x}$ and $\boldsymbol{x}/\|\boldsymbol{x}\|_1$ are the same. As a result, $\boldsymbol{x}$ is elicitable if and only if $\boldsymbol{x}/\|\boldsymbol{x}\|_1$ is elicitable. From Lemma B.2, $\boldsymbol{x}$, if elicitable, is elicitable with budget $\|\boldsymbol{x}\|_1$ and similarly, $\boldsymbol{x}/\|\boldsymbol{x}\|_1$, if elicitable, is elicitable with budget 1. □

B.2 Proof of Lemma 4.7

Proof of Lemma 4.7. This follows directly from the characterization of elicitability of a single output vector $\boldsymbol{x}$ for a given feature weights matrix $\boldsymbol{\alpha}$ provided by Proposition B.1. The condition characterizing whether an aggregation operation $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is elicitability-expanding given a feature weights matrix $\boldsymbol{\alpha}$ is the condition that each $\boldsymbol{x}^{(k)}$ is elicitable under $\boldsymbol{\alpha}$ and $\boldsymbol{x}^{(A)}$ is not elicitable. Each of these conditions are provided by Proposition B.1. The condition for $\boldsymbol{x}^{(k)}$ to be elicitable under $\boldsymbol{x}^{(k)}$ to be elicitable is that $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$ and $\{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ have empty intersection and the condition for $\boldsymbol{x}^{(A)}$ to not be elicitable is that $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ and $\{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ have non-empty intersection. □

B.3 Proof of Corollary 4.6

Proof of Corollary 4.6. The assumptions are (i) no conic constraint is binding for these vectors, and (ii) $\boldsymbol{x}^{(A)}$ is not full support (i.e., $\mathcal{S}(\boldsymbol{x}^{(A)}) \neq [M]$) or there exists $j \in [M]$ such that no $\boldsymbol{x}^{(k)}$ has support $[M] \setminus \{j\}$.

By Theorem 4.4 and Theorem 4.3, the aggregation operation is elicitability-expanding if and only if the power-characterizing condition is satisfied. Because of assumption (i), $\mathcal{V}(\boldsymbol{x}^{(k)}) = \emptyset$ for every $k \in [K]$, so $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}(\boldsymbol{x}^{(k)})|}$ is trivially zero in the second case of Definition 4.2. The condition therefore reduces to either feasibility expansion or a strengthened form of support expansion (involving only $\boldsymbol{\lambda}^{(k)}$). It suffices to show that under assumptions (i) and (ii), strengthened support expansion is equivalent to support expansion.

We know that (ii) coupled with support expansion implies that either

$$\mathcal{S}(\boldsymbol{x}^{(A)}) \not\subseteq \mathcal{S}(\boldsymbol{x}^{(k)}) \text{ for all } k \in [K] \quad \text{and} \quad \mathcal{S}(\boldsymbol{x}^{(A)}) \neq [M],$$

or

$$\mathcal{S}(\boldsymbol{x}^{(A)}) = [M] \text{ and } \mathcal{S}(\boldsymbol{x}^{(k)}) \neq [M] \forall k \in [K] \text{ and there exists } j \in [M] \text{ s.t. } \mathcal{S}(\boldsymbol{x}^{(k)}) \neq [M] \setminus \{j\} \forall k \in [K].$$

A useful equivalent form of (ii) coupled with support expansion. We first show that (ii) coupled with support expansion is equivalent to the existence of indices $j(k) \in \mathcal{S}(\boldsymbol{x}^{(A)}) \setminus \mathcal{S}(\boldsymbol{x}^{(k)})$ for each $k \in [K]$ such that the set

$$J \coloneqq \bigcup_{k \in [K]} \{j(k)\}$$

is a *strict* subset of $[M]$. Indeed, for any selection of $j(k)$ we always have $J \subseteq \mathcal{S}(\boldsymbol{x}^{(A)})$. If $\mathcal{S}(\boldsymbol{x}^{(A)}) \neq [M]$, then necessarily $J \subsetneq [M]$. Otherwise, $\mathcal{S}(\boldsymbol{x}^{(A)}) = [M]$, and the existence of some j with $[M] \setminus \{j\} \not\subseteq \mathcal{S}(\boldsymbol{x}^{(k)})$ for all k is equivalent to being able to choose all $j(k)$ from $[M] \setminus \{j\}$, which forces $J \subseteq [M] \setminus \{j\} \subsetneq [M]$.

(ii) coupled with support expansion ⇒ Strengthened support expansion. Assume the above condition holds: there exists $J \subsetneq [M]$ and indices $j(k) \in \mathcal{S}(\boldsymbol{x}^{(A)}) \setminus \mathcal{S}(\boldsymbol{x}^{(k)})$ with $J = \bigcup_{k \in [K]} \{j(k)\}$. Fix any $c > 0$ and define $\boldsymbol{d} \in \mathbb{R}^M$ coordinate-wise by

$$\boldsymbol{d}_j \coloneqq \begin{cases} -1, & j \in J, \\ c, & j \notin J, \end{cases}$$

where c is a constant chosen such that

$$\frac{|J| - 1}{M - |J|} < c < \frac{|J|}{M - |J|}.$$

Since $J \subseteq \mathcal{S}(\boldsymbol{x}^{(A)})$, setting $\boldsymbol{d}_j < 0$ on J does not violate feasibility for coordinates outside the support, and we have $\boldsymbol{d}_{\mathcal{S}(\boldsymbol{x}^{(A)})^c} \geq 0$. Moreover,

$$\mathbb{1}^\top \boldsymbol{d} = -|J| + (M - |J|)c < 0.$$

Thus $\boldsymbol{d} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\emptyset}$. Finally, for each $k \in [K]$ we have $\boldsymbol{d}_{j(k)} = -1$ and therefore

$$-|\mathbb{1}^\top \boldsymbol{d}| = -|J| + (M - |J|)c > -1 = \boldsymbol{d}_{j(k)},$$

so strengthened support expansion holds.

Strengthened support expansion ⇒ support expansion. This follows by the same argument as Proposition 4.5. □

B.4 Proof of equivalence between the versions of the power-characterizing condition defined in Definition 4.2 and Definition 4.8

To prove Theorem 4.4 and Theorem 4.3, we will use an alternate but equivalent way of expressing the power-characterizing condition defined in Definition 4.2. This equivalent condition is defined in Definition 4.8. The equivalence between both conditions is stated in the following proposition.

Proposition B.6. *The conditions defined in Definition 4.2 and Definition 4.8 are equivalent.*

Proof. For ease of notation, let $V_k := \mathcal{V}(\boldsymbol{x}^{(k)})$, $S_k := \mathcal{S}(\boldsymbol{x}^{(k)})$ for $k \in [K]$, and let $V_A := \mathcal{V}(\boldsymbol{x}^{(A)})$, $S_A := \mathcal{S}(\boldsymbol{x}^{(A)})$. Both definitions agree in the feasibility-expansion case, so it suffices to compare their second cases. We will show the following per-k equivalence: for any fixed $\boldsymbol{d} \in \mathcal{B}_{S_A,V_A}$ with $\boldsymbol{d} \not\leq 0$ and any $k \in [K]$,

(1) $\{\boldsymbol{u} + \lambda \boldsymbol{d} : \boldsymbol{u} \in \mathbb{R}^M_{\geq 0},\ \lambda \geq 0\} \cap \mathcal{B}_{S_k,V_k} = \varnothing$, if and only if

(2) there exist $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}^{|V_k|}_{\geq 0}$ and $\boldsymbol{\lambda}^{(k)} \in \mathbb{R}^{|S_k^c|}_{\geq 0}$ such that, writing $\boldsymbol{w}^{(k)} := \big(\boldsymbol{\gamma}^{(k)}\big)^\top \boldsymbol{C}_{V_k} - \big(\boldsymbol{\lambda}^{(k)}\big)^\top I_{S_k^c}$,

$$\boldsymbol{w}^{(k)}\boldsymbol{d} \ + \ \big|\mathbb{1}^\top \boldsymbol{d}\big| \cdot \min_{j \in [M]} \min\big(\boldsymbol{w}^{(k)}_j,\, 0\big) \ > \ 0.$$

The per-k equivalence implies equivalence of the two definitions: the second case of Definition 4.8 requires (1) to hold for every $k \in [K]$ for a single $\boldsymbol{d}$, and the second case of Definition 4.2 requires (2) to hold for every $k \in [K]$ for a single $\boldsymbol{d}$.

Reformulating empty intersection as feasibility of an LP. The intersection in (1) is non-empty if and only if there exist $\boldsymbol{u} \in \mathbb{R}^M_{\geq 0}$ and $\lambda \geq 0$ such that

$$\boldsymbol{C}_{V_k}(\boldsymbol{u} + \lambda \boldsymbol{d}) \leq \boldsymbol{0}, \qquad -I_{S_k^c}(\boldsymbol{u} + \lambda \boldsymbol{d}) \leq \boldsymbol{0}, \qquad \mathbb{1}^\top(\boldsymbol{u} + \lambda \boldsymbol{d}) < 0.$$

Both (1) and the inequality in (2) are positively homogeneous in $\boldsymbol{d}$: scaling $\boldsymbol{d} \mapsto t\boldsymbol{d}$ for $t > 0$ rescales λ in (1) and rescales both terms equally in (2). We may therefore normalize $\boldsymbol{d}$ so that $\mathbb{1}^\top \boldsymbol{d} = -1$; under this normalization, $|\mathbb{1}^\top \boldsymbol{d}| = 1$. Note that λ must be strictly positive: $\boldsymbol{u} \geq \boldsymbol{0}$ implies $\mathbb{1}^\top \boldsymbol{u} \geq 0$, while $\mathbb{1}^\top(\boldsymbol{u} + \lambda \boldsymbol{d}) < 0$ together with $\mathbb{1}^\top \boldsymbol{d} = -1$ forces $\lambda > 0$. Setting $\boldsymbol{v} := \boldsymbol{u}/\lambda \in \mathbb{R}^M_{\geq 0}$ and dividing the inequalities by λ, the non-empty intersection is equivalent to the existence of $\boldsymbol{v} \in \mathbb{R}^M_{\geq 0}$ with $\mathbb{1}^\top \boldsymbol{v} < 1$ such that

$$\boldsymbol{C}_{V_k}(\boldsymbol{d} + \boldsymbol{v}) \leq \boldsymbol{0} \qquad \text{and} \qquad -I_{S_k^c}(\boldsymbol{d} + \boldsymbol{v}) \leq \boldsymbol{0}.$$

Dualizing the inequality system. A finite system of linear inequalities $A\boldsymbol{y} \leq \boldsymbol{0}$ holds if and only if every non-negative weighted combination of its rows holds, i.e., $\boldsymbol{\gamma}^\top A\boldsymbol{y} \leq 0$ for every $\boldsymbol{\gamma} \geq \boldsymbol{0}$. Applying this to the system above, the non-empty intersection is equivalent to the existence of $\boldsymbol{v} \in \mathbb{R}^M_{\geq 0}$ with $\mathbb{1}^\top \boldsymbol{v} \leq 1$ such that

$$\big(\boldsymbol{\gamma}^{(k)\top} \boldsymbol{C}_{V_k} - \boldsymbol{\lambda}^{(k)\top} I_{S_k^c}\big)(\boldsymbol{d} + \boldsymbol{v}) \ \leq \ 0 \qquad \forall \boldsymbol{\gamma}^{(k)} \in \mathbb{R}^{|V_k|}_{\geq 0},\ \boldsymbol{\lambda}^{(k)} \in \mathbb{R}^{|S_k^c|}_{\geq 0}.$$

We have replaced the strict inequality $\mathbb{1}^\top \boldsymbol{v} < 1$ with the closed inequality ≤ 1; this is without loss of generality because the value of the objective is continuous in $\boldsymbol{v}$ on a closed feasible set. We may also restrict

$\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)}$ to bounded norm $\|\boldsymbol{\gamma}^{(k)}\|_1 \leq 1$, $\|\boldsymbol{\lambda}^{(k)}\|_1 \leq 1$ without loss of generality, since the objective is positively homogeneous in $(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)})$. So the non-empty intersection is equivalent to

$$\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^M_{\geq 0} \\ \mathbb{1}^\top \boldsymbol{v}\leq 1}} \sup_{\substack{\boldsymbol{\gamma}^{(k)}\in\mathbb{R}^{|V_k|}_{\geq 0},\ \|\boldsymbol{\gamma}^{(k)}\|_1\leq 1 \\ \boldsymbol{\lambda}^{(k)}\in\mathbb{R}^{|S_k^c|}_{\geq 0},\ \|\boldsymbol{\lambda}^{(k)}\|_1\leq 1}} \boldsymbol{w}^{(k)}(\boldsymbol{d}+\boldsymbol{v}) \ \leq\ 0,$$

where $\boldsymbol{w}^{(k)} := \boldsymbol{\gamma}^{(k)\top} \boldsymbol{C}_{V_k} - \boldsymbol{\lambda}^{(k)\top} I_{S_k^c}$.

Applying the minimax theorem. The objective $\boldsymbol{w}^{(k)}(\boldsymbol{d}+\boldsymbol{v})$ is bilinear in $(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)})$ and $\boldsymbol{v}$, and both feasible sets are convex and compact. Sion's minimax theorem applies, so we may swap the order of inf and sup:

$$\sup_{\substack{\boldsymbol{\gamma}^{(k)}\in\mathbb{R}^{|V_k|}_{\geq 0},\ \|\boldsymbol{\gamma}^{(k)}\|_1\leq 1 \\ \boldsymbol{\lambda}^{(k)}\in\mathbb{R}^{|S_k^c|}_{\geq 0},\ \|\boldsymbol{\lambda}^{(k)}\|_1\leq 1}} \inf_{\substack{\boldsymbol{v}\in\mathbb{R}^M_{\geq 0} \\ \mathbb{1}^\top \boldsymbol{v}\leq 1}} \boldsymbol{w}^{(k)}(\boldsymbol{d}+\boldsymbol{v}) \ \leq\ 0.$$

Closed form of the inner infimum. For fixed $\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)}$ (and hence fixed $\boldsymbol{w}^{(k)}$), we compute

$$\inf_{\boldsymbol{v}\in\mathbb{R}^M_{\geq 0},\ \mathbb{1}^\top \boldsymbol{v}\leq 1} \boldsymbol{w}^{(k)}\boldsymbol{v}.$$

If $\boldsymbol{w}^{(k)}$ has any strictly negative coordinate, the infimum is $\min_j \boldsymbol{w}_j^{(k)}$, achieved by placing all of $\boldsymbol{v}$'s mass on the most negative coordinate of $\boldsymbol{w}^{(k)}$. If $\boldsymbol{w}^{(k)} \geq \mathbf{0}$, the infimum is 0, achieved at $\boldsymbol{v} = \mathbf{0}$. In both cases, $\inf_{\boldsymbol{v}} \boldsymbol{w}^{(k)}\boldsymbol{v} = \min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0)$, so

$$\inf_{\substack{\boldsymbol{v}\in\mathbb{R}^M_{\geq 0} \\ \mathbb{1}^\top \boldsymbol{v}\leq 1}} \boldsymbol{w}^{(k)}(\boldsymbol{d}+\boldsymbol{v}) \ =\ \boldsymbol{w}^{(k)}\boldsymbol{d} + \min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0).$$

Identifying the certificate. Combining, the non-empty intersection is equivalent to

$$\sup_{\substack{\boldsymbol{\gamma}^{(k)}\in\mathbb{R}^{|V_k|}_{\geq 0},\ \|\boldsymbol{\gamma}^{(k)}\|_1\leq 1 \\ \boldsymbol{\lambda}^{(k)}\in\mathbb{R}^{|S_k^c|}_{\geq 0},\ \|\boldsymbol{\lambda}^{(k)}\|_1\leq 1}} \Big(\boldsymbol{w}^{(k)}\boldsymbol{d} + \min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0)\Big) \ \leq\ 0,$$

i.e., $\boldsymbol{w}^{(k)}\boldsymbol{d} + \min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0) \leq 0$ for every bounded-norm $\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)} \geq \mathbf{0}$. Negating and using positive homogeneity in $(\boldsymbol{\gamma}^{(k)}, \boldsymbol{\lambda}^{(k)})$ to drop the bounded-norm restriction, the empty intersection (1) (under the normalization $\mathbb{1}^\top \boldsymbol{d} = -1$) is equivalent to: there exist $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}^{|V_k|}_{\geq 0}$ and $\boldsymbol{\lambda}^{(k)} \in \mathbb{R}^{|S_k^c|}_{\geq 0}$ such that $\boldsymbol{w}^{(k)}\boldsymbol{d} + \min_{j\in[M]} \min(\boldsymbol{w}_j^{(k)}, 0) > 0$. Undoing the normalization (using positive homogeneity in $\boldsymbol{d}$ to restore the factor $|\mathbb{1}^\top \boldsymbol{d}|$), (1) is equivalent to: there exist $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}^{|V_k|}_{\geq 0}$ and $\boldsymbol{\lambda}^{(k)} \in \mathbb{R}^{|S_k^c|}_{\geq 0}$ such that

$$\boldsymbol{w}^{(k)}\boldsymbol{d} \ +\ \big|\mathbb{1}^\top \boldsymbol{d}\big| \cdot \min_{j\in[M]} \min\big(\boldsymbol{w}_j^{(k)}, 0\big) \ >\ 0,$$

which is exactly (2). □

B.5 Proof of Theorem 4.4

Using the equivalence of the power-characterizing condition in Definition 4.2 and the alternative power-characterizing condition in Definition 4.8 that we established in Proposition B.6, we will show the necessity of the power-characterizing condition for elicitability-expansion by showing the necessity of the alternate condition. In the proof of Theorem 4.4, we will use the single-agent characterization of elicitability (Lemma B.4) rather than the multi-agent characterization of elicitability-expansion (Lemma 4.7).

Proof of Theorem 4.4. We will prove the contrapositive. Suppose that the power-characterizing condition is not satisfied. By Proposition B.6, this means that the alternate power-characterizing condition (Definition 4.8) is also not satisfied. Violation of the alternate power-characterizing condition means that $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is not feasibility-expanding. That is, $\boldsymbol{x}^{(A)}$ and each $\boldsymbol{x}^{(k)}$ for $k \in [K]$ is feasible.

We will show that $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ is not elicitability-expanding by showing that for any feature-weights matrix $\boldsymbol{\alpha}$ that makes $\boldsymbol{x}^{(A)}$ not elicitable, there exists some $k \in [K]$ such that $\boldsymbol{x}^{(k)}$ is also not elicitable under $\boldsymbol{\alpha}$.

For any feature weights matrix $\boldsymbol{\alpha}$ that makes $\boldsymbol{x}^{(A)}$ inelicitable, by Lemma B.4, there is a $\boldsymbol{d}^{(A)} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(A)}),\mathcal{V}(\boldsymbol{x}^{(A)})}$ such that $\boldsymbol{\alpha}\boldsymbol{d}^{(A)} \geq 0$. This $\boldsymbol{d}$ must have some positive entry i.e., $\boldsymbol{d}^{(A)} \not\leq 0$. Since $\mathbb{1}^\top \boldsymbol{d}^{(A)} < 0$, $\boldsymbol{d}^{(A)}$ must have some strictly negative coordinate.

Since the alternate power-characterizing condition is violated, there exists $\boldsymbol{x}^{(k)}$ with $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$ having non-empty intersection with $\{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)}\}$. It suffices to show that $\boldsymbol{x}^{(k)}$ is not elicitable under feature mapping $\boldsymbol{\alpha}$. To see this, let $\boldsymbol{d}^{(k)}$ denote an element of the intersection $\mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})} \cap \{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)}\}$. We can then write $\boldsymbol{d}^{(k)} = \boldsymbol{u} + \lambda \boldsymbol{d}^{(A)}$. Note that $\boldsymbol{\alpha}\boldsymbol{d}_i = \boldsymbol{\alpha}\boldsymbol{u} + \lambda \boldsymbol{\alpha}\boldsymbol{d}^{(A)}$. We know that $\boldsymbol{\alpha}\boldsymbol{u} \geq 0$ since $\boldsymbol{u} \geq 0$ and $\boldsymbol{\alpha}$ has non-negative entries. Additionally, $\boldsymbol{\alpha}\boldsymbol{d}^{(A)} \geq 0$. Hence $\boldsymbol{\alpha}\boldsymbol{d}^{(k)} \geq 0$ for $\boldsymbol{d}^{(k)} \in \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$. By Lemma B.3, this means that $\boldsymbol{x}^{(k)}$ is not elicitable.

□

B.6 Proof of Theorem 4.3

Proof of Theorem 4.3. Suppose the power-characterizing condition is satisfied. By Proposition B.6, this means that the alternate power-characterizing condition is also satisfied. Then we know that we are in one of two cases.

Case 1: $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)}$ implements feasibility expansion. Consider a feature mapping with a single feature and all dimensions contribute equal weights of one to this feature. All output vectors with the same ℓ_1 norm result in the same reward for all reward functions, and thus all feasible outcomes are elicitable. That is any output vector is elicitable if and only if it is feasible. A feasible output is elicitable with budget equal to its ℓ_1 norm. Under this construction, feasibility-expansion implies elicitability-expansion.

Case 2: there exists $\boldsymbol{d}^{(A)} \in \mathcal{B}_{S(\boldsymbol{x}^{(A)},\mathcal{V}(\boldsymbol{x}^{(A)})}$ with $\boldsymbol{d}^{(A)} \not\leq 0$ such that for all $\boldsymbol{u} \geq 0, \lambda \geq 0$, $\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)} \notin \mathcal{B}_{\mathcal{S}(\boldsymbol{x}^{(k)}),\mathcal{V}(\boldsymbol{x}^{(k)})}$ for $k \in [K]$. We will construct a feature mapping $\boldsymbol{\alpha}$ based on $\boldsymbol{d}^{(A)}$ such that the set of directions weakly increasing feature values i.e., the set $D_{\boldsymbol{\alpha}} = \{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is a subset of $\{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)} : \boldsymbol{u} \geq 0, \lambda \geq 0\}$. By Lemma 4.7, this implies that $\boldsymbol{x}^{(A)}$ is not elicitable but for all other outputs $\boldsymbol{x}^{(k)}$, $D_{\boldsymbol{\alpha}} \cap \mathcal{B}_{S(\boldsymbol{x}^{(k)},\mathcal{V}(\boldsymbol{x}^{(k)})}$ is empty and hence each $\boldsymbol{x}^{(k)}$ is elicitable under $\boldsymbol{\alpha}$.

To complete this argument, we will explicitly construct such an $\boldsymbol{\alpha}$ based on $\boldsymbol{d}^{(A)}$. Let $P_0 = \{i \in [m] : \boldsymbol{d}_i^{(A)} > 0\}$ denote the positive coordinates of $\boldsymbol{d}^{(A)}$ and let $N_0 = \{i \in [m] : \boldsymbol{d}_i^{(A)} \leq 0\}$ denote the negative or zero coordinates. Note that P_0 is non-empty since $\boldsymbol{d}^{(A)} \not\leq 0$. We construct two sets of features:

- For every $p \in P_0$, there is a corresponding feature F_p whose row in $\boldsymbol{\alpha}$ is the vector e_p which is the vector with 1 at coordinate p and zero everywhere else. That is, dimension p has weight 1 on feature F_p and all other dimensions have zero weight.
- The next set of features are defined for every pair $p \in P_0$, $q \in N_0$. This feature $F_{p,q}$ has a corresponding row in $\boldsymbol{\alpha}$ that is the vector $\boldsymbol{d}_p^{(A)} e_q - \boldsymbol{d}_q^{(A)} e_p$. That is, the only dimensions with possible non-zero weights to $F_{p,q}$ are dimensions p, q. The weight from dimension p is $|\boldsymbol{d}_q^{(A)}|$ and the weight from dimension q is $|\boldsymbol{d}_p^{(A)}|$.

Now let us show that the set $D_{\boldsymbol{\alpha}} = \{\boldsymbol{d} : \boldsymbol{\alpha}\boldsymbol{d} \geq 0\}$ is a subset of $B_A = \{\boldsymbol{u} + \lambda \boldsymbol{d}^{(A)}\}$. Take any $\boldsymbol{d} \in D_{\boldsymbol{\alpha}}$.

For every $p \in P_0$, since $\boldsymbol{d}$ weakly improves value of F_p, it holds that $\boldsymbol{d}_p \geq 0$. By ensuring that $\lambda \leq \boldsymbol{d}_p / \boldsymbol{d}_p^{(A)}$ for all $p \in P_0$, we can ensure that $\boldsymbol{d}_p - \lambda \boldsymbol{d}_p^{(A)} \geq 0$.

For every $p \in P_0, q \in N_0$, since $\boldsymbol{d}$ weakly improves value of $F_{p,q}$, it holds that $-\boldsymbol{d}_p \boldsymbol{d}_q^{(A)} + \boldsymbol{d}_q \boldsymbol{d}_p^{(A)} \geq 0$. In other words, $\boldsymbol{d}_q \geq \boldsymbol{d}_p \boldsymbol{d}_q^{(A)} / \boldsymbol{d}_p^{(A)}$. By choosing λ less than $\boldsymbol{d}_p^{(A)} \boldsymbol{d}_q / \boldsymbol{d}_p \boldsymbol{d}_q^{(A)}$ for every $q \in N_0$, we get $\boldsymbol{d}_q - \lambda \boldsymbol{d}_q^{(A)} \geq 0$ for every $q \in N_0$.

□

B.7 How addition and intersection aggregation rules can or cannot implement the mechanisms

In Section 3.1, we showed examples of intersection and addition aggregation rules expanding elicitability by implementing each of the feasibility-expansion, support expansion or binding set contraction mechanisms. In this part, we will discuss the mechanisms that each aggregation rule can or cannot implement. These results are summarized in Table 2.

	Feasibility Expansion	**Support Expansion**	**Binding Set Contraction**
Intersection aggregation	✓ (Example 3.2)	× (Proposition B.7)	✓ (Example 3.6)
Addition aggregation	× (Proposition B.8)	✓ (Example 3.4)	✓ (Example B.9)

Table 2: Implementability of mechanisms in Section 3.1 for the intersection aggregation rule and addition aggregation rule. The symbol ✓ denotes that there exists a problem instance where the aggregation rule implements that mechanism. The symbol × denotes that the aggregation rule does not implement the mechanism for any problem instance.

Intersection aggregation does not implement support expansion for any problem instance, as the following result formalizes.

Proposition B.7 (Intersection does not expand support)**.** *Consider any aggregation operation of the form* $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)} = \mathcal{A}_{intersect}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)})$*. For any* $k \in [K]$*, this aggregation operation does not implement support expansion relative to* k*.*

Proof. Proposition B.7 follows from the fact that the support of $\mathcal{A}_{\text{intersect}}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)})$ is always a subset of the support of each $\boldsymbol{x}^{(k)}$ for $k \in [K]$. □

Intersection aggregation can implement feasibility-expansion as shown in Example 3.2 and binding set-contraction as shown in Example 3.6. In fact, these examples go one step further and demonstrate that elicitability expansion is achievable via these mechanisms.

Addition aggregation does not implement feasibility expansion for any problem instance, as the following result formalizes.

Proposition B.8 (Addition cannot expand feasibility)**.** *Consider constraints* $\boldsymbol{C}$*. Any aggregation operation of the form* $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)} \to \boldsymbol{x}^{(A)} = \mathcal{A}_{add}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}; \boldsymbol{w})$ *does not implement feasibility expansion relative to* $\boldsymbol{C}$*.*

Proof. Proposition B.8 directly follows from the fact that the constraint set $\boldsymbol{C}$ is conic. □

On the other hand, addition aggregation operations can implement the other two mechanisms. Example 3.4 already constructed a problem instance where addition aggregation implements support expansion. The next example constructs a problem instance where addition aggregation can implement binding set contraction (Definition 3.5) and achieve elicitability-expansion for some feature mapping.

Example B.9 (Addition can result in binding set contraction)**.** *Consider the constraint matrix*

$$C = \begin{pmatrix} 1 & -1 & 0 \\ 1 & -\frac{1}{4} & -1 \end{pmatrix},$$

and consider vectors $\boldsymbol{x}^{(1)} = (1, 1, 2)$ *and* $\boldsymbol{x}^{(2)} = (2, 4, 1)$*. Note that they are both feasible and* $\boldsymbol{x}^{(1)}$ *is binding in the first constraint and* $\boldsymbol{x}^{(2)}$ *in the second. Their sum is* $\mathcal{A}_{add}(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(K)}; \boldsymbol{w})(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}; [1, 1]) \to \boldsymbol{x}^{(A)} = (3, 5, 3)$*, which is also feasible but does not have any binding constraints.*

C Empirical setup for Section 2.4

Task and Setup. We study a citation task where the system designer seeks a list of 10 influential LLM papers spanning five perspectives: (1) ML theory, (2) NLP/CL, (3) cognitive science, (4) AI alignment and human–AI interaction, and (5) multi-agent systems. The system designer issues five prompts, each targeting one perspective, to gpt-4o-mini-2024-07-18 and then aggregates the resulting lists. We prompt another LLM (also gpt-4o-mini-2024-07-18) to aggregate these five lists, instantiating aggregation rules that are inspired by intersection aggregation $\mathcal{A}_{\text{intersect}}$ and union aggregation $\mathcal{A}_{\text{add}}$. Specifically, the model is prompted with *aggregation instructions* along with the five different lists of 10 references, and produces an aggregated list of 10 references. The *intersection-style aggregation instructions* ask for references which are central and broadly relevant across all five perspectives, thus approximating intersection even when the literal overlap of references is empty. The *addition-style* aggregation instructions ask for references that jointly cover and reflect the combined topical space of all five perspectives. We defer the specific prompts and other details of the empirical setup to Section C.

Model output generation. The outputs are generated using gpt-4o-mini-2024-07-18 with the temperature set to 1.0. These are the five prompts that are used to produce model outputs:

1. "From a machine learning theory perspective, list 10 influential papers that have shaped our current understanding of large language models."
2. "From the perspective of natural language processing and computational linguistics, list 10 key research papers that have been most influential in the development of modern large language models."
3. "From a cognitive science and psycholinguistics standpoint, list 10 important papers that inform our understanding of how large language models represent, process, or acquire linguistic and conceptual structure."
4. "From the standpoint of AI alignment and human–AI interaction, list 10 important papers that have shaped how large language models are aligned, instructed, or trained with feedback."
5. "From a multi-agent and game-theoretic perspective, list 10 influential papers that contribute to the development or understanding of large language models"

These prompts produce five outputs $X_1, \ldots, X_5$, each a list of 10 papers tailored to its respective perspective. Next, we pass the concatenated outputs $(X_1, \ldots, X_5)$ to gpt-4o-mini-2024-07-18 by prompting the model with *aggregation instructions* followed by the concatenation of the 5 lists of papers, where each list is preceded by "List of papers: [insert output number]". The intersection-style and addition-style aggregation operations are performed using the following *aggregation instructions*.

- *Addition-style aggregation:* "Each of the following lists contains influential papers on large language models in specializing in different areas: machine learning theory, natural language processing, computational linguistics, AI alignment, human–AI interaction, and multi-agent systems. Based on these lists, generate a new list of 10 papers that reflects the union of their themes and coverage. Your list should be freshly generated (not a literal set union), but it should include papers that plausibly come from any of the provided lists, covering as much of the combined topical space as possible."

- *Intersection-style aggregation:* "Each of the following lists contains influential papers on large language models in specializing in different areas: machine learning theory, natural language processing, computational linguistics, AI alignment, human–AI interaction, and multi-agent systems. Based on these lists, generate a new list of 10 papers that reflects their intersection. That is, papers belonging to many of these areas of specialization. Your list should be freshly generated (not a literal intersection), selecting papers that could plausibly appear in all of the lists. If the literal intersection is empty, still generate the best possible list of papers that are central, broadly relevant, and thematically compatible with all lists."

These aggregation prompts produce outputs $X_{\text{addition}}, X_{\text{intersection}}$.

Output vector computation. We now describe in more detail how we compute the embeddings shown in Figure 2. We embed and visualize the set $\{X_1, \ldots, X_5, X_{\text{addition}}, X_{\text{intersection}}\}$. We calculate the 768-dimensional embeddings using all-mpnet-base-v2 [Reimers and Gurevych, 2019], which is built into the sentence-transformers package in pytorch. To make these embeddings fit into our framework, we translate them to the nonnegative orthant by applying an additive shift $\mathbf{s} \in \mathbb{R}^{768}_{\geq 0}$ To do this, we compute the embeddings of the 805 gpt-4o-mini-2024-07-18 outputs from the helpful-base dataset in AlpacaEval [Li et al., 2023]. The additive shift $\mathbf{s}$ is taken to be negative of the minimum coordinate along each dimension in this set of 805 embeddings. We translate all 5 outputs and the aggregated outputs by adding $\mathbf{s}$. We compute the variance across the 5 translated outputs vectors along each of the 768 dimensions, and select the top 2 and top 3 dimensions according to variance. We also compute the ℓ_2-distance between outputs, which is invariant to the additive shift.

D Other empirical details for Section 5

All of our models use the latest version of GPT-4o-mini as of January 2026. We set the temperature to be 0.7 for generating the output vectors, and set the temperature to be 0 for the LLM-as-a-judge calls to elicit highest probability tokens from the LLM.

Prompting. A prompt is defined by an inclusion set $G^{inc} \subseteq G$, exclusion set $G^{exc} \subseteq G$ that is disjoint from the inclusion set G^{inc}, and operators $\text{op}^{\text{inc}}, \text{op}^{\text{exc}} \in \{\text{and}, \text{or}\}$. If the inclusion list has more than one element, we choose an inclusion list operator op_{inc} that is one of $\{\text{and}, \text{or}\}$ to combine the topics in the list for inclusion. Similarly, if the exclusion list has more than one element, we choose an exclusion list operator op_{exc} that is AND or OR. The prompt based on $G^{inc} = \{g_1^{inc}, \ldots, g_a^{inc}\}, G_{exc} = \{g_1^{exc}, \ldots, g_b^{exc}\}$, op_{inc}, op_{exc} is "List up to 20 papers on g_1^{inc} op_{inc} $\ldots$ op_{inc} g_a^{inc}, but EXCLUDE any papers about g_1^{exc} op_{exc} $\ldots$ op_{exc} g_b^{exc}. For example, the prompt with $G^{inc} = \{\text{Topic 1}, \text{Topic 2}\}$, $G^{exc} = \{\text{Topic 3}, \text{Topic 4}\}$, op_{inc} : AND, op_{exc} : OR is *List up to 20 papers on Topic 1 AND Topic 2, but EXCLUDE any papers about Topic 3 or Topic 4.*

LLM-as-judge setup. We use an LLM-as-judge to measure whether a list L exhibits a given topic T by prompting it.

```
For each of the following research papers, determine whether it belongs
to each topic.
Treat each paper independently - do not let other papers in the list
influence your classification.

Papers:
{paper_list}

Topics:
{topic_list}
```

```
For each paper, independently answer yes/no for each topic.

Output format:
1. {topic_keys_str.replace(', ', ': yes/no, ')}: yes/no
2. {topic_keys_str.replace(', ', ': yes/no, ')}: yes/no
...

SUMMARY (count of "yes" for each topic):
{chr(10).join(f'{key}: [count]' for key in topics.keys())}
```

Aggregation. To perform aggregation, we first reformat the titles produced by an LLMs response by making everything lower case, removing punctuations and removing any extra whitespaces. We further remove duplicates by considering two titles equivalent if one is a substring of another. The aggregation rule $\mathcal{A}^{\cap}$ intersects the lists and $\mathcal{A}^{\cup}$ takes the union of the lists. Note that in contrast with Appendix C, we consider this form of literal set intersection and union as it allows us to consider a non-LLM based aggregation that does not inherit the behavior of the LLMs.

Feasibility and elicitability. To analyze feasibility and elicitability, we brute force over all possible prompt specifications. This brute force search enables us to identify the closest output vector $x^*(x^{(A)})$ to the aggregated output vector.

Confidence intervals. For each output vector, we average over 30 trials and report a coordinate-wise 95% confidence interval. We use a Wilson confidence interval. To measure a confidence interval for the ℓ_1 distance between two output vectors, we use the confidence intervals for the two vectors themselves, and then we compute minimum possible and maximum possible ℓ_1 distance for those confidence sets. We use these confidence sets to compute $\boldsymbol{x}_O^*(\boldsymbol{x}^{(A)})$ and $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$. Specifically, for each candidate output vector x, we compute the *lower* bound in the confidence set for the ℓ_1 distance between $\boldsymbol{x}$ and $\boldsymbol{x}^{(A)}$, and then we choose the $\boldsymbol{x}$ that minimizes this lower bound.

E Additional experiments

Building on the setup in Section 5, we conduct the following additional experiments where we vary our model configuration in three ways.

1. Experimental setup **E1** considers different settings for the temperature (0.3, 0.5, 0.7), and we report our findings in Tables 3-5.
2. Experimental setup **E2** considers different models (GPT-5.4 at temperatures 0 and 0.7, and GPT-5-mini at default temperature), and we report our findings in Tables 6-8.
3. Experimental setup **E3** considers the aggregation of heterogeneous models (GPT-4o-mini and GPT-5-mini; GPT-4o-mini and GPT-5.4; GPT-5-mini and GPT-5.4), and we report our findings in Tables 9-11.

We summarize our key findings.

- We find that support expansion readily generalizes. In fact, these mechanisms hold for the same instances (i.e., task, aggregation operation, and prompt/output specification) as in our original experiments in all of the cases.
- We find that binding-set contraction generalizes in nearly all cases. The exception is one case (Table 8; **E2** for GPT-5-mini), where intersection aggregation produces an empty list ($\boldsymbol{x}^{(A)} = \mathbf{0}$), even after switching to a slightly modified prompt specification intended to encourage a non-empty intersection. As

with the feasibility expansion failures below, this failure mode is driven by intersection of the produced lists being empty. In all other cases, binding-set contraction holds for the same task, aggregation operation, output specification, and prompt specifications as in our original experiments.

- We find that feasibility expansion also generalizes to **E1** and **E2**. Again, these mechanisms hold for the same tasks and instances as our original experiments. However, feasibility expansion is no longer exhibited by the instance that we constructed for **E3** in two out of the three cases. The failure mode is that intersection aggregation produces an empty list likely due to model heterogeneity. We defer constructing instances that exhibit feasibility expansion for other models to future work.

Altogether, these experiments provide further support for these mechanisms, and additionally demonstrate the robustness of our experiments across almost all the generalized settings.

Experiment 1 (E1): Changing temperature

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.63, 0.70]$	$[0.00, 0.00]$	$[0.04, 0.08]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.75, 0.82]$	$[0.03, 0.06]$
$\boldsymbol{x}^{(A)}$	$[0.34, 0.41]$	$[0.43, 0.51]$	$[0.03, 0.06]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.00, 0.01]$	$[0.78, 0.88]$	$[0.01, 0.05]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.66, 0.73]$	$[0.35, 0.42]$	$[0.01, 0.02]$
$\boldsymbol{x}^{(2)}$	$[0.65, 0.72]$	$[0.00, 0.00]$	$[0.32, 0.40]$
$\boldsymbol{x}^{(A)}$	$[0.70, 0.77]$	$[0.00, 0.00]$	$[0.14, 0.20]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.63, 0.75]$	$[0.00, 0.01]$	$[0.29, 0.42]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.77, 0.84]$	$[0.09, 0.15]$	$[0.00, 0.00]$	$[0.00, 0.02]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(2)}$	$[0.17, 0.23]$	$[0.00, 0.00]$	$[0.64, 0.72]$	$[0.04, 0.08]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(A)}$	$[0.93, 0.96]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.01, 0.03]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.78, 0.88]$	$[0.07, 0.15]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.02, 0.07]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.59, 0.91]
Feasibility expansion	[0.09, 0.43]
Binding-set contraction	[0.12, 0.45]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 3: Results for **gpt-4o-mini** and **temperature 0.3**. The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. These results demonstrate support expansion, binding-set contraction, and feasibility expansion.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.65, 0.72]$	$[0.00, 0.00]$	$[0.03, 0.07]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.73, 0.80]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(A)}$	$[0.34, 0.42]$	$[0.44, 0.52]$	$[0.03, 0.06]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.00, 0.01]$	$[0.76, 0.86]$	$[0.03, 0.09]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.67, 0.74]$	$[0.36, 0.44]$	$[0.04, 0.07]$
$\boldsymbol{x}^{(2)}$	$[0.65, 0.72]$	$[0.00, 0.00]$	$[0.33, 0.41]$
$\boldsymbol{x}^{(A)}$	$[0.56, 0.64]$	$[0.00, 0.00]$	$[0.14, 0.20]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.61, 0.74]$	$[0.00, 0.01]$	$[0.30, 0.43]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.80, 0.86]$	$[0.09, 0.14]$	$[0.00, 0.00]$	$[0.00, 0.02]$	$[0.01, 0.04]$
$\boldsymbol{x}^{(2)}$	$[0.18, 0.24]$	$[0.00, 0.01]$	$[0.62, 0.69]$	$[0.07, 0.11]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(A)}$	$[0.91, 0.95]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.78, 0.88]$	$[0.05, 0.13]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.04]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.56, 0.91]
Feasibility expansion	[0.10, 0.49]
Binding-set contraction	[0.08, 0.37]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 4: Results for **gpt-4o-mini** and **temperature 0.5**. The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. These results demonstrate support expansion, binding-set contraction, and feasibility expansion.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.70, 0.77]$	$[0.00, 0.00]$	$[0.04, 0.08]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.74, 0.81]$	$[0.03, 0.06]$
$\boldsymbol{x}^{(A)}$	$[0.39, 0.47]$	$[0.43, 0.51]$	$[0.03, 0.06]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.68, 0.80]$	$[0.00, 0.01]$	$[0.05, 0.12]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.65, 0.72]$	$[0.36, 0.44]$	$[0.06, 0.10]$
$\boldsymbol{x}^{(2)}$	$[0.65, 0.72]$	$[0.00, 0.00]$	$[0.35, 0.42]$
$\boldsymbol{x}^{(A)}$	$[0.63, 0.70]$	$[0.00, 0.00]$	$[0.19, 0.25]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.60, 0.73]$	$[0.00, 0.01]$	$[0.33, 0.46]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.79, 0.85]$	$[0.09, 0.14]$	$[0.00, 0.02]$	$[0.00, 0.02]$	$[0.03, 0.06]$
$\boldsymbol{x}^{(2)}$	$[0.16, 0.22]$	$[0.00, 0.01]$	$[0.61, 0.69]$	$[0.04, 0.08]$	$[0.02, 0.05]$
$\boldsymbol{x}^{(A)}$	$[0.84, 0.89]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.78, 0.88]$	$[0.07, 0.16]$	$[0.00, 0.01]$	$[0.00, 0.03]$	$[0.00, 0.04]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.63, 1.02]
Feasibility expansion	[0.07, 0.39]
Binding-set contraction	[0.07, 0.35]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 5: Results for **gpt-4o-mini** and **temperature 0.7** (the default model and temperature used in the main text). This is the same data as in Table 1.

Experiment 2 (E2): Changing model

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.65, 0.78]$	$[0.00, 0.01]$	$[0.00, 0.03]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.03]$	$[0.40, 0.53]$	$[0.16, 0.28]$
$\boldsymbol{x}^{(A)}$	$[0.34, 0.48]$	$[0.42, 0.56]$	$[0.02, 0.08]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.24, 0.31]$	$[0.29, 0.36]$	$[0.21, 0.27]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.47, 0.61]$	$[0.45, 0.59]$	$[0.20, 0.32]$
$\boldsymbol{x}^{(2)}$	$[0.36, 0.50]$	$[0.00, 0.01]$	$[0.70, 0.81]$
$\boldsymbol{x}^{(A)}$	$[0.99, 1.00]$	$[0.00, 0.01]$	$[0.74, 0.85]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.96, 0.99]$	$[0.05, 0.12]$	$[0.70, 0.82]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.73, 0.85]$	$[0.20, 0.32]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}^{(2)}$	$[0.31, 0.45]$	$[0.00, 0.01]$	$[0.51, 0.65]$	$[0.09, 0.18]$	$[0.00, 0.01]$
$\boldsymbol{x}^{(A)}$	$[0.15, 0.26]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.82, 0.91]$	$[0.00, 0.04]$	$[0.00, 0.04]$	$[0.00, 0.01]$	$[0.07, 0.15]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.22, 0.76]
Feasibility expansion	[0.03, 0.31]
Binding-set contraction	[0.61, 0.99]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 6: Results for **gpt-5.4** and a temperature of 0. The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. These results demonstrate support expansion, binding-set contraction, and feasibility expansion.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.72, 0.79]$	$[0.00, 0.00]$	$[0.01, 0.04]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.46, 0.54]$	$[0.17, 0.23]$
$\boldsymbol{x}^{(A)}$	$[0.40, 0.48]$	$[0.44, 0.52]$	$[0.03, 0.07]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.29, 0.37]$	$[0.37, 0.45]$	$[0.15, 0.21]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.49, 0.57]$	$[0.48, 0.56]$	$[0.17, 0.24]$
$\boldsymbol{x}^{(2)}$	$[0.37, 0.45]$	$[0.00, 0.00]$	$[0.77, 0.83]$
$\boldsymbol{x}^{(A)}$	$[0.98, 1.00]$	$[0.11, 0.16]$	$[0.88, 0.93]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.93, 0.96]$	$[0.02, 0.04]$	$[0.68, 0.75]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.83, 0.89]$	$[0.12, 0.17]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.02]$
$\boldsymbol{x}^{(2)}$	$[0.33, 0.41]$	$[0.00, 0.00]$	$[0.55, 0.63]$	$[0.11, 0.16]$	$[0.01, 0.02]$
$\boldsymbol{x}^{(A)}$	$[0.30, 0.37]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.71, 0.78]$	$[0.02, 0.04]$	$[0.01, 0.03]$	$[0.00, 0.01]$	$[0.14, 0.20]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.12, 0.52]
Feasibility expansion	[0.22, 0.47]
Binding-set contraction	[0.49, 0.77]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 7: Results for **gpt-5.4** and **temperature 0.7**. The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. These results demonstrate support expansion, binding-set contraction, and feasibility expansion.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.83, 0.89]$	$[0.00, 0.00]$	$[0.05, 0.09]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.00]$	$[0.40, 0.48]$	$[0.26, 0.33]$
$\boldsymbol{x}^{(A)}$	$[0.41, 0.49]$	$[0.40, 0.48]$	$[0.12, 0.18]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.36, 0.49]$	$[0.13, 0.23]$	$[0.34, 0.47]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.33, 0.40]$	$[0.71, 0.78]$	$[0.01, 0.03]$
$\boldsymbol{x}^{(2)}$	$[0.32, 0.40]$	$[0.00, 0.02]$	$[0.79, 0.85]$
$\boldsymbol{x}^{(A)}$	$[0.02, 0.05]$	$[0.02, 0.05]$	$[0.02, 0.05]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.00, 0.01]$	$[0.04, 0.10]$	$[0.01, 0.04]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.60, 0.68]$	$[0.28, 0.35]$	$[0.03, 0.06]$	$[0.00, 0.00]$	$[0.08, 0.13]$
$\boldsymbol{x}^{(2)}$	$[0.25, 0.32]$	$[0.00, 0.01]$	$[0.66, 0.74]$	$[0.02, 0.05]$	$[0.01, 0.03]$
$\boldsymbol{x}^{(A)}$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$	$[0.00, 0.00]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.32, 0.83]
Feasibility expansion	[0.01, 0.18]
Binding-set contraction	[0.00, 0.00]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 8: Results for **gpt-5-mini** with the default temperature. The empirical setup and construction of the instance (i.e., task, aggregation operation, and output specification) is the same as in Section 5. For support expansion and feasibility expansion, we use the same prompt specification as in Section 5. For binding-set contraction, the prompts used to generate $x^{(1)}$ and $x^{(2)}$ slightly differ: we set $G_1^{\text{inc}} = \left\{\mathcal{T}_1^P\right\}, G_1^{\text{exc}} = \left\{\mathcal{T}_2^P\right\}$, and we set $G_2^{\text{inc}} = \left\{\mathcal{T}_1^P, \mathcal{T}_2^P\right\}, \text{op}_2^{\text{inc}} = \{\text{or}\}$. These results demonstrate support expansion and feasibility expansion (modestly), but not binding-set contraction: intersection aggregation produces an empty list ($\boldsymbol{x}^{(A)} = \mathbf{0}$), even with the modified prompt specifications described above.

Experiment 3 (E3): Combining heterogeneous models

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.62, 0.71]$	$[0.00, 0.01]$	$[0.01, 0.04]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.01]$	$[0.80, 0.87]$	$[0.01, 0.03]$
$\boldsymbol{x}^{(A)}$	$[0.29, 0.38]$	$[0.45, 0.54]$	$[0.01, 0.03]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.24, 0.31]$	$[0.29, 0.36]$	$[0.21, 0.27]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.66, 0.75]$	$[0.32, 0.42]$	$[0.01, 0.04]$
$\boldsymbol{x}^{(2)}$	$[0.37, 0.47]$	$[0.00, 0.01]$	$[0.72, 0.80]$
$\boldsymbol{x}^{(A)}$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.01, 0.04]$	$[0.00, 0.01]$	$[0.00, 0.01]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.80, 0.87]$	$[0.08, 0.14]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.02]$
$\boldsymbol{x}^{(2)}$	$[0.30, 0.39]$	$[0.00, 0.01]$	$[0.56, 0.66]$	$[0.09, 0.16]$	$[0.00, 0.02]$
$\boldsymbol{x}^{(A)}$	$[0.21, 0.29]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.02, 0.08]$	$[0.00, 0.01]$	$[0.01, 0.04]$	$[0.00, 0.03]$	$[0.09, 0.18]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.26, 0.66]
Feasibility expansion	[0.00, 0.07]
Binding-set contraction	[0.21, 0.54]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 9: Results for aggregating two different models: **gpt-4o-mini** (temp 0) and **gpt-5.4** (temp 0). Support expansion and binding set contraction seems to work. The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is chosen via brute force search over prompt topics and over the two models. These results demonstrate support expansion, binding-set contraction. Feasibility expansion does not appear to occur for this instance due to intersection resulting in empty lists.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.70, 0.78]$	$[0.00, 0.01]$	$[0.04, 0.09]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.01]$	$[0.76, 0.83]$	$[0.01, 0.04]$
$\boldsymbol{x}^{(A)}$	$[0.32, 0.41]$	$[0.45, 0.55]$	$[0.03, 0.08]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.00, 0.01]$	$[0.42, 0.55]$	$[0.21, 0.33]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.66, 0.75]$	$[0.31, 0.41]$	$[0.01, 0.05]$
$\boldsymbol{x}^{(2)}$	$[0.30, 0.40]$	$[0.00, 0.02]$	$[0.74, 0.82]$
$\boldsymbol{x}^{(A)}$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.01, 0.04]$	$[0.00, 0.01]$	$[0.00, 0.01]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.59, 0.69]$	$[0.28, 0.37]$	$[0.01, 0.04]$	$[0.00, 0.01]$	$[0.07, 0.12]$
$\boldsymbol{x}^{(2)}$	$[0.15, 0.22]$	$[0.00, 0.01]$	$[0.68, 0.76]$	$[0.05, 0.10]$	$[0.00, 0.02]$
$\boldsymbol{x}^{(A)}$	$[0.99, 1.00]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.78, 0.88]$	$[0.07, 0.16]$	$[0.00, 0.01]$	$[0.00, 0.03]$	$[0.00, 0.04]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.43, 0.84]
Feasibility expansion	[0.00, 0.07]
Binding-set contraction	[0.18, 0.46]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 10: Results for aggregating two different models: **gpt-4o-mini** (temp 0) and **gpt-5-mini** (temp default). The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is chosen via brute force search over prompt topics and over the two models. These results demonstrate support expansion, binding-set contraction. Feasibility expansion does not appear to occur for this instance due to intersection resulting in empty lists.

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.70, 0.78]$	$[0.00, 0.01]$	$[0.04, 0.09]$
$\boldsymbol{x}^{(2)}$	$[0.00, 0.01]$	$[0.36, 0.45]$	$[0.15, 0.23]$
$\boldsymbol{x}^{(A)}$	$[0.32, 0.41]$	$[0.45, 0.54]$	$[0.05, 0.10]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.24, 0.31]$	$[0.29, 0.36]$	$[0.21, 0.27]$

(a) Support Expansion

	x_1	x_2	x_3
$\boldsymbol{x}^{(1)}$	$[0.34, 0.43]$	$[0.72, 0.80]$	$[0.01, 0.03]$
$\boldsymbol{x}^{(2)}$	$[0.39, 0.48]$	$[0.00, 0.01]$	$[0.69, 0.77]$
$\boldsymbol{x}^{(A)}$	$[0.30, 0.40]$	$[0.21, 0.29]$	$[0.21, 0.29]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.41, 0.55]$	$[0.25, 0.37]$	$[0.36, 0.50]$

(b) Feasibility Expansion

	x_1	x_2	x_3	x_4	x_5
$\boldsymbol{x}^{(1)}$	$[0.62, 0.71]$	$[0.25, 0.34]$	$[0.02, 0.05]$	$[0.00, 0.01]$	$[0.05, 0.10]$
$\boldsymbol{x}^{(2)}$	$[0.32, 0.41]$	$[0.00, 0.01]$	$[0.55, 0.64]$	$[0.09, 0.16]$	$[0.00, 0.01]$
$\boldsymbol{x}^{(A)}$	$[0.16, 0.24]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$
$\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$	$[0.15, 0.26]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.00, 0.01]$	$[0.41, 0.55]$

(c) Binding-set contraction

Experiment	$\|\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)}) - \boldsymbol{x}^{(A)}\|_1$
Support expansion	[0.20, 0.66]
Feasibility expansion	[0.08, 0.70]
Binding-set contraction	[0.40, 0.69]

(d) ℓ_1 distance from $\boldsymbol{x}^{(A)}$

Table 11: Results for aggregating two different models: **gpt-5.4** (temp 0) and **gpt-5-mini** (temp default). The empirical setup and construction of the instance (i.e., task, aggregation operation, and prompt/output specification) is the same as in Section 5. $\boldsymbol{x}_P^*(\boldsymbol{x}^{(A)})$ is chosen via brute force search over prompt topics and over the two models. These results demonstrate support expansion, binding-set contraction, and feasibility expansion.