Are LLMs Reliable Coders of Communication Content in Economic Experiments?

Working Paper #0115

Andrzej Baranski, David J. Cooper and Jeong Kyu Lee

NYU Abu Dhabi

June 2026

جامعة نيويورك أبوظبي

NYU ABU DHABI

Are LLMs Reliable Coders of Communication Content in Economic Experiments?*

Andrzej Baranski † David J. Cooper ‡ Jeong Kyu Lee §

June 13, 2026

Analysis of free-form communication from experiments has largely relied on manual coding by research assistants (RAs), a costly and time-consuming process. We outline an easily implemented method for coding communication data using large language models (LLMs) and propose a novel standard for evaluating the performance of LLM-based coding ("reliability"). Using data from three published articles, we find that LLM-based coding meets our two reliability conditions: (1) differences between LLM-based and RA-based coding are no larger than differences between the RA-based and original coding and (2) the LLM-based coding largely replicates qualitative conclusions from the original papers. That said, there are cases where the LLM-based coding agrees poorly with the RA-based coding or fails to replicate statistical results from the original papers. We demonstrate that these problems can be ameliorated with better prompt design. We conclude that use of LLMs can reduce research costs and time without sacrificing reliability, making content analysis a more accessible tool for experimental economists. However, only with a combination of test coding by RAs and prompt design by researchers can we avoid significant problems with LLM-based coding, highlighting the continued importance of human input.

Keywords: communication, LLM, methodology, annotation tasks

*Andrzej Baranski gratefully acknowledges financial support from Tamkeen under the NYU Abu Dhabi Research Institute Award CG005. The authors are grateful for the support from the Social Science Experimental Laboratory at NYUAD

†Division of Social Science & Center for Behavioral Institutional Design, NYU Abu Dhabi, email: *a.baranski@nyu.edu*.

‡Tippie College of Business, The University of Iowa, email: *david-j-cooper@uiowa.edu*.

§NYU Abu Dhabi Social Science Experimental Laboratory, email: *jkl499@nyu.edu*.

1 Introduction

Experimental economics has a rich tradition of studies involving communication (Brandts et al., 2019; Martinelli and Palfrey, 2020). Within this literature, experiments using free-form communication are valuable as a rich source of process data.[1] Quantifying and analyzing the content of free-form communication ("content analysis") can yield insights into subjects' preferences, their reasoning processes, and the mechanisms by which communication affects behavior and outcomes.

Given the valuable insights it can provide, experimental economists have made surprisingly little use of content analysis. A systematic review of the literature returned 114 articles published between 2000 and 2025 in major economics journals where subjects sent free-form written messages.[2] Of these studies, 96 reported the results of a quantitative analysis of communication content. That may sound like a large number, but amounts to less than four articles per year!

The limited use of content analysis in experimental economics likely reflects the challenging process necessary to quantify communication data ("coding").[3] The coding process typically begins with the researchers developing categories that represent major themes in the communication data (e.g., all messages in which subjects make promises about their future behavior). The researchers explain the categories to a team of research assistants ("RAs") who then independently assign blocks of text to categories.[4] For a reasonable size dataset, the entire process takes months to complete and costs many thousands of dollars. The high cost of coding in time and money has almost certainly been a major barrier to the use of content analysis in experimental economics.

The advent of large language models (LLMs) has the potential to reduce the costs of

[1] Other forms of process data used by experimental economists include response times, eye-tracking, Mouselab, facial expressions, and biometric measures such as heart rate, skin conductance, and brain activation (Cooper et al., 2019).

[2] We searched 19 economics journals for keyword sets "chat AND experiment", "communication AND experiment" as well as articles cited in two reviews (Brandts et al., 2019; Martinelli and Palfrey, 2020). We inspected and removed studies without free-form communication (i.e., pre-defined messages or action space messages). Table A1 summarizes our review of the literature.

[3] For discussions of coding methodology for the social sciences see Boyatzis (1998), Gibbs (2018), Adu (2019), and Saldaña (2021). We describe the coding methods most commonly used in experimental economics. See Houser and Xiao (2011) for an alternative technique using coordination games between coders.

[4] The unit of coding varies from study to study; common choices include coding each message separately or coding the entire conversation associated with a single decision as a unit. We refer to the unit of coding as a *communication episode*

coding and unlock content analysis as an easily used source of process data.[5] Economists have long used natural language processing techniques to analyze text (Gentzkow et al., 2019) and have recently started using LLMs to perform a variety of text analysis tasks that previously required the use of RAs.[6] Experimental economists have also begun to use LLMs for coding (Lee and Hoffman, 2025; Erkut and Reuben, 2025) and there is every reason to believe that this will become more common. Reducing the costs of content analysis is highly desirable, but the lack of clear methodological standards raises the possibility that widespread use of LLMs will harm the quality of the resulting coding and make it harder to compare results across studies.[7] This article is intended to provide some first steps towards a validated methodology for the use of LLMs to code the content of communication from economic experiments.

To this end, we introduce a simple method for using LLMs to code communication from economic experiments. Our goal is *not* to argue whether researchers should use LLMs for coding — that ship has sailed — but rather to provide an easily implemented method, evaluate its performance, and explore how to best implement LLM-based coding. We do not aim to identify an optimal coding method, but rather to describe a method that most researchers with a reasonable level of expertise with LLMs can successfully adopt.[8]

Our baseline setup employs three LLMs. The LLMs are given a description of the experiment, including the subject instructions, instructions about how coding should be done, and descriptions of the coding categories (see Subsections 4.3 and 4.4 for a detailed discussion of the prompts and Appendix D for the full prompts). Each LLM performs the coding task five times and we aggregate the responses by taking the mode of each category in each communication episode. The three LLM codings are aggregated into a single LLM-based coding by taking the mode.

We apply our baseline method of LLM-based coding to communication data from three experiments (Charness and Dufwenberg, 2006; Brandts et al., 2015; Baranski and Haas, 2023). These articles vary the type of game studied, the structure of the communication channels (simple one-way messaging vs. complex multilateral communication), and the scope of interactions (single-shot, repeated with fixed matching, and repeated with ran-

[5]The cost of the LLM-based coding for this paper was less than a fifth of the cost of the RA-based coding.

[6]See Gorodnichenko et al. (2025) and Bastianello et al. (2024) for examples.

[7]This has also been a problem for traditional RA-based coding where there do not exist commonly accepted standards for basic issues such as the number of coders or the unit of coding.

[8]For example, we use commercially available LLMs, in line with our goal of easy implementation, rather than training a custom model.

dom rematching). We use experiments with differing characteristics to ascertain whether our method is broadly applicable.

We did not expect the LLMs to perfectly replicate coding created by RAs. RAs often disagree about how to code content, reflecting inherent ambiguity in the text, and we expect the LLMs and RAs to disagree in a similar fashion. We therefore introduce a new standard for evaluating the performance of LLM-based coding that allows for disagreements between LLMs and RAs. An LLM-based coding method is *reliable* if it satisfies two criteria: (1) the differences between an LLM-based coding and an RA-based coding are no larger than the differences between two independent implementations of RA-based coding and (2) conclusions reached about the relationship between the content of communication and outcomes should not depend on whether the coding was created by LLMs or RAs. More concretely, the second criterion asks whether we reach the same conclusions as the original papers if we replicate their statistical analyses using LLM-based coding.

Our two criteria for reliability are related but not identical. We develop a simple model of coding that illustrates this point. Critically, our model moves away from two assumptions that are implicit in virtually all of the existing literature. First, we do not assume that RA-based coding is a good proxy for the "ground truth". We instead model it as a signal of the "ground truth". Second, we do not treat coding by LLMs as a noisy approximation of the "ground truth". LLMs are designed to mimic humans and coding prompts typically instruct the LLM to act as if it is a research assistant or something similar. We therefore treat the LLM-based coding as a noisy signal of the RA-based coding. We show that these two departures from standard approaches to comparing LLM-based and RA-based coding have an important implication: the LLM-based coding will agree less with the ground truth than the RA-based coding even if it agrees well with the RA-based coding. It follows from this observation that we can construct examples where Criterion 1 for reliability holds but Criterion 2 does not. The two criteria are obviously related, but Criterion 2 is not redundant.

Having established the need for both criteria, the first main conclusion from our empirical analysis is that the baseline LLM-based coding method produces reliable results, satisfying Criteria 1 and 2. To verify that Criterion 1 holds, we need a second implementation of coding by RAs. We therefore recoded all communication data from the original papers using uniform methods that match the LLM-based coding.[9] Specifically, the number of RAs matched the number of LLMs (three of each) and both sets of coders were given the same

[9]The three papers used different coding methods, varying on basic issues such as the number of coders.

materials. We henceforth refer to the new coding as the "RA-based" coding and the coding in the published articles as the "original" coding.

The results of the LLM-based and RA-based coding do not differ significantly more than the RA-based and original coding differ. The absolute value of the difference between the LLM-based coding and the RA-based coding is small on average (4.2%). This is not significantly more than the difference between the RA-based coding and the original coding (2.8%). Cohen's kappa (Cohen, 1960), the most commonly used measure of agreement between coders, leads to a similar conclusion.[10] Comparing the modal LLM-based coding with the modal RA-based coding, the average Cohen's kappa is .750. This is larger than the average Cohen's kappa between the RA-based coding and the original coding (.682); the LLM-based coding agrees with the RA-based coding slightly more than the RA-based and original coding agree with each other.

We also verify that Criterion 2 holds: The use of LLM-based coding does not alter conclusions the three papers reached about the relationship between the content of communication and outcomes. For example, Charness and Dufwenberg study a modified trust game in which trustees send pre-play messages to trustors. They find that the efficient outcome is more likely when the trustee's message contains a promise. This conclusion depends on accurate coding of promises. We replicate their efficiency result using our new coding of promises *regardless of whether the coders are RAs (z-stat = 3.61) or LLMs (z-stat = 4.68).* The precise levels of statistical significance are not identical, but that's not the point of Charness and Dufwenberg; everything a reader should care about is unaffected by using different coders. We do similar replication exercises for all three papers; the qualitative conclusions reached about the relationships between communication and outcomes do not depend on whether RA-based or LLM-based coding is used. To summarize, based on both criteria, LLM-based coding is reliable as defined above.

The remainder of the paper considers ways our basic methodology can be changed to reduce costs and/or improve performance. We first explore alternative prompt designs. While the LLM-based coding is reliable overall, the quality of the coding varies across categories. For several categories, the Cohen's kappa between RA-based and LLM-based coding falls below the threshold normally considered to indicate substantial agreement. Likewise, while the LLM-based coding does a good job overall of replicating qualitative results from the

[10]Cohen's kappa measures how much two coders agree beyond what would be expected by chance. Cohen's kappa ranges between 0 and 1 with values greater than .6 interpreted as indicating substantial agreement.

original papers, there are details it misses. We examine whether three alternative prompts can address these two problems. The one that works best modifies the baseline prompt to provide better guidance about how to interpret subjects' communication. Groups often develop shorthand terminology for common concepts. This presents little difficulty for RAs but leads to systematic errors by the LLMs. Explaining the meaning of such terms leads to less misinterpretation by the LLMs, improving their performance. Unlike the baseline prompt, with guidance the LLM-based and RA-based coding agree substantially for the most problematic category and all the results from the original papers can be replicated using LLM-based coding.

This leads to our second main conclusion: Completely eliminating human input (researchers and RAs) from the coding process is likely to yield substantial errors. We strongly recommend that researchers have RAs code a subset of their data and compare it to the LLM-based coding to identify terms and expressions that the LLMs need guidance to interpret correctly.

Our baseline coding method would be less expensive with fewer LLMs and/or fewer replications (i.e., repeated codings using the same LLM and prompt) per LLM. However, LLMs are inherently stochastic. It has become common practice to reduce randomness by using multiple models and by prompting each model multiple times, a practice that our baseline coding method follows. We explore whether this is necessary. Expanding upon our simple model, we show that increasing the number of LLMs, but *not* the number of replications, improves agreement between LLM-based coding and the ground truth. Our empirical results are more nuanced. As predicted, additional replications have little value given the high similarity between them (even without taking steps to limit variability). Adding LLMs is more effective but only yields modest increases in agreement between LLM-based and RA-based coding. We argue that shared biases among the LLMs attenuate the positive effects of adding LLMs. Budget-constrained researchers are better off spending money on more LLMs rather than more replications per LLM, but should only expect modest returns from adding LLMs.

Finally, we examine why LLM-based and RA-based coding disagree. Disagreements tend to occur on messages where the RAs themselves disagree. This suggests that disagreements generally reflect fundamental ambiguity in the meaning of messages rather than blatant errors by the LLMs.

In summary, we present an easily implemented LLM-based method for coding commu-

nication data from experiments. We define "reliability", a new standard for judging the performance of LLM-based coding and show that the LLM-based coding is reliable. That said, eliminating researchers and RAs from the coding process is likely to yield substantial coding errors. Human input is necessary to construct prompts that give LLMs good guidance on how to interpret subjects' communication.

We hope that our work will encourage the development of standards for LLM-based coding and facilitate greater use of content analysis by experimental economists. To these ends, our data, prompts, and code are available at https://github.com/jeongk31/encoding-communication-content-humans-vs-ai.

The remainder of the article proceeds as follows. Section 2 offers a review of related work. Section 3 develops a simple model of coding. Section 4 provides a description of the communication data included in our sample, the coding methods (human and LLM), and how we construct our data set for analysis. Section 5 contains our main results comparing agreement rates. Section 6 discusses and concludes the article.

2 Related Literature

There exists a broad literature, primarily in computer science but also in other social sciences, evaluating the performance of LLMs at various tasks that involve quantifying communication data ("*annotation tasks*").[11] The annotation tasks LLMs are asked to perform (e.g., sentiment analysis of tweets) often differ substantially from the coding tasks we consider. Given that the agreement rate between LLMs and RAs varies widely between different types of annotation tasks (Bavaresco et al., 2025), we cannot assume that findings about the reliability of LLM-based coding will translate from other annotation tasks to the coding of communication data from experiments. Even within the domain of economic experiments, differences in communication structure may affect reliability. We therefore use data from three experiments with different structures of communication.

Our approach differs fundamentally from the broader literature on LLM-based annotation. Most studies evaluate LLM performance by measuring agreement with human coders, treating RA-based coding as ground truth. Agreement with RA-based coders is important,

[11]For example, recent work has evaluated LLMs as annotators in domains such as low-resource language NLP tasks involving topic classification, tweet sentiment analysis, and emotion classification (Nasution and Onan, 2024), latent content analysis of texts measuring sentiment, political leaning, emotional intensity, and sarcasm (Bojic et al., 2025), and conversational safety annotation involving toxicity, harassment, and harmful dialogue detection (Movva et al., 2024).

but how much is enough? While appropriate when a verifiable ground truth exists, does it make sense to treat RA-based coding as ground truth when communication data from economic experiments is often ambiguous and RAs frequently disagree in their interpretations? We address these questions through reliability. Criterion 1 provides a benchmark for when agreement is good enough: an LLM-based coding is considered reliable if it agrees with RA-based coding as much as two independent RA-based codings agree. By this standard, LLM-based coding can be viewed as a substitute for RA-based coding whenever it is no less reliable than an independent RA-based coding. Our motivation for Criterion 2, which is our largest departure from the existing literature, relies on the possibility that RA-based coding is not a proxy for the ground truth. Criterion 1 holding does not guarantee that Criterion 2 holds, and ultimately we care more about the relationship between the content of communication and subjects' behavior than whether LLM-based and RA-based coding have high agreement. If we believe that the results from the original studies are valid, LLM-based coding should reproduce them.

To be clear, we have a broadly positive view of the existing literature on the use of LLMs for annotation tasks. Tremendous progress has been made on how to construct prompts for annotation tasks and what types of models work best. Our differences with this literature are largely driven by our central research question: Is LLM-based coding of communication data from economic experiments an acceptable substitute for RA-based coding?

Unlike most papers in the broader literature, Bavaresco et al. (2025) supplement a standard approach to evaluating LLMs' performance on annotation tasks with a measure related to our first criterion for reliability. Most of their analysis focuses on agreement between LLM-based and RA-based coding and lacks a benchmark for what constitutes acceptable performance. However, they also consider a measure ("upper bound") which resembles the alternative approach we use to assess Criterion 1 for reliability (see Subsection 5.1.1).[12] They note that "Except for a few datasets ... model scores remain notably below the upper bound." This result suggests caution regarding the use of LLM-based coding, although its applicability to economic experiments is unclear given that the annotation tasks they examine are not closely related to coding communication from economic experiments. The most important differences between our work and Bavaresco et al. (2025) are conceptual. The upper bound plays a supporting role in their analysis, subordinate to agreement between LLMs and RAs. They view the upper bound as providing a limit on how well LLM-based

[12]See also (Asirvatham et al., 2026) for use of a measure that is similar to the upper bound.

coding can perform, which makes sense if RA-based coding is treated as a proxy for the ground truth. We do not treat the RA-based coding as a proxy for ground truth and, for us, Criterion 1 plays a central role by providing a benchmark for when one can view agreement between LLM-based and RA-based coding as being sufficiently high.

Two other recent studies are closely related to our work because they focus on using LLMs for coding in economic experiments. Like us, Çelebi and Penczynski (2026) study the relationship between LLM-based and RA-based coding, but the two papers focus on different issues and therefore take different approaches. Çelebi and Penczynski (2026) document that LLM-based coding agrees well with RA-based coding, but this is a secondary issue for them. Instead, their main concern is what features of the prompt and model improve agreement. Like the broader literature, they lack a benchmark for what constitutes acceptable performance by LLM-based coding, but this matters less given their emphasis on ordinal comparisons between different prompting methods. For us, the primary issue is whether researchers can safely replace RA-based coding with LLM-based coding. We also look at details of the coding process, but these are secondary concerns. Analyzing raw agreement rates (or even Cohen's kappa) is not sufficient for our purposes; we do need a benchmark. The two criteria of reliability provide benchmarks for whether LLM-based coding is an acceptable substitute for RA-based coding.

Cooper et al. (2026) also study the use of LLMs for coding, but do not emphasize the relationship between RA-based and LLM-based coding. Instead, they focus on two issues: (1) the use of LLMs to develop coding categories and (2) replicability of LLM-based coding. Their goal is to make LLM-based coding similar to running a regression; if one researcher shares their code and data with another, they should, as much as possible, get the same results. We stress reliability (as defined above) and ease of use rather than replicability. The two papers are best viewed as complements.

3 A Simple Model of Coding

This section presents a simple model of coding intended to motivate our two criteria for reliability and to show that Criterion 1 holding does not imply Criterion 2 must hold.

Let the set of messages to be coded be $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$. These are sent to receivers (one per message) whose choices are potentially affected by the contents of the messages. Without loss of generality, assume there is a single coding category. Let $\mathcal{C} =$

$\{GT, RA1, RA2, LLM\}$ be the set of coders. Let c_x for $x \in \mathcal{C}$ be a coding of the messages; these are binary variables. The coding c_{GT} is the ground truth. It perfectly captures all information in each message that is relevant for the receivers' choice. Conceptually, c_{RA1} takes the role of our RA-based coding, the new coding we have created for this paper. Criterion 1 compares the RA-based coding with an independent RA-based coding, using the original coding for this purpose. In our model, c_{RA2} plays this role. The role of c_{LLM} is self-explanatory. Critically, we as researchers played a role in the creation of c_{RA1} and c_{LLM} and therefore expect them to be related. This is not true for c_{RA2}. For simplicity, we replace classes of coders with single coders (i.e., there is a single LLM coder rather than three).

For each message, c_{GT} equals 1 with probability p_{GT} and equals zero otherwise. The RAs' codings are noisy versions of c_{GT}. The probability that an RA's coding of a message equals 1 is given by $p_{RA} * c_{GT} + (1 - p_{RA}) * (1 - c_{GT})$; in other words, p_{RA} is the probability an RA's coding matches the ground truth. Subject to the ground truth, the two RAs' codings are independent. The probability that c_{LLM} equals 1 is given by $p_{LLM} * (\delta * c_{GT} + (1 - \delta) * c_{RA1}) + (1 - p_{LLM}) * (1 - (\delta * c_{GT} + (1 - \delta) * c_{RA1}))$. The LLM's coding reflects a combination of the codings of GT and RA1, but not RA2. If $\delta = 1$, p_{LLM} is the probability that c_{LLM} and c_{GT} match. If $\delta = 0$, p_{LLM} is the probability that c_{LLM} and c_{RA1} match.

LLMs are designed to imitate humans. A typical prompt would give LLM and $RA1$ identical materials and tell the LLM to act as if it is $RA1$. Our formula for the LLM's coding probability captures the possibility that the LLM is trying to code messages the same way $RA1$ would, with $1 - \delta$ measuring how much weight the LLM puts on imitating $RA1$ rather than capturing the ground truth. If $\delta = 0$, the LLM's coding does not depend directly on the ground truth coding – the LLM has probability p_{LLM} of agreeing with $RA1$ regardless of what the ground truth coding is.

After a receiver gets a message m, they take an action, a_m, which is a binary choice; a_m equals 1 with probability $p_{GT=1}$ if $c_{GT} = 1$ and equals 1 with probability $p_{GT=0}$ if $c_{GT} = 0$. We assume $p_{GT=1} > p_{GT=0}$. In other words, the receiver is more likely to take the action if $c_{GT} = 1$. There is a causal relationship between c_{GT} and a_m. The other codings, c_{RA1}, c_{RA2}, and c_{LLM}, can predict a_m but do not cause it.

Suppose the LLM's coding solely reflects the ground truth ($\delta = 1$) and additionally assume $p_{RA} = p_{LLM}$. All three codings, c_{RA1}, c_{RA2}, and c_{LLM}, are equally noisy reflections of the ground truth. This has two implications. Because all three codings are equally likely to agree with c_{GT} and are independent subject to the ground truth, they are equally

likely to agree with each other. This implies that Criterion 1 will hold in the limit (with a finite sample, it may fail due to the randomness of what specific codings are generated). Suppose a researcher tries to find the relationship between a_m and c_{GT}, which is unobserved, by regressing a_m on one of the three codings. Because the data generating processes are identical, the likelihood of finding a significant relationship is identical for all three codings. With $\delta = 1$, Criterion 1 implies Criterion 2.

This is not true if $\delta \neq 1$. Take the opposite extreme, $\delta = 0$; the LLM's coding solely reflects RA1. For concreteness and to facilitate simulations, we henceforth use specific values for the model's parameters: let the number of messages $M = 225$, $p_{GT} = .5$, $p_{RA} = .9$, $p_{LLM} = .82$, $p_{GT=1} = .25$, and $p_{GT=0} = .10$. As additional notation, define $\kappa_{x,y}$ as the Cohen's kappa between the codings c_x and c_y for $x, y \in \mathcal{C}$.[13] The value of p_{LLM} is chosen such that the probability that c_{LLM} and c_{RA1} agree is identical to the probability that c_{RA1} and c_{RA2} agree. The mean values of c_{RA1}, c_{RA2} and c_{LLM} all equal .5 in the limit as the number of iterations goes to infinity. Given this equality, $\kappa_{LLM,RA1}$ and $\kappa_{RA1,RA2}$ are equal in expectation, equaling .64. Taken together, the preceding implies that Criterion 1 is likely to hold. Moreover, most researchers would view the LLM as doing well – we have chosen parameters such that c_{LLM} typically has "substantial agreement" with c_{RA1}.

The structure of the model suggests that Criterion 2 may fail. Given that $\delta = 0$, c_{RA1} is a noisy version of c_{GT} and c_{LLM} is a noisy version of c_{RA1}. This implies that c_{LLM} is a noisier version of c_{GT} than c_{RA1}. For the specific parameters being used, c_{RA1} has 90% chance of agreeing with c_{GT}. While c_{LLM} does a good job of agreeing with c_{RA1}, it is not perfect. More importantly, it only reflects the ground truth *through* RA1's coding. The probability that c_{LLM} agrees with c_{GT} is only 75.6%. In other words, even though it agrees well with c_{RA1}, c_{LLM} has a weaker relationship with c_{GT} than c_{RA1} does. If we want to predict receivers' choices, what matters is agreement with the ground truth, not agreement with another coding.

We use a Monte Carlo exercise to confirm that Criterion 2 will often fail when Criterion 1 holds. We run 2500 iterations. For each iteration, we start by randomly drawing c_{GT} for each message. We then randomly generate c_{RA1} and c_{RA2}. Given c_{RA1}, we next randomly generate c_{LLM}. Finally, given c_{GT}, we randomly generate a_m for each message. At this point we have a randomly generated dataset with 225 observations of c_{GT}, c_{RA1}, c_{RA2}, c_{LLM}, and a_m. To address Criterion 2, we run two regressions, regressing a_m on c_{RA1} and c_{LLM}

[13] See the beginning of Section 5 for discussion of Cohen's kappa and its interpretation.

respectively. We classify the result of a regression as significant if the p-stat $< .05$ for the estimated parameter on the coding (c_{RA1} or c_{LLM}). Criterion 2 holds if the significance matches for the two regressions. As a benchmark, we also regress a_m on c_{GT}.

As should be the case, for all four codings (c_{GT}, c_{RA1}, c_{RA2}, and c_{LLM}) the median coding equals 0.50. Likewise, the median values of $\kappa_{LLM,RA1}$ and $\kappa_{RA1,RA2}$ equal .64, as expected.[14] We classify an iteration as fulfilling Criterion 1 if two conditions hold: the coding probabilities for c_{LLM} and c_{RA1} are *not* significantly different at the 5% level (using McNemar's test) and $\kappa_{LLM,RA1}$ is at least as high as $\kappa_{RA1,RA2}$. This is a tough standard and given the random nature of the simulated data, we do not expect it to hold for all iterations. As it turns out, Criterion 1 holds for 49% of the iterations. For 36% of the iterations where Criterion 1 holds, Criterion 2 fails. Almost always (94%) when Criterion 2 does not hold, it fails because the regression was significant for c_{RA1} but not for c_{LLM}. The failures of Criterion 2 are not purely caused by random noise.

To understand why Criterion 2 can fail even though Criterion 1 holds, it helps to look at the parameter estimates. The theoretical marginal effect of c_{GT} on the receiver's choice is .150. Reflecting this, when we regress a_m on c_{GT}, the median parameter estimate is .148. If we regress a_m on c_{RA1}, the median parameter decreases to .118. With classic measurement error we expect to see parameter estimates attenuated, and that is exactly what occurs. When we regress a_m on c_{LLM}, the median parameter drops again to .076. Because c_{LLM} is noisier relative to c_{GT} than c_{RA1}, the measurement error and the attenuation are stronger. Given the reduced parameter estimates, it is less likely that statistical significance will be achieved with c_{LLM} than c_{RA1}. Using c_{RA1}, the median value of the p-stat is .019. This increases to .137 for c_{LLM}.

Before concluding, we need to make one point clear. We are not claiming that LLM-based coding cannot directly reflect the ground truth. Instead we argue that it cannot be assumed that LLM-based coding is *solely* a noisy reflection of the ground truth. That is not what LLMs are designed to do, nor is it what their prompts imply they should do. If an LLM interprets its task as agreeing with a coder, it may succeed while doing a poorer job of agreeing with the ground truth. Rather than assuming that Criterion 1 implies Criterion 2, we feel it is prudent to directly check that Criterion 2 holds regardless of whether Criterion 1 is satisfied.

[14] These equalities fail with more digits due to the randomness of the data generation process. As the number of iterations goes to infinity, these values will converge to their theoretical predictions.

Appendix B provides an extended version of the model described above. We use the extended model to show that adding LLMs is more likely to improve agreement with the ground truth than adding replications.

4 Coding Methods

This section describes how the RA-based and LLM-based coding were created to evaluate our LLM-based coding method. We start by outlining the structure of the data and introducing the articles from which we obtained the communication data. We then turn to the details of the coding process for RAs and LLMs.

4.1 Data Structure

To ease comparisons, we use the same unit of coding, referred to as a *communication episode*, as the original authors. Communication data is coded at the level of a communication episode.

Each communication episode is coded using the same categories as the original paper. Categories correspond to themes in the communication. For example, a category could include all communication episodes in which a subject sends a promise to take some action. All categories are binary. A communication episode is coded for a category (i.e., assigned a value of 1) if it includes material consistent with the category. Otherwise, the category is not coded (i.e., assigned a value of 0). Coders are allowed to code a communication episode for zero, one, or multiple categories.

4.2 Studies in Sample

We collected communication content (raw data) from three papers, choosing articles with different research questions and structure of communication. Table 1 summarizes the main characteristics and dataset content of the three papers.

Brandts et al. (2015) study the turnaround game. They test whether leadership or increased financial incentives are more effective mechanisms for overcoming coordination failure. Groups play repeatedly, with the leader role and group membership remaining fixed. In treatments with active leaders, the leader sends a message to their group at the beginning

Table 1: Summary of Studies and Coded Data

	Brandts et al. 2015	Charness and Dufwenberg 2006	Baranski and Haas 2023
Label	*Legitimacy*	*Promises*	*Timing*
Game type	Coordination	Trust	Bargaining
Game dynamics	Repeated Partners	One-shot	Repeated Strangers
Communication	One-way Single message	One-way Single message	Two- & Three-way Dialogue
Categories coded	11	3	4
# of Comm. episodes	361	91	750
# of Codings	3,971	273	4,754

of each round. A communication episode consists of a single message sent by a leader. Henceforth, we refer to this study as *Legitimacy*.

Charness and Dufwenberg (2006) investigate the role of *cheap talk* communication in a modified trust game, where second movers send a pre-play message to the first mover. Subjects play the game once. A communication episode consists of a single pre-play message sent by a second mover. We refer to this study as *Promises*.

The original coding scheme for *Promises* includes a single category with three possible labels: no message, promises, and empty (i.e., a message that does not contain a promise). For consistency with the binary coding schemes in the other papers, this category has been transformed into three binary categories: No Messages, Empty, and Promise. Coding No Messages is purely mechanical, with no disagreements occurring among any of the coders, either RAs or LLMs. Subject to a message being sent, Empty and Promise are perfectly negatively correlated by construction. Thus, Promises has a single category in practice. The summary tables contain data for all three categories, but for most of the analysis No Messages and Empty are dropped.

Baranski and Haas (2023) study a three-player multilateral bargaining game that allows for communication at different stages of the game and between different subsets of players in a given stage. Furthermore, the design allows for back-and-forth messaging, as opposed to one-way messages. We focus on the chat that occurs in the proposal stage of the game.

A communication episode consists of the entire conversation between a subset of players.[15] We refer to this study as *Timing*.

4.3 RA Coding

The three papers used different methods for the original coding. Most notably, the number and identity of the coders varied across papers. In *Promises*, one of the authors was the sole coder. *Legitimacy* and *Timing* both employed RAs as coders, but used different numbers of coders (two vs. three). To ensure comparability across studies and coder types (i.e., LLM-based vs. RA-based coding), we recoded the data from all three papers using parallel methods.

For the recoding, all three papers were coded by three different RAs. All the RAs (7 in total) were students at NYU Abu Dhabi and were paid on an hourly basis. For each dataset, the RAs were provided with a manual explaining how to code the data. As much as possible, the structure of the manuals was parallel across studies, ensuring that coders had access to comparable information for the three coding exercises. The RAs were also given the experimental instructions, providing them with additional context to evaluate the meaning of messages. The RAs were instructed to code the messages independently. All the materials can be found in the appendix.

4.4 LLM Coding

Identifying the "best" LLM is not a goal of our paper. The performance of commercially available LLMs is constantly changing, so any conclusions we reached would probably be out of date before the ink dried. Our work is aimed at experimental economists who want to use an easily implemented method for doing LLM-based coding but are concerned whether this is a valid substitute for RA-based coding. Consistent with our emphasis on easy implementation, instead of attempting to identify the *best* model, we selected three currently available commercial models that were prominent, had a good reputation, and were reasonably priced: GPT-4o, Gemini-2.0-Flash, and DeepSeek-Chat.

The LLM-based coding, as expected, is faster and cheaper than the RA-based coding. It took about 7.5 hours for the LLMs to code the data and the total costs were $185. Our RAs

[15]There were multiple windows open during the chat. These consist of private chat windows (bilateral communication between two players) and public chat windows (trilateral communication between all three players).

reported 67 hours of work and, because RAs do not work continuously, the coding took over five weeks to complete. The total cost of the RA-based coding was in excess of $1000. To be clear, our RAs are experienced and, based on past coding exercises conducted by us in other work, we believe they were unusually fast. We would be unsurprised if the gap between RAs and LLMs is typically larger than what is reported here.

The LLM-based coding is done using a *prompt* — instructions which are fed to the LLM via an API.[16] Mirroring the materials given to RAs, our basic prompt consists of two components: experimental instructions and instructions about how to complete the coding task (including a description of the coding strategy). The prompt also included technical material on how the data was formatted and how to output the coding results. We refer to our basic prompt as the *Baseline* prompt.[17] The full text of all prompts used in this paper, including the *Baseline* prompt and the alternative prompts described in Subsection 5.2.1, are provided in Appendix D.

Using an LLM requires us to select *inference parameters* that govern how it implements the prompt. Of particular interest, the *temperature* parameter controls the degree of randomness in output generation, with higher values generating more variability in responses. We use the default value (1.0). There is nothing special about the default value, but, consistent with ease of implementation, our goal is to require as few modifications as possible for the baseline prompt. All other inference parameters were also set to the default levels (see Table D5 in the appendix).

For each LLM and prompt, *Baseline* or alternative, we repeat the coding procedure five times. We refer to these repeated codings using the same LLM with the same prompt as "replications". Given the stochastic nature of LLMs, the coding will vary across replications even though the LLM, prompt, and inference parameters are held constant. For each communication episode, we aggregate the five replications from each LLM by taking the modal coding for each category. Finally, we aggregate across the three LLMs by taking, for each category and communication episode, the mode of the three LLMs' modal codings.[18]

[16] An Application Programming Interface (API) is the software that allows our Python code to send prompts to the LLM and receive responses.

[17] As the use of LLMs has become widespread, a literature has emerged on *prompting* techniques (Schulhoff et al., 2024). No broad consensus exists on nomenclature at this early stage, but, if we had to classify our *Baseline* prompt, it is close in spirit to what has been referred to as *zero-shot* prompting.

[18] Given that the numbers of LLMs and replications are both odd and coding is binary, taking the mode is equivalent to taking the majority.

5 Results

The primary measure of agreement between two codings used in this paper is Cohen's kappa (Cohen, 1968). Agreement rate, defined as the proportion of communication episodes in which both coders (i.e., human and LLM) coded a category identically, is an intuitive measure of similarity between coders. However, it does not account for agreement by chance and therefore is biased by how often a category is coded.[19] To account for agreement by chance, Cohen's kappa reports how much *more* two coders agree than would occur by chance. Possible values of κ range between -1 and 1. If $\kappa = 0$, the two coders are statistically independent, agreeing no more than would occur by chance. At the other extreme, $\kappa = 1$ means the two coders always agree. The standard interpretation of Cohen's kappa is that $\kappa \geq .6$ indicates substantial agreement and $\kappa \geq 0.8$ is considered near perfect agreement (Landis and Koch, 1977).[20]

Some of the categories in our dataset have low coding frequencies (i.e., less than 10%). This implies that our data is prone to the kappa paradox; categories for which coders almost always agree will often have a low Cohen's kappa in spite of the high agreement between coders.[21] Gwet's AC1 is an alternative measure of similarity between two codings developed to address this issue. We therefore include both Cohen's kappa and Gwet's AC1 in some of our tables, and tables in the main text that only use Cohen's kappa are replicated in the appendix with Gwet's AC1.[22]

[19] For example, consider two coders who create completely uncorrelated codings for a category. If both code the category with probability .5, the agreement rate will be .5. If the coding probability is instead .9, the agreement rate increases to .82. The agreement rate makes it appear that the coders agree much more in the second case, when in both cases there is no relationship between the two codings, because the coders are more likely to agree by chance in the second case.

[20] Formally, Cohen's kappa (κ) for a given category is defined as

$$\kappa = \frac{P_o - P_e}{1 - P_e},$$

where P_o denotes observed agreement and P_e denotes expected agreement by chance.

[21] For example, suppose both coders code a category for 1% of communication episodes. The expected agreement rate if the coders are statistically independent is 98%, leaving little room for the agreement rate to be greater than expected by chance. A small number of random errors therefore can have a disproportionate effect on Cohen's kappa.

[22] Higher values of AC1 indicate higher similarity but there are no agreed upon benchmarks for good agreement like Cohen's kappa. By construction, AC1 must be greater than or equal to Cohen's kappa.

5.1 Reliability

Recall that the LLM-based coding must satisfy two criteria to be considered reliable: (1) the results of LLM-based coding should not differ from RA-based coding substantially more than two different implementations of RA-based coding differ from each other and (2) the conclusions reached about the relationship between the content of communication and outcomes (e.g., the qualitative conclusions drawn from regressions using coding results as independent variables) should not depend on whether LLM-based or RA-based coding is used. These two criteria are addressed in subsections 5.1.1 and 5.1.2.

5.1.1 Coding Reliability

Table 2: Mean Responses by Category and Coder Type

Study	Category	LLM	RA	Original
Legitimacy	1a	0.017	0.014	0.014
Legitimacy	1b	0.011	0.014	0.011
Legitimacy	1c	0.003	0.003	0.003
Legitimacy	1d	0.008	0.008	0.007
Legitimacy	1e	0.590	0.598	0.659
Legitimacy	1f	0.091	0.039	0.015
Legitimacy	2	0.144	0.158	0.150
Legitimacy	3	0.022	0.017	0.018
Legitimacy	4	0.319	0.299	0.338
Legitimacy	5	0.055	0.042	0.047
Legitimacy	6	0.454	0.194	0.237
Promises	Empty Talk	0.330	0.286	0.363
Promises	No Message	0.110	0.110	0.110
Promises	Promise	0.560	0.604	0.527
Timing	All-way split	0.158	0.128	0.104
Timing	Competition	0.038	0.027	0.050
Timing	Future Coalition	0.140	0.152	0.232
Timing	MWC	0.205	0.400	0.336
*Average**		0.176	0.169	0.172

*Categories Promises, Empty and Promises, No Messages are excluded from this average.

In our discussion of Criterion 1 for reliability, the RA-based coding serves as the baseline. This data comes from our recoding exercise using common methods for all three papers. The LLM-based coding uses the same instructions and materials as the RA-based coding, but differs in the nature of the coders (RAs vs. LLMs). The original coding serves as an

independent coding created by RAs. The categories, data, and nature of the coders are the same as the RA-based coding,[23] but the instructions and methods (e.g., number of coders, identity of the coders) vary across the papers.

Table 2 reports the average coding by category for the LLM-based, RA-based, and original codings. The last row reports the average across categories, weighting each category equally. As explained earlier, the Empty and No Promises categories for *Promises* are redundant and are therefore excluded from this average and all subsequent analyses.

The average coding across categories is very similar for the LLM-based, RA-based, and original codings. This is not overly informative given that positive differences in one category can be offset by negative differences in others. We therefore focus on absolute differences; for each pair of codings (e.g., LLM-based vs. RA-based), we compute the absolute difference between the mean coding within each category and then average these absolute differences across categories. The average absolute difference captures aggregate disagreement (i.e., do two codings classify categories at similar overall rates?).

If LLM-based coding satisfies Criterion 1 for reliability, the average absolute difference between the LLM-based coding and the RA-based coding should differ little from the average absolute difference between the original coding and the RA-based coding. This is indeed the case: the average absolute difference between LLM-based and RA-based coding is 4.2%, compared to 2.8% between the original and RA-based coding. The LLM-based coding differs little from the RA-based coding on average and, more importantly for reliability, it differs from the RA-based coding by about as much as the original coding does. Running a simple paired t-test, the absolute differences between LLM-based coding and RA-based coding are not significantly different from the absolute differences between the original and RA-based coding ($t = 0.789$; $p = .442$; d.f. $= 15$).[24]

While the absolute differences between RA-based coding and either LLM-based coding or the original coding are small on average, there are categories with substantial differences. In Section 5.2 we examine why the absolute difference is large for one of these cases, *Timing*: Minimum Winning Coalition (MWC, hereafter), and show that this difference can be reduced with better prompt design.[25]

[23] To be precise, the coding for Charness and Dufwenberg (2006) was done by one of the researchers, not RAs.

[24] Each observation corresponds to the absolute difference in average codings for a single category. This is an informal test that should be interpreted cautiously given the high degree of aggregation and the potential lack of independence between observations.

[25] *Legitimacy*: Category 6 also has a large absolute difference between the LLM-based coding and the two

Absolute differences between codings do not capture whether the codings agree at the level of individual communication episodes, since offsetting disagreements at the communication episode level can leave category means unchanged. We therefore turn to Cohen's kappa and Gwet's AC1. The left panel of Table 3 reports the Cohen's kappa, by category, between the LLM-based and RA-based codings and between the original and RA-based codings. The right panel reports equivalent data for Gwet's AC1.[26] The row at the bottom reports the average Cohen's kappa (or Gwet's AC1) across categories, weighting each category equally.

Table 3: Inter-Coder Reliability vs. RA-based Coding by Category and Coder Type

		Cohen's Kappa		Gwet's AC1	
Study	Category	LLM–RA	Original–RA	LLM–RA	Original–RA
Legitimacy	1a	0.908	0.898	0.997	0.997
Legitimacy	1b	0.888	0.888	0.997	0.997
Legitimacy	1c	1.000	1.000	1.000	1.000
Legitimacy	1d	1.000	0.530	1.000	0.993
Legitimacy	1e	0.937	0.805	0.941	0.829
Legitimacy	1f	0.257	0.083	0.896	0.949
Legitimacy	2	0.687	0.755	0.892	0.914
Legitimacy	3	0.854	0.876	0.994	0.996
Legitimacy	4	0.838	0.702	0.879	0.769
Legitimacy	5	0.730	0.705	0.973	0.973
Legitimacy	6	0.401	0.480	0.497	0.735
Promises	Empty Talk	0.846	0.776	0.885	0.824
Promises	No Message	1.000	1.000	1.000	1.000
Promises	Promise	0.865	0.800	0.872	0.806
Timing	All-way split	0.638	0.728	0.882	0.930
Timing	Competition	0.780	0.398	0.985	0.952
Timing	Future Coalition	0.749	0.725	0.917	0.875
Timing	MWC	0.462	0.536	0.589	0.595
*Average**		0.750	0.682	0.894	0.894

*Categories Promises (Empty Talk and No Message) are excluded from this average.

Focusing on Cohen's kappa, there is a high degree of agreement between the RA-based and LLM-based coding. Ignoring the two redundant categories for *Promises*, there is substantial

RA-based codings, but this category has limited economic interest; it is a catch-all category for social banter unrelated to the game.

[26]To calculate Cohen's kappa and Gwet's AC1 between the RA-based and original codings, we use the mean of the pairwise comparisons with each original coder for *Legitimacy*; *Legitimacy* only had two original coders and hence does not have a well-defined mode. For *Promises* and *Timing*, the mode of the original coders is used.

agreement ($\kappa \geq .6$) for 13 of 16 categories and near perfect agreement ($\kappa \geq .8$) for 8 categories. More importantly for our purposes, the agreement between the LLM-based and RA-based codings is better than between the RA-based and original codings; the average Cohen's kappa is higher, there are more categories with substantial agreement, and there are more categories with near perfect agreement.[27] The difference is small and not significant, as shown by a simple paired t-test comparing Cohen's kappa for the LLM-RA pairings and original-RA pairings (t = 1.683; p = .113; d.f. = 15),[28] but the LLM-based coding agrees with the RA-based coding at least as well as the RA-based coding agrees with the original coding. Criterion 1 is again fulfilled.

Using Gwet's AC1, we also conclude that Criterion 1 holds: Comparing the LLM-based coding with the RA-based coding yields the same Gwet's AC1 as comparing the RA-based and original coding. Comparing the RA-based and LLM-based codings, Gwet's AC1 is high for all categories except *Legitimacy:* Category 6 and *Timing:* MWC. Cohen's kappa is low for *Legitimacy* 1f, but Gwet's AC1 is high, suggesting that the low Cohen's kappa is a case of the kappa paradox. The preceding observations reinforce our decision to focus on *Timing:* MWC when exploring how better prompt design improves agreement between the LLM-based and RA-based coding (see fn. 25 for discussion of why we do not focus on *Legitimacy:* Category 6).

Thus far, we have shown reliability by comparing the RA-based coding with the original coding. A potential concern with this approach is that the original coding used different methods (e.g., different instructions for the coders, differing numbers of coders) than the RA-based coding. Our goal is to establish that the LLM-based coding differs from the RA-based coding no more than an independent coding created by RAs would, but the original coding may be *too independent*. Specifically, if the use of different methods depresses agreement between the RA-based coding and the original coding, it weakens the reliability standard for the LLM-based coding.

As an alternative method of establishing reliability that addresses the preceding concern, we compare the Cohen's kappa between a single RA and a single LLM with the Cohen's kappa between two RAs, using the average across all possible pairs in both cases.[29] The

[27]Comparing the RA-based and original codings and ignoring the two redundant categories for *Promises*, there is substantial agreement for 11 of 16 categories and near perfect agreement for 5 categories.

[28]Each observation corresponds to the Cohen's kappa for a single category. This is an informal test that should be interpreted cautiously as noted previously.

[29]There are three possible RA-RA pairs and nine possible RA-LLM pairs. For each LLM, we use its modal coding.

RAs used for this exercise are from the new recoding, not the original coding, and therefore received the same materials and coding instructions as the LLMs. If the LLM-based coding is reliable, the average Cohen's kappa should be similar for RA-LLM pairs and RA-RA pairs. Table E6 in the appendix shows the results of this comparison. Averaging across categories, Cohen's kappa is slightly higher for RA-LLM pairs (.681) than RA-RA pairs (.668), reinforcing our conclusion that LLM-based coding fulfills Criterion 1.

Conclusion 1: The LLM-based coding fulfills Criterion 1 for reliability: The LLM-based and RA-based codings differ no more than the RA-based and original codings differ for either absolute differences between codings or agreement (as measured by Cohen's kappa) between codings.

5.1.2 Reliability of Results

We have thus far shown that the LLM-based coding fulfills Criterion 1 for reliability, but the Monte Carlo exercise developed in Section 3 demonstrates that Criterion 2 can fail when Criterion 1 holds. We therefore turn to Criterion 2. All three papers statistically test how the content of communication affects behavior in the experiment. If the LLM-based coding is reliable, the qualitative results of these analyses should *not* depend on whether we use the RA-based coding or the LLM-based coding.

To check Criterion 2, we attempt to replicate a specific conclusion from each paper that relies on statistical analysis of the coding. Conclusion 5 in *Legitimacy* finds that, "Elected leaders are more likely to send relevant messages, particularly messages suggesting and explaining use of effort level 40 (i.e., maximum effort)." Their Table 6 provides regression analysis backing this conclusion. *Promises* concludes, "... the *IN* rate, the *Roll* rate, and the ex post (*In, Roll*) realizations were much higher following a promise than otherwise." This conclusion is supported by statistical analysis in their Table III. *Timing* states, "We find a strong relationship between communication content and observed bargaining outcomes: notably, MWCs are more likely to be observed when group members discuss MWCs." The upper left panel of their Figure 5 summarizes regression results supporting this conclusion.

Table 4 reports the results of our replication exercise using the RA-based and LLM-based codings. Because the three papers use different types of econometric analysis (2SLS regressions in *Legitimacy*, one-tailed two proportion z-tests in *Promises*, and OLS regressions in *Timing*), we focus on two elements common to all three analyses: the direction of the

effect and the p-values.[30]

Table 4 has 3 panels, one for each paper. The first three columns show p-values from the original paper, the replication with RA-based coding, and the replication with LLM-based coding. The next two columns report whether the signs of the estimates match with different codings (original vs. RA-based; RA-based vs. LLM-based). The final two columns examine whether statistical significance matches with different codings. For this exercise, we define a test result to be statistically significant if $p < 0.1$. The significance matches if a test is significant for both codings *or* not significant for both.

Comparing results for the RA-based and LLM-based codings, there are few differences. The signs of the estimates match in all twelve comparisons. For 10 of 12 comparisons, statistical significance also matches. Results based on the LLM-based coding match those based on the RA-based coding slightly more often than results based on the RA-based coding match those based on the original coding; comparing the RA-based and original codings, signs match for 11 of 12 comparisons and significance matches for 9 of 12 comparisons.

We do not claim that the LLM-based coding perfectly replicates the statistical results based on either the original or RA-based coding, but for most applications, the exact values of the estimates and p-values are not important. What matters is the qualitative relationship between communication content and subjects' choices. Statistical analyses using the LLM-based coding largely support the three conclusions we highlighted from the original papers.

The two exceptions are weak. In *Legitimacy*, elected leaders are not significantly more likely to send relevant messages if we use the LLM-based coding, but the estimate is close to significance (p = .110). The other failure to match significance comes from *Timing*; voter discussion of MWCs does not significantly increase their likelihood if the LLM-based coding is used. This is a boundary case, given that $p = .100$, and comes closer to replicating the conclusion from the original paper than the RA-based coding. In Subsection 5.2.1, we show that both of these exceptions to the original conclusions are eliminated if we use a prompt that gives the LLMs more guidance on how to interpret subjects' messages.

Conclusion 2: The LLM-based coding fulfills Criterion 2: The qualitative conclusions reached from statistical analysis of the LLM-based coding parallel those reached using either the original or the RA-based coding.

[30]See Appendix F for a full reporting of the statistical tests carried out for the replication exercise.

Table 4: Experimental Behavior and Communication Content: Replicating Statistical Analyses with Different Codings — LLM Coding: Baseline

Outcome	***p*-value**			**Match Sign**		**Match Significance**	
	Original	RA-based	LLM-based	Original–RA	RA–LLM	Original–RA	RA–LLM
Panel A: Legitimacy (Brandts et al. (2015))							
Elected Leader (DV: Use 40)	$p = 0.004$	$p = 0.030$	$p = 0.008$	Yes	Yes	Yes	Yes
Elected Leader (DV: Relevant)	$p = 0.050$	$p = 0.037$	$p = 0.110$	Yes	Yes	Yes	No
Panel B: Promises (Charness and Dufwenberg (2006))							
A's In Rate (P vs. NP)	$p<0.001$	$p = 0.003$	$p<0.001$	Yes	Yes	Yes	Yes
B's Roll Rate (P vs. NP)	$p<0.001$	$p<0.001$	$p<0.001$	Yes	Yes	Yes	Yes
(In, Roll) (P vs. NP)	$p<0.001$	$p<0.001$	$p<0.001$	Yes	Yes	Yes	Yes
Panel C: Timing (Baranski and Haas (2023))							
MWC × Proposer	$p = 0.014$	$p = 0.007$	$p = 0.023$	Yes	Yes	Yes	Yes
MWC × Voter	$p = 0.036$	$p = 0.381$	$p = 0.100$	Yes	Yes	No	Yes
3-Way Split × Proposer	$p<0.001$	$p = 0.004$	$p = 0.052$	Yes	Yes	Yes	Yes
3-Way Split × Voter	$p = 0.182$	$p = 0.131$	$p = 0.060$	Yes	Yes	Yes	No
Competition × Proposer	$p = 0.861$	$p = 0.012$	$p = 0.072$	Yes	Yes	No	Yes
Competition × Voter	$p = 0.403$	$p = 0.008$	$p = 0.096$	No	Yes	No	Yes
Future Coalition × Voter	$p = 0.816$	$p = 0.326$	$p = 0.225$	Yes	Yes	Yes	Yes

Panel A (Legitimacy): The p-value is the two-tailed p-value for Elected Leader in the 2SLS regressions of Models 3 and 4 of Table 6.

Panel B (Promises): The p-value comes from a one-tailed two-proportion z-test comparing the rate of each behavioral outcome between Promise-coded and No-Promise-coded episodes. The tests correspond to the third row ("Pooled") of Table III.

Panel C (Timing): The p-value is the two-tailed p-value on the row's independent variable in the OLS regression of the MWC-outcome indicator corresponding to the analysis behind Figure 5.

5.2 Methodological Issues

We have shown LLM-based coding is reliable in the sense defined in the introduction, but there is no reason to believe our baseline methodology is optimal. It uses *vanilla* prompts that are easy to create but agreement between the LLM-based and RA-based codings is poor for several categories. *Timing*: MWC is especially concerning because this category relates to a central concept in *Timing* and poor agreement cannot be explained by the kappa paradox. Another problem is that there are two exceptions to our general finding that conclusions from the original paper can be replicated using LLM-based coding. Can we address these problems with better prompt design? One of the advantages of LLM-based coding is lower costs, but our baseline methodology uses multiple replications of multiple LLMs, increasing the costs. Can we economize by reducing the number of replications and/or LLMs, or will this substantially harm the quality of the coding? This subsection addresses these two questions.

5.2.1 Prompt Design

Our discussion of Tables 2 and 3 noted that the LLM-based coding performs poorly for the category *Timing*: MWC. The absolute difference between the LLM-based and RA-based coding is relatively large for this category and Cohen's kappa between the two codings falls below the 0.600 threshold for substantial agreement. Both criteria for reliability are met by the LLM-based coding using the *Baseline* prompt, but ideally there would not be economically relevant categories where the LLM-based coding does poorly. An additional problem, described in our discussion of Table 4, is the existence of two exceptions to our general finding that conclusions from the original paper can be replicated using LLM-based coding. While the *Baseline* prompt does a good enough job of replicating the original conclusions to be considered reliable, we would prefer to replicate them perfectly.

This subsection studies whether the two preceding problems can be addressed by use of a more sophisticated prompt. We consider three specific changes to the prompt:

- *RA Consensus:* The baseline prompt does not include any examples extracted directly from the episodes to illustrate the categories. The *RA Consensus* prompt adds examples of communication episodes where all three human coders agree. These unambiguous examples are meant to serve as archetypes for the categories.[31]

[31]For a review discussion on prompting techniques see Schulhoff et al. (2024). In the literature, this

- *Meta-Prompt:* Meta-prompting is a technique that involves the LLM in the process of refining its own prompt. An initial prompt provides the LLM with the full experimental instructions and coding instructions (including the categories). Instead of proceeding to coding, the initial prompt instructs the LLM to internalize the materials it has been given and produce a new prompt. This new prompt should "...contain all necessary context about the experiment and coding categories so that a coder (or AI) can understand the task without seeing the original instructions." Intuitively, we give the LLM the *Baseline* prompt and ask it to rewrite the prompt in its own words. The LLM is then given the coding prompt it just created and asked to code the categories.[32]

- *Guidance*: Inspecting the text of communication episodes that are coded differently by the RAs and LLMs in *Timing*, we found that subjects often relied on numeric shorthand or symbolic expressions that carry context-specific meaning within the experiment. For example, "50-50 me you?" is shorthand used by subjects to propose an even split of the pie between two players, implicitly excluding a third. RAs had no trouble interpreting this sort of shorthand terminology, but the LLMs struggled. We therefore modified the baseline prompt to include examples of common shorthand phrases along with guidance on how to interpret them.

  A similar issue occurred for *Legitimacy*. Leaders often request repetition of the previous round's outcome without specifically stating what action should be taken (e.g., "Same as before" or "do it again"). The RAs had no difficulty understanding phrases like these as requests for a specific action, but the LLMs often failed to interpret them correctly.[33] The *Guidance* prompt addresses this problem by instructing the LLMs to use information from past rounds to interpret backwards looking messages.

Table 5 examines whether the first problem raised above, poor agreement between the LLM-based and RA-based coding for *Timing*: MWC, can be ameliorated by the use of alternative prompts. The top panel shows the average coding for the four categories from

technique is often referred to as *few-shot prompting* Zhou et al. (2024); Brown et al. (2020); Schulhoff et al. (2024); Liu et al. (2021).

[32]For each replication, the LLM is asked to create a new prompt and then code. In other words, there are five replications with five new prompts, not five replications using a single new prompt. This approach of using LLMs to generate or refine prompts prior to task execution has been explored in other works as well (Zhou et al., 2024; Liu et al., 2025).

[33]This is partially a technical issue. For coding with the Baseline prompt, the LLM was prompted separately for each communication episode. While easier to implement, this meant that it did not have access to information from previous rounds when coding a communication episode. The Guidance prompt provides the LLM with information about previous communication episodes.

Timing, comparing the RA-based coding with four versions of the LLM-based coding using different prompts: *Baseline*, *RA Consensus*, *Meta-Prompt*, and *Guidance*. The bottom panel shows the Cohen's kappa by category between the RA-based coding and the four versions of the LLM-based coding.

Table 5: Timing, Alternative Prompts

Category	RA-Based	Baseline	RA Consensus	Meta-Prompt	Guidance
Panel A: Average Coding					
All-way split	0.128	0.158	0.137	0.112	0.112
Competition	0.027	0.038	0.038	0.032	0.036
Future Coalition	0.152	0.140	0.131	0.149	0.137
MWC	0.400	0.205	0.169	0.206	0.431
Panel B: Cohen's Kappa vs. RA-based Coding					
All-way split	n/a	0.638	0.722	0.735	0.795
Competition	n/a	0.780	0.826	0.784	0.832
Future Coalition	n/a	0.749	0.730	0.814	0.747
MWC	n/a	0.462	0.420	0.489	0.758

Looking at the first three categories, the average codings are similar for the RA-based coding and all versions of the LLM-based coding. For *Timing*: MWC, the average coding using the *Baseline* prompt is about half of the RA-based coding, as noted in our discussion of Table 2. Matters are little better with the *RA Consensus* and *Meta-Prompt* prompts, but with the *Guidance* prompt the average coding increases to essentially the same level as the RA-based coding.

The story is similar for Cohen's kappa. For the first three categories, it matters little which prompt is used as Cohen's kappa always exceeds the .600 threshold for substantial agreement. For *Timing*: MWC, Cohen's kappa between the *Baseline* coding and the RA-based coding falls well below the .600 threshold. This is true for the *RA Consensus* and *Meta-Prompt* prompts as well. With the *Guidance* prompt, Cohen's kappa easily exceeds the .600 threshold. Modifying the prompt to give LLMs guidance about how to interpret subjects' communication, such as explaining common shorthand phrases, substantially improves the ability of the LLM-based coding to agree with RA-based coding.

To address the second problem from above, the two cases where the results of the original papers are not replicated with LLM-based coding, Table F14 reproduces Table 4 using the *Guidance* prompt rather than the *Baseline* prompt. For the second regression in Panel A, "Elected Leader (DV: Relevant)", the *Baseline* prompt yields $p = 0.110$ compared to

$p = 0.050$ for the original coding. With the *Guidance* prompt this becomes $p = 0.058$, almost identical to the original coding. The story is similar for the other problematic regression, "MWC × Voter". The *Baseline* prompt yields $p = 0.100$ compared to $p = 0.036$ for the original coding. With the *Guidance* prompt this becomes $p = 0.043$, again mirroring the original coding.[34] Giving the LLMs guidance about how to interpret subjects' communication improves the ability of LLM-based coding to replicate results from the original papers.[35]

The positive impact of adding guidance makes sense given how LLMs work. An LLM relies on a training set to interpret text. The language that subjects invent to discuss specific experiments is unlikely to be in an LLM's training set. The *Guidance* prompt corrects this deficit by providing missing terminology and context as part of the prompt, giving the LLM information about how subjects communicate that it otherwise lacks.

Conclusion 3: Use of an alternative prompt that provides the LLM with guidance about how to interpret subjects' communication improves the ability of LLM-based coding to agree with RA-based coding and replicate results from the original papers.

Conclusion 3 implies that LLM-based coding should be supplemented with a limited coding conducted by RAs. To illustrate the preceding point, we have run a Monte Carlo exercise that randomly selects 10% of the observations from *Timing* (with replacement). For each randomly selected subset of the data, we calculate the Cohen's kappa for *Timing:* MWC between the LLM-based and RA-based codings. For 97% of the iterations, the Cohen's kappa is less than 0.600. Even with a partial coding by RAs it is extremely likely that we would identify that *Timing:* MWC is a problematic category where the LLM-based and RA-based codings do not substantially agree. The LLMs' performance can be substantially enhanced with an improved prompt, but it is hard to fine-tune the prompt if you do not know where the LLMs are performing poorly or what sort of language is generating the problem.

5.2.2 Number of LLMs and Replications

Our baseline methodology uses three LLMs with five replications per LLM. The extended theory developed in Appendix B suggests that adding more models has greater value than

[34] Oddly, use of the Guidance prompt yields a *p-value* that is closer to the original coding and farther from the RA-based coding ($p = 0.381$).

[35] We have also recreated Table 4 using the *RA Consensus* and *Meta-Prompt* prompts. The results are mixed. For both prompts, the statistical significance using the original coding is matched for "Elected Leader (DV: Relevant)" but not "MWC × Voter".

adding replications, but how much either matters is ultimately an empirical question. In this subsection we explore the impact of using fewer LLMs and/or replications.

The definition of reliability implies that any methodological change that substantially decreases agreement between the LLM-based and RA-based codings makes the LLM-based coding less likely to be reliable. We therefore focus on the relationship between the number of LLMs and/or replications and the Cohen's kappa between the LLM-based and RA-based coding. Table 6 reports the Cohen's kappa between LLM-based and RA-coding by category using differing numbers of LLMs and replications for the LLM-based coding: Baseline (i.e., three LLMs with five replications per LLM), one LLM with five replications, three LLMs with one replication, and one LLM with one replication. For all cases other than Baseline, we report the average over all possible combinations of LLMs and replications. For example, there are 125 possible combinations of one replication from each of the three LLMs. For each of these combinations, we take the mode of the three codings and calculate the Cohen's kappa with the RA-based coding. The column labeled "3 LLMs/1 Rep" reports the average Cohen's kappa over the 125 possible combinations.[36] The other categories are calculated in an analogous fashion. The final row reports the average Cohen's kappa across categories, weighting each category equally.

Consistent with expectations, increasing either the number of LLMs or the number of replications per LLM improves the average agreement (as measured by Cohen's kappa) with the RA-based coding, but it is striking how small the effects are. The theory implies that increasing the number of LLMs will have a larger effect than increasing the number of replications, and this is indeed true, but the effect of tripling the number of LLMs is underwhelming. The patterns observed in the averages are largely present on a category-by-category basis. Increasing the number of LLMs improves Cohen's kappa for 27 of 32 possible comparisons (ignoring the two redundant categories), versus 22 of 32 with additional replications. In 28 of 32 comparisons, increasing the number of LLMs has a larger effect than increasing the number of replications. The problem is that, limiting ourselves to cases where increasing the number of LLMs improves Cohen's kappa, the average increase is a minuscule .028. The maximum improvement is a modest .107. We anticipated a small effect from increasing the number of replications, but adding LLMs does little better.

Understanding why increasing the number of replications has little effect is straightfor-

[36]The Baseline prompt uses a total of $5 \times 3 = 15$ replications. The replications used to generate Table 6 are drawn from these 15 replications rather than creating new replications.

Table 6: Changing the Number of LLMs or Replications per LLM

Study	Category	Baseline	1 LLM/5 Reps	3 LLMs/1 Rep	1 LLM/1 Rep
Legitimacy	1a	0.908	0.851	0.908	0.851
Legitimacy	1b	0.888	0.925	0.888	0.925
Legitimacy	1c	1.000	1.000	1.000	1.000
Legitimacy	1d	1.000	1.000	1.000	1.000
Legitimacy	1e	0.937	0.939	0.942	0.935
Legitimacy	1f	0.257	0.233	0.238	0.221
Legitimacy	2	0.687	0.709	0.693	0.709
Legitimacy	3	0.854	0.748	0.844	0.744
Legitimacy	4	0.838	0.806	0.823	0.800
Legitimacy	5	0.730	0.708	0.718	0.704
Legitimacy	6	0.401	0.388	0.408	0.384
Promises	Empty Talk	0.846	0.828	0.844	0.827
Promises	No Message	1.000	1.000	1.000	1.000
Promises	Promise	0.865	0.850	0.863	0.848
Timing	All-way split	0.638	0.610	0.638	0.609
Timing	Competition	0.780	0.728	0.794	0.716
Timing	Future Coalition	0.749	0.746	0.751	0.747
Timing	MWC	0.462	0.432	0.442	0.428
*Average**		0.750	0.730	0.747	0.726

*The *Promises:* Empty Talk and *Promises:* No Message categories are excluded from this average.

ward. Averaging across categories and LLMs, the Cohen's kappa between a pair of replications of the same LLM equals .959. There is near perfect agreement between replications of the same LLM. It is unsurprising that adding replications has little effect given that the replications are almost identical to each other.

Similar logic explains why increasing the number of LLMs has a small effect. Averaging across categories and all possible pairs of LLMs, the Cohen's kappa between a pair of replications from *different* LLMs is .821. While less than the Cohen's kappa between two replications from the same LLM, this figure still indicates near perfect agreement and is substantially higher than the agreement between two RAs ($\kappa = .668$) or an RA and a single replication of an LLM ($\kappa = .700$). The different LLMs agree with each other much more than RAs agree with each other or with LLMs. While not as extreme as adding a replication, adding one more LLM amounts to adding a rather similar coding and has little effect.

At a less mechanical level, Appendix B describes two countervailing effects from adding additional LLMs. Having more LLMs reduces the likelihood of disagreement due to coder effects (i.e., biases that are specific to a single LLM) but accentuates systematic differences between LLMs and RAs (i.e., biases that are shared by all LLMs). For our data, it appears

that the second effect is nearly as strong as the first, leading to a minimal overall effect.

Conclusion 4: Adding additional LLMs has a bigger impact on the Cohen's kappa with RA-based coding than adding more replications, but both effects are relatively small.

5.3 When Do the LLMs and RAs Disagree?

Across all studies, using the *Baseline* prompt, the LLM-based and RA-based coding disagree for 730/8998 observations (8.1%). Given that coding is not an exercise where a ground truth can be identified, having some disagreement is not necessarily a major concern. Communication between subjects is often ambiguous and it is not uncommon for RAs to disagree among themselves about how to code a particular communication episode. Nevertheless, studying why LLMs and RAs disagree is a useful diagnostic exercise. If disagreement is largely due to cases where the communication is inherently ambiguous, disagreements reflect the nature of the coding exercise and are not a source of concern about the validity of LLM-based coding. Our concern about the use of LLMs would be much higher if disagreements between the LLMs and RAs stem from cases where the RAs view the coding as unambiguous and the LLMs strongly disagree.

To separate these two possibilities, Table 7 breaks down the data by whether the RAs unanimously agreed with each other and whether the LLMs unanimously agreed. Each cell of the table gives the percentage of observations for which the modal coding for the RAs and LLMs disagreed, with the number of observations in parentheses.

Table 7: When do the RA and LLM Codings Disagree?

	Unanimous LLM	**Heterogeneous LLM**	**Total**
Unanimous Human	2.0% (7,336)	37.5% (515)	4.3% (7,850)
Heterogeneous Human	30.4% (845)	44.4% (302)	43.2% (1147)
Total	4.9% (8,181)	40.0% (817)	8.1% (8,998)

When the RAs and LLMs are both unanimous among themselves, the disagreement rate between RAs and LLMs is low (2.0%). When the coding is ambiguous, meaning that either the RAs or the LLMs did not unanimously agree among themselves, the disagreement rate

increases to 35.1%. Observations where either the LLMs or RAs do not agree unanimously among themselves account for only 15.1% of the data, but include 63.4% of the cases where the modal coding disagrees between the LLMs and RAs. Disagreements between LLMs and RAs largely reflect communication episodes where the content is inherently ambiguous. As such, the presence of disagreement is not a major source of concern about LLM-based coding.

Conclusion 5: Disagreements between the LLM-based and RA-based coding primarily reflect communication episodes where the underlying communication is inherently ambiguous.

6 Conclusion

The primary purpose of this paper is to evaluate the use of LLM-based coding to quantify communication data from economic experiments. We develop a simple baseline prompt and apply it to data from three papers using three LLM models. To evaluate the results of the LLM-based coding we introduce the concept of "reliability". It is unrealistic to expect the LLM-based and RA-based coding to agree perfectly given that RAs do not agree among themselves. Instead, reliability asks whether LLM-based coding is a good substitute for RA-based coding. Is the agreement between LLM-based and RA-based coding as good as the agreement between two independent implementations of coding by RAs? Are the conclusions drawn from the coding the same with LLM-based and RA-based coding? Our first main conclusion is that LLM-based coding is reliable, fulfilling both criteria for reliability. As such, researchers should feel comfortable using LLM-based coding.

Even though the three papers feature differing structures of communication (i.e., one shot vs. repeated, one-way vs. multilateral), the average Cohen's kappa between the LLM-based and RA-based coding is greater than 0.600 for all three papers, the commonly-accepted threshold for substantial agreement.[37] This suggests that the reliability of LLM-based coding is not limited to a narrow class of communication structures.

Beyond evaluating LLM-based coding, we also address several methodological issues related to its implementation. The most important is whether modifying the prompt improves the ability of the LLM-based coding to agree with RA-based coding for categories that yield a low Cohen's kappa and to replicate results based on the original coding. We find that providing the LLMs with additional guidance for interpreting subjects' communication is

[37] Averaging across categories, the Cohen's kappas between the LLM-based and RA-based coding are 0.773, 0.865, and 0.657 for *Legitimacy*, *Promises*, and *Timing* respectively.

quite effective. This leads to our second main conclusion: We strongly recommend the continued use of RA-based coding on a limited basis. Having a random subset of communication episodes coded by RAs is necessary to identify cases where the LLMs perform poorly. We expect LLMs to improve over time and therefore our view is largely positive in line with other related studies, but as of now, their use is best complemented with human coding.

A second methodological point we address is the effect of changing the number of LLMs and/or replications. We develop a simple theory of coding that suggests that adding LLMs will be more valuable than adding replications. The data support this conjecture, but with the caveat that increasing the number of LLMs does not have an enormous effect. The theory provides an explanation for the limited effect of increasing the number of LLMs: Having more LLMs accentuates their shared biases. For our data, it seems that this effect is sufficiently strong to almost entirely eliminate improvement with more LLMs.

More generally, we study when the LLM-based and RA-based coding disagree. Disagreement tends to occur for cases where the LLMs and/or the RAs have difficulty agreeing among themselves. This suggests that disagreements reflect systematic ambiguity in the communication rather than a fundamental difference between coding by LLMs and RAs.

In interpreting our work, a few points should be kept in mind. We use the theory in Section 3 to argue that Criterion 2 need not hold when Criterion 1 holds. Our explanation makes use of a specific feature of the model, the possibility that an LLM is (at least in part) trying to imitate an RA. We feel this approach has conceptual value, but it is not the only way we could get Criterion 2 to fail when Criterion 1 holds. Anything that makes agreement between the LLM-based coding and the ground truth weaker without affecting agreement between the LLM-based and RA-based codings will have similar implications.

Turning to methodological points, we do not claim to have designed an optimal prompt. This is not our intent. Our goal is to show that a vanilla version of LLM-based coding is reliable. We use off-the-shelf LLMs, do little customization (i.e., we use the default settings), and employ a straightforward prompt. Researchers do not need a super-sophisticated understanding of LLMs or computer programming to generate reliable coding.

Along similar lines, we do not attempt to identify the best LLM for coding. The LLMs in this paper were chosen because they were commonly-used models at the time and the three models generate similar performance (i.e., similar agreement with the RA-based coding). If we were restarting this project today, we would likely not choose the same three models. That's part of our point. Presumably the performance of LLMs is improving over time, but

they are already good enough to generate reliable coding. There is no reason to expect that to change as models evolve.

If we reran the LLM-based coding, the results would certainly not be identical due to the stochastic nature of LLMs. The qualitative conclusions would presumably be unaffected (this is at the heart of reliability), but running our baseline prompt is not like running a regression (i.e., running the same code with the same data will not perfectly replicate the original results). It is a more complex problem to generate LLM-based coding with replicability than to generate a reliable LLM-based coding. See Cooper et al. (2026) for development of LLM-based coding methods with a high degree of replicability.

We finish with some practical advice for researchers implementing LLM-based coding: (1) Many researchers are attracted to LLM-based coding as a way of saving time and money, but it remains critical to retain an element of human involvement. Having RAs code a subset of the data is necessary for refining the coding prompt. (2) We strongly recommend that more than one LLM be used to generate coding. Even though the effect of adding LLMs is small in our dataset, there is a positive effect and the theoretical benefits in reducing LLM-specific bias are clear. (3) We see little value in adding replications per LLM, either from a theoretical or empirical point of view.

LLM-based coding is a cost effective method of coding natural language communication in economics experiments. As LLMs improve in quality and performance, we expect that LLMs will be used more widely for coding and hopefully will become more reliable. The concept of reliability that we introduce provides a framework for evaluating the performance of LLM-based coding which has been lacking in the broader literature evaluating LLM performance in annotation tasks. Our hope is that our work increases the use of experiments with natural language communication by making quantification of communication data more economical without sacrificing the quality of the analysis.

REFERENCES

Abbink, K., Dong, L., and Huang, L. (2022). Talking behind your back: Communication and team cooperation. *Management Science*, 68(7):5187–5200.

Adu, P. (2019). *A step-by-step guide to qualitative data coding*. Routledge.

Alberti, F. and Mantilla, C. (2024). A mechanism requesting prices and quantities may

increase the provision of heterogeneous public goods. *Experimental Economics*, 27(1):244–270.

Andersson, O., Galizzi, M. M., Hoppe, T., Kranz, S., Van Der Wiel, K., and Wengström, E. (2010). Persuasion in experimental ultimatum games. *Economics Letters*, 108(1):16–18.

Andreoni, J. and Rao, J. M. (2011). The power of asking: How communication affects selfishness, empathy, and altruism. *Journal of Public Economics*, 95(7-8):513–520.

Antinyan, A., Corazzini, L., D'Agostino, E., and Pavesi, F. (2023). Watch your words: An experimental study on communication and the opportunity cost of delegation. *Journal of Economic Behavior & Organization*, 214:216–232.

Arad, A. and Penczynski, S. P. (2024). Multi-dimensional reasoning in competitive resource allocation games: Evidence from intra-team communication. *Games and Economic Behavior*, 144:355–377.

Arganov, M. and Tergiman, C. (2014). Communication in multilateral bargaining. *Journal of Public Economics*, 118:75–85.

Asirvatham, H., Mokski, E., and Shleifer, A. (2026). Gpt as a measurement tool. Working Paper 34834 `http://www.nber.org/papers/w34834`.

Babin, J. J. and Chauhan, H. S. (2023). Initiating free-flow communication in trust games. *Frontiers in Behavioral Economics*, 2:1120448.

Babutsidze, Z., Hanaki, N., and Zylbersztejn, A. (2021). Nonverbal content and trust: An experiment on digital communication. *Economic Inquiry*, 59(4):1517–1532.

Backstrom, J. D., Eckel, C. C., Rholes, R., and Tangvatcharapong, M. (2025). Behind the screens: A replication and extension of coasian bargaining experiments in the digital age. *European Economic Review*, 175:105024.

Baranski, A. and Cox, C. A. (2023). Communication in multilateral bargaining with joint production. *Experimental Economics*, 26(1):55–77.

Baranski, A. and Haas, N. (2023). The timing of communication and retaliation in bargaining: An experimental study. *Journal of Economic Psychology*, 96:102621.

Baranski, A. and Kagel, J. H. (2015). Communication in legislative bargaining. *Journal of the Economic science Association*, 1(1):59–71.

Baron, D. P., Bowen, T. R., and Nunnari, S. (2017). Durable coalitions and communication: Public versus private negotiations. *Journal of Public Economics*, 156:1–13.

Bastianello, F., Decaire, P., and Guenzel, M. (2024). Mental models and financial forecasts. *Jacobs Levy Equity Management Center for Quantitative Financial Research Paper*.

Bavaresco, A., Bernardi, R., Bertolazzi, L., Elliott, D., Fernández, R., Gatt, A., Ghaleb, E., Giulianelli, M., Hanna, M., Koller, A., Martins, A. F. T., Mondorf, P., Neplenbroek, V., Pezzelle, S., Plank, B., Schlangen, D., Suglia, A., Surikuchi, A. K., Takmaz, E., and Testoni, A. (2025). LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255. Association for Computational Linguistics.

Ben-Ner, A. and Putterman, L. (2009). Trust, communication and contracts: An experiment. *Journal of Economic Behavior & Organization*, 70(1-2):106–121.

Ben-Ner, A., Putterman, L., and Ren, T. (2011). Lavish returns on cheap talk: Two-way communication in trust games. *The Journal of Socio-Economics*, 40(1):1–13.

Bershadskyy, D. and Seidel, A. (2024). Choosing a victim you know: Introducing communication to the mobbing game. *Journal of Behavioral and Experimental Economics*, 112:102265.

Bigoni, M., Potters, J., and Spagnolo, G. (2019). Frequency of interaction, communication and collusion: An experiment. *Economic Theory*, 68(4):827–844.

Bischoff, I. (2007). Institutional choice versus communication in social dilemmas—an experimental approach. *Journal of Economic Behavior & Organization*, 62(1):20–36.

Bjedov, T., Madiès, T., and Villeval, M. C. (2016). Communication and coordination in a two-stage game. *Economic Inquiry*, 54(3):1519–1540.

Bochet, O., Page, T., and Putterman, L. (2006). Communication and punishment in voluntary contribution experiments. *Journal of Economic Behavior & Organization*, 60(1):11–26.

Bojic, L. et al. (2025). Comparing large language models and human annotators in latent content analysis of sentiment, political leaning, emotional intensity and sarcasm. *Scientific Reports*, 15:96508.

Bolton, G. E., Chatterjee, K., and McGinn, K. L. (2003). How communication links influence coalition bargaining: A laboratory investigation. *Management Science*, 49(5):583–598.

Bougheas, S., Nieboer, J., and Sefton, M. (2015). Risk taking and information aggregation in groups. *Journal of Economic Psychology*, 51:34–47.

Boyatzis, R. E. (1998). *Transforming Qualitative Information: Thematic Analysis and Code Development*. SAGE Publications, Thousand Oaks, CA.

Brandts, J. and Cooper, D. J. (2007). It's what you say, not what you pay: An experimental study of manager-employee relationships in overcoming coordination failure. *Journal of the European Economic Association*, 5(6):1223–1268.

Brandts, J. and Cooper, D. J. (2025). Managerial leadership, truth-telling, and efficient coordination. *The Economic Journal*, 135(670):1942–1979.

Brandts, J., Cooper, D. J., and Rott, C. (2019). Communication in laboratory experiments. *Handbook of Research Methods and Applications in Experimental Economics*, 401.

Brandts, J., Cooper, D. J., and Weber, R. A. (2015). Legitimacy, communication, and leadership in the turnaround game. *Management Science*, 61(11):2627–2645.

Brandts, J., Ellman, M., and Charness, G. (2016a). Let's talk: How communication affects contract design. *Journal of the European Economic Association*, 14(4):943–974.

Brandts, J., Gerhards, L., and Mechtenberg, L. (2022). Deliberative structures and their impact on voting under economic conflict. *Experimental Economics*, 25(2):680–705.

Brandts, J., Rott, C., and Solà, C. (2016b). Not just like starting over-Leadership and revivification of cooperation in groups. *Experimental Economics*, 19(4):792–818.

Brookins, P., Lightle, J. P., and Ryvkin, D. (2018). Sorting and communication in weak-link group contests. *Journal of Economic Behavior & Organization*, 152:64–80.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Bruttel, L., Eisenkopf, G., and Nithammer, J. (2025). Pre-election communication in public good games with endogenous leaders. *Economics Letters*, 251:112317.

Bruttel, L. and Stolley, F. (2020). Getting a yes. An experiment on the power of asking. *Journal of Behavioral and Experimental Economics*, 86:101550.

Bruttel, L. and Werner, V. (2024). Does communication increase the precision of beliefs? *Economics Letters*, 244:112032.

Casari, M., Zhang, J., and Jackson, C. (2016). Same process, different outcomes: Group performance in an acquiring a company experiment. *Experimental Economics*, 19(4):764–791.

Cason, T. N. and Gangadharan, L. (2013). Cooperation spillovers and price competition in experimental markets. *Economic Inquiry*, 51(3):1715–1730.

Cason, T. N. and Mui, V.-L. (2015). Rich communication, social motivations, and coordinated resistance against divide-and-conquer: A laboratory investigation. *European Journal of Political Economy*, 37:146–159.

Cason, T. N., Sheremeta, R. M., and Zhang, J. (2012). Communication and efficiency in competitive coordination games. *Games and Economic Behavior*, 76(1):26–43.

Cason, T. N., Sheremeta, R. M., and Zhang, J. (2017). Asymmetric and endogenous within-group communication in competitive coordination games. *Experimental Economics*, 20(4):946–972.

Çelebi, C. and Penczynski, S. (2026). Using large language models for text classification in experimental economics. `https://can-celebi.github.io/llmClassification.pdf`.

Charness, G. and Dufwenberg, M. (2006). Promises and partnership. *Econometrica*, 74(6):1579–1601.

Charness, G. and Dufwenberg, M. (2011). Participation. *American Economic Review*, 101(4):1211–1237.

Charness, G., Oprea, R., and Yuksel, S. (2021). How do people choose between biased information sources? Evidence from a laboratory experiment. *Journal of the European Economic Association*, 19(3):1656–1691.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213.

Cooper, D. J., Kagel, J., Qi, S., and McElvain, C. (2026). A methodological study of coding communication content from experimental economics. Working paper.

Cooper, D. J. and Kagel, J. H. (2005). Are two heads better than one? Team versus individual play in signaling games. *American Economic Review*, 95(3):477–509.

Cooper, D. J., Krajbich, I., and Noussair, C. N. (2019). Choice-process data in experimental economics. *Journal of the Economic Science Association*, 5(1):1–13.

Corgnet, B., DeSantis, M., and Porter, D. (2024). Let's chat... when communication promotes efficiency in experimental asset markets. *Management Science*, 70(10):6550–6568.

Cox, C. A. and Stoddard, B. (2018). Strategic thinking in public goods games with teams. *Journal of Public Economics*, 161:31–43.

Dechenaux, E. and Mago, S. D. (2019). Communication and side payments in a duopoly with private costs: An experiment. *Journal of Economic Behavior & Organization*, 165:157–184.

Deck, C., Servátka, M., and Tucker, S. (2013). An examination of the effect of messages on cooperation under double-blind and single-blind payoff procedures. *Experimental Economics*, 16(4):597–607.

DeScioli, P. and Kimbrough, E. O. (2019). Alliance formation in a side-taking experiment. *Journal of Experimental Political Science*, 6(1):53–70.

Di Bartolomeo, G., Dufwenberg, M., Papa, S., and Passarelli, F. (2023). Promises or agreements? Moral commitments in bilateral communication. *Economics Letters*, 222:110931.

Ding, T. and Schotter, A. (2017). Matching and chatting: An experimental study of the impact of network communication on school-matching mechanisms. *Games and Economic Behavior*, 103:94–115.

Ding, T. and Schotter, A. (2019). Learning and mechanism design: An experimental test of school matching mechanisms with intergenerational advice. *The Economic Journal*, 129(623):2779–2804.

Dugar, S. and Shahriar, Q. (2018). Restricted and free-form cheap-talk and the scope for efficient coordination. *Games and Economic Behavior*, 109:294–310.

Eisenkopf, G. (2014). The impact of management incentives in intergroup contests. *European Economic Review*, 67:42–61.

Eisenkopf, G. (2018). The long-run effects of communication as a conflict resolution mechanism. *Journal of Economic Behavior & Organization*, 154:121–136.

Ellingsen, T. and Johannesson, M. (2004). Promises, threats and fairness. *The Economic Journal*, 114(495):397–420.

Erkut, H. and Reuben, E. (2025). Social networks and organizational helping behavior: Experimental evidence from the helping game. *Journal of Public Economics*, 246:105388.

Feltovich, N. and Swierzbinski, J. (2011). The role of strategic uncertainty in games: An experimental study of cheap talk and contracts in the Nash demand game. *European Economic Review*, 55(4):554–574.

Fischer, C. and Normann, H.-T. (2019). Collusion and bargaining in asymmetric Cournot duopoly—an experiment. *European Economic Review*, 111:360–379.

Fonseca, M. A., Li, Y., and Normann, H.-T. (2018). Why factors facilitating collusion may not predict cartel occurrence—experimental evidence. *Southern Economic Journal*, 85(1):255–275.

Fonseca, M. A. and Normann, H.-T. (2012). Explicit vs. tacit collusion—the impact of communication in oligopoly experiments. *European Economic Review*, 56(8):1759–1772.

Freitag, A., Roux, C., and Thöni, C. (2021). Communication and market sharing: An experiment on the exchange of soft and hard information. *International Economic Review*, 62(1):175–198.

Gangadharan, L., Nikiforakis, N., and Villeval, M. C. (2017). Normative conflict and the limits of self-governance in heterogeneous populations. *European Economic Review*, 100:143–156.

Gantner, A., Horn, K., and Kerschbamer, R. (2019). The role of communication in fair division with subjective claims. *Journal of Economic Behavior & Organization*, 167:72–89.

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3):535–574.

Gibbs, G. R. (2018). *Analyzing Qualitative Data*. SAGE Publications Ltd, London, 2 edition.

Goeree, J. K. and Yariv, L. (2011). An experimental study of collective deliberation. *Econometrica*, 79(3):893–921.

Goeree, J. K. and Zhang, J. (2014). Communication & competition. *Experimental Economics*, 17(3):421–438.

Gomez-Martinez, F., Onderstal, S., and Sonnemans, J. (2016). Firm-specific information and explicit collusion in experimental oligopolies. *European Economic Review*, 82:132–141.

Gorodnichenko, Y., Pham, T., and Talavera, O. (2025). Central bank communication on social media: What, to whom, and how? *Journal of Econometrics*, 249:105869.

Harrington, J. E., Hernan Gonzalez, R., and Kujal, P. (2016). The relative efficacy of price announcements and express communication for collusion: Experimental findings. *Journal of Economic Behavior & Organization*, 128(C):251–264.

He, S., Offerman, T., and van de Ven, J. (2019). The power and limits of sequential communication in coordination games. *Journal of Economic Theory*, 181:238–273.

Heine, F. and Riedl, A. (2026). Let's (not) escalate this! Leadership and communication in a group contest. *European Economic Review*, 181:105161.

Hernandez-Lagos, P. (2019). Cooperative initiative through pre-play communication in simple games. *Journal of Behavioral and Experimental Economics*, 80:108–120.

Houser, D. and Xiao, E. (2011). Classification of natural language messages using a coordination game. *Experimental Economics*, 14(1):1–14.

Hu, Y., Kagel, J., Yang, H., and Zhang, L. (2020). The effects of pre-play communication in a coordination game with incomplete information. *Journal of Economic Behavior & Organization*, 176:403–415.

Isaak, A., Schwieren, C., and Iida, Y. (2022). Reaching agreement on contribution behavior in different cultures-a public goods game with representatives in japan and germany. *Journal of Behavioral and Experimental Economics*, 99:101894.

Ismayilov, H. and Potters, J. (2016). Why do promises affect trustworthiness, or do they? *Experimental Economics*, 19(2):382–393.

Janssen, M. A., Anderies, J. M., and Joshi, S. R. (2011). Coordination and cooperation in asymmetric commons dilemmas. *Experimental Economics*, 14(4):547–566.

Jarke-Neuert, J. (2023). Coordination and cooperation in asymmetric commons dilemmas: A replication study. *Journal of the Economic Science Association*, 9(1):123–135.

Kagel, J. H. (2018). Cooperation through communication: Teams and individuals in finitely repeated prisoners' dilemma games. *Journal of Economic Behavior & Organization*, 146:55–64.

Kingsley, D. C. and Muise, D. (2018). More talk, less need for monitoring: Communication and deterrence in a public good game. *Journal of Experimental Political Science*, 5(2):88–106.

Kittel, B., Luhan, W., and Morton, R. (2014). Communication and voting in multi-party elections: An experimental study. *The Economic Journal*, 124(574):F196–F225.

Kleine, M., Langenbach, P., and Zhurakhovska, L. (2016). Fairness and persuasion: How stakeholder communication affects impartial decision making. *Economics Letters*, 141:173–176.

Koch, C., Nikiforakis, N., and Noussair, C. N. (2021). Covenants before the swords: The limits to efficient cooperation in heterogeneous groups. *Journal of Economic Behavior & Organization*, 188:307–321.

Koukoumelis, A., Levati, M. V., and Weisser, J. (2012). Leading by words: A voluntary contribution experiment with one-way communication. *Journal of Economic Behavior & Organization*, 81(2):379–390.

Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.

Langenbach, P. and Friehe, T. (2023). The willingness to pay for voice in dictator games. *Journal of Behavioral and Experimental Economics*, 107:102117.

Läpple, D., Maertens, A., and Barham, B. L. (2023). Communication and advice-taking: Evidence from a laboratory experiment. *Economics Letters*, 228:111110.

Lee, J. Y. and Hoffman, E. (2025). How much you talk matters: Cheap talk and collusion in a bertrand oligopoly game. *Theory and Decision*, 98(2):277–297.

Lei, V., Masclet, D., and Vesely, F. (2014). Competition vs. communication: An experimental study on restoring trust. *Journal of Economic Behavior & Organization*, 108:94–107.

Leibbrandt, A. and Sääksvuori, L. (2012). Communication in intergroup conflicts. *European Economic Review*, 56(6):1136–1147.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Liu, W., Wang, Y., Li, Y., et al. (2025). A systematic survey of automatic prompt optimization techniques. *Transactions on Machine Learning Research*.

Lohse, T. and Simon, S. A. (2021). Compliance in teams–implications of joint decisions and shared consequences. *Journal of Behavioral and Experimental Economics*, 94:101745.

Martinelli, C. and Palfrey, T. R. (2020). Communication and information in games of collective decision: A survey of experimental results. In *Handbook of Experimental Game Theory*, pages 348–375. Edward Elgar Publishing.

Meub, L. and Proeger, T. (2017). The impact of communication regimes and cognitive abilities on group rationality: Experimental evidence. *Journal of Economic Behavior & Organization*, 135:229–238.

Mohlin, E. and Johannesson, M. (2008). Communication: Content or relationship? *Journal of Economic Behavior & Organization*, 65(3-4):409–419.

Morton, R. B., Ou, K., and Qin, X. (2020). Reducing the detrimental effect of identity voting: An experiment on intergroup coordination in china. *Journal of Economic Behavior & Organization*, 174:320–331.

Movva, R., Koh, P. W., and Pierson, E. (2024). Annotation alignment: Comparing LLM and human annotations of conversational safety. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9048–9062, Miami, Florida, USA. Association for Computational Linguistics.

Nasution, A. H. and Onan, A. (2024). ChatGPT label: Comparing the quality of human-generated and LLM-generated annotations in low-resource language NLP tasks. *IEEE Access*, 12:71876–71900.

Oprea, R., Charness, G., and Friedman, D. (2014). Continuous time and communication in a public-goods experiment. *Journal of Economic Behavior & Organization*, 108:212–223.

Palfrey, T., Rosenthal, H., and Roy, N. (2017). How cheap talk enhances efficiency in threshold public goods games. *Games and Economic Behavior*, 101:234–259.

Penczynski, S. P. (2016). Persuasion: An experimental study of team decision making. *Journal of Economic Psychology*, 56:244–261.

Pronin, K. and Woon, J. (2023). Does allowing private communication lead to less prosocial collective choice? *Social Choice and Welfare*, 60(4):625–645.

Qin, X., Wang, S., and Wu, M. Z. (2024). Is it what you say or how you say it? *Experimental Economics*, 27(4):874–921.

Saldaña, J. (2021). *The Coding Manual for Qualitative Researchers*. SAGE Publications Ltd, London, 4 edition.

Sauermann, J. (2021). The effects of communication on the occurrence of the tyranny of the majority under voting by veto. *Social Choice and Welfare*, 56(1):1–20.

Schulhoff, S., Ilie, M., Balepur, N., et al. (2024). The Prompt report: A systematic survey of prompt engineering techniques.

Sheremeta, R. M. and Zhang, J. (2010). Can groups solve the problem of over-bidding in contests? *Social Choice and Welfare*, 35(2):175–197.

Sheremeta, R. M. and Zhang, J. (2014). Three-player trust game with insider communication. *Economic Inquiry*, 52(2):576–591.

Strømland, E., Tjøtta, S., and Torsvik, G. (2018). Mutual choice of partner and communication in a repeated prisoner's dilemma. *Journal of Behavioral and Experimental Economics*, 75:12–23.

Sutter, M. and Strassmair, C. (2009). Communication, cooperation and collusion in team tournaments—an experimental study. *Games and Economic Behavior*, 66(1):506–525.

van Elten, J. and Penczynski, S. P. (2020). Coordination games with asymmetric payoffs: An experimental study with intra-group communication. *Journal of Economic Behavior & Organization*, 169:158–188.

Vanberg, C. (2008). Why do people keep their promises? An experimental test of two explanations. *Econometrica*, 76(6):1467–1480.

Vossler, C. A., Poe, G. L., Schulze, W. D., and Segerson, K. (2006). Communication and incentive mechanisms based on group performance: An experimental study of nonpoint pollution control. *Economic Inquiry*, 44(4):599–613.

Waichman, I., Requate, T., and Siang, C. K. (2014). Communication in Cournot competition: An experimental study. *Journal of Economic Psychology*, 42:1–16.

Waichman, I. and von Blanckenburg, K. (2020). Is there no "I" in "team"? Interindividual-intergroup discontinuity effect in a Cournot competition experiment. *Journal of Economic Psychology*, 77:102181.

Wang, S. and Houser, D. (2019). Demanding or deferring? An experimental analysis of the economic value of communication with attitude. *Games and Economic Behavior*, 115:381–395.

Woon, J., Jang, M., Pronin, K., and Schiller, J. (2024). Discussion and fairness in a laboratory voting experiment. *Journal of Experimental Political Science*, 11(2):224–236.

Yang, H., Ye, Z., and Zhang, L. (2025). Teams versus individuals in pre-play cheap talk communication. *Journal of Behavioral and Experimental Economics*, 114:102325.

Zhang, J. and Casari, M. (2012). How groups reach agreement in risky choices: An experiment. *Economic Inquiry*, 50(2):502–515.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. (2024). Large language models are human-level prompt engineers.

Appendix

A Experiments with Natural Language Communication

Table A1: Content Analysis of Communication Experiments

Reference	Content Analysis?	Method
Bolton et al. (2003)	Yes	Human
Ellingsen and Johannesson (2004)	Yes	No information
Cooper and Kagel (2005)	Yes	Human
Bochet et al. (2006)	Yes	Human
Charness and Dufwenberg (2006)	Yes	Human
Vossler et al. (2006)	No	
Bischoff (2007)	No	
Brandts and Cooper (2007)	Yes	Human
Vanberg (2008)	Yes	Human
Mohlin and Johannesson (2008)	Yes	Human
Sutter and Strassmair (2009)	Yes	Human
Ben-Ner and Putterman (2009)	No	
Andersson et al. (2010)	Yes	Human
Sheremeta and Zhang (2010)	No	
Houser and Xiao (2011)	Yes	Human
Charness and Dufwenberg (2011)	Yes	Human
Andreoni and Rao (2011)	Yes	Human
Feltovich and Swierzbinski (2011)	No	
Goeree and Yariv (2011)	Yes	Human
Ben-Ner et al. (2011)	Yes	Human
Janssen et al. (2011)	Yes	No information
Zhang and Casari (2012)	Yes	Algorithm
Cason et al. (2012)	Yes	Human

Reference	Content Analysis?	Method
Koukoumelis et al. (2012)	Yes	Human
Fonseca and Normann (2012)	Yes	Human
Leibbrandt and Sääksvuori (2012)	Yes	Human
Deck et al. (2013)	Yes	Human
Cason and Gangadharan (2013)	Yes	Human
Eisenkopf (2014)	Yes	Human
Goeree and Zhang (2014)	Yes	Human
Kittel et al. (2014)	Yes	Human
Lei et al. (2014)	Yes	Human
Oprea et al. (2014)	Yes	Human
Sheremeta and Zhang (2014)	Yes	Human
Waichman et al. (2014)	No	
Arganov and Tergiman (2014)	Yes	Human
Brandts et al. (2015)	Yes	Human
Bougheas et al. (2015)	No	
Baranski and Kagel (2015)	Yes	Human
Cason and Mui (2015)	Yes	Human
Brandts et al. (2016a)	Yes	Human
Brandts et al. (2016b)	Yes	Human
Ismayilov and Potters (2016)	Yes	Human
Casari et al. (2016)	No	
Harrington et al. (2016)	No	
Penczynski (2016)	Yes	Human
Kleine et al. (2016)	No	
Bjedov et al. (2016)	Yes	Human
Gomez-Martinez et al. (2016)	No	
Baron et al. (2017)	Yes	Human
Cason et al. (2017)	Yes	Human
Ding and Schotter (2017)	Yes	Human
Gangadharan et al. (2017)	Yes	Human
Meub and Proeger (2017)	No	

Reference	Content Analysis?	Method
Palfrey et al. (2017)	Yes	Human
Kingsley and Muise (2018)	Yes	Human
Dugar and Shahriar (2018)	Yes	Human
Kagel (2018)	Yes	Human
Eisenkopf (2018)	Yes	Human
Brookins et al. (2018)	Yes	Human
Strømland et al. (2018)	Yes	Human
Fonseca et al. (2018)	Yes	Algorithm
Cox and Stoddard (2018)	Yes	Human
Bigoni et al. (2019)	Yes	Human
DeScioli and Kimbrough (2019)	Yes	Human
Dechenaux and Mago (2019)	Yes	Human
Fischer and Normann (2019)	Yes	Human
Gantner et al. (2019)	Yes	Human
Hernandez-Lagos (2019)	Yes	Human
He et al. (2019)	Yes	Human
Wang and Houser (2019)	Yes	Human
Ding and Schotter (2019)	Yes	Human
Hu et al. (2020)	Yes	Human
van Elten and Penczynski (2020)	Yes	Human
Morton et al. (2020)	Yes	Human
Waichman and von Blanckenburg (2020)	No	
Bruttel and Stolley (2020)	Yes	Human
Sauermann (2021)	Yes	Human
Babutsidze et al. (2021)	Yes	Human
Charness et al. (2021)	Yes	Human
Freitag et al. (2021)	Yes	Human
Koch et al. (2021)	Yes	Human
Lohse and Simon (2021)	Yes	Human
Abbink et al. (2022)	Yes	Human
Brandts et al. (2022)	Yes	Human

Reference	Content Analysis?	Method
Isaak et al. (2022)	Yes	Human
Pronin and Woon (2023)	Yes	Human
Antinyan et al. (2023)	Yes	Human
Baranski and Cox (2023)	Yes	Human
Di Bartolomeo et al. (2023)	No	
Baranski and Haas (2023)	Yes	Human
Jarke-Neuert (2023)	No	
Läpple et al. (2023)	Yes	Human
Langenbach and Friehe (2023)	No	
Babin and Chauhan (2023)	Yes	Human
Woon et al. (2024)	Yes	Key Word search
Alberti and Mantilla (2024)	Yes	Human
Arad and Penczynski (2024)	Yes	Human
Bershadskyy and Seidel (2024)	Yes	Human
Bruttel and Werner (2024)	No	
Corgnet et al. (2024)	Yes	Human
Qin et al. (2024)	Yes	Human
Lee and Hoffman (2025)	Yes	LLM
Bruttel et al. (2025)	Yes	Algorithm
Backstrom et al. (2025)	No	
Erkut and Reuben (2025)	Yes	LLM
Yang et al. (2025)	Yes	Human
Brandts and Cooper (2025)	Yes	Human
Heine and Riedl (2026)	Yes	Human

B Extended Model

This section extends the simple model of coding presented in Section 3. We use the extended model to explore the relative impact of increasing the number of LLMs versus the number of replications.

Let the set of messages to be coded be $\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$. Without loss of generality, assume there is a single coding category. Let E_m be the evidence contained in message $m \in \mathcal{M}$. The "ground truth" coding of m equals 1 if $E_m > 0$ and zero otherwise. Let the distribution of E_m be normal with mean $\bar{E}$ and variance 1.

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_C\}$ be the set of coders and let $\mathcal{R}^c = \{r_1^c, r_2^c, \ldots, r_R^c\}$ be the set of replications (i.e., repeated codings by the same coder) for coder c. Assume the number of replications, R, is identical for all coders. For each message $m \in \mathcal{M}$, the coder $c \in \mathcal{C}$ draws a noisy signal of the evidence for each replication $r \in \mathcal{R}^c$, denoted by:

$$E_{m,c,r} \coloneqq E_m + \mu_c + \nu_{m,c} + \epsilon_{m,c,r}$$

Coder c codes the message as a 1 if $E_{m,c,r} > 0$ and codes 0 otherwise. The error term for $E_{m,c,r}$ has three terms. The first term, μ_c, captures the coder's bias across *all* messages. The second, $\nu_{m,c}$, is the coder's message-specific bias. We refer to these two terms collectively as the "coder's biases". The final term, $\epsilon_{m,c,r}$, is an i.i.d. error term. All three error terms are normally distributed with mean zero and variances given by σ_μ^2, σ_ν^2, and σ_ϵ^2 respectively.

Messages are classified based on the modal coding across all replications of all coders. Because coding is binary, this is equivalent to taking the median of the (unobserved) values of $E_{m,c,r}$ and coding a message as 1 if the median is greater than zero. Holding the number of replications R fixed and assuming the coder biases are independent, the modal coding converges to the ground truth as the number of coders C goes to infinity. It is not true that the modal coding converges to the ground truth as the number of replications R goes to infinity (holding C fixed). Replications from the same coder all share the same bias term: $\mu_c + \nu_{m,c}$. As R goes to infinity, holding C fixed, the median of the latent variable $E_{m,c,r}$ converges to the median of the coders' bias terms rather than $\bar{E}$.

The preceding implies that adding LLMs is more likely to be useful than adding replications per LLM. Suppose there are two types of coders (i.e., RAs and LLMs) and take the modal coding for each type. Assuming independence of coder biases, the coder biases tend

to cancel each other out as the number of coders gets large and the modal coding for each type converges to the ground truth. It follows that agreement between the modal codings increases as the number of coders grows. But replications of an LLM are not independent. As the number of replications grows, the modal coding for each type converges to the type's modal bias. The two modal codings generally won't reflect the ground truth in the limit and may agree with each other less rather than more.

Finally, it is worth that if coder biases are correlated within type, as may be the case when RAs share common training or LLMs share common prompts, the positive effect of increasing the number of coders will be attenuated.

C Encoding Categories

Table C2: Encoding categories for Legitimacy

Code	Description
1a	Suggested effort level — 0
1b	Suggested effort level — 10
1c	Suggested effort level — 20
1d	Suggested effort level — 30
1e	Suggested effort level — 40
1f	Suggested effort level — ambiguous suggestion, positive but not specific
2	Provided an explanation for choosing suggested effort (e.g., make more money, maximize mutual payoffs)
3	Statements about needing to trust each other
4	Positive feedback about previous outcome (e.g. great job)
5	Negative feedback about previous outcome
6	Social banter – friendly chatter not directly related to the game

Table C3: Encoding categories for Promises

Code	Description
Promise	The message explicitly states an intention to choose "Roll" (i.e. to cooperate) if player A chooses "In". This includes direct promises, commitments, or statements of intended action. Examples: "I will roll", "if you choose In, I will roll", "don't worry, I promise to roll."
Empty Talk	The message does not express any promise or intention to Roll. This includes greetings, good luck wishes, jokes, general thoughts, comments irrelevant to the game decision, or messages expressing uncertainty about their intended action.
No Message	No message was sent (blank or opted out). This category applies when Player B had the option to send a message but explicitly declined to do so.

Table C4: Encoding categories for Timing

Code	**Description**
All-way split	whenever a member states that the total fund should be split between all three members, that is, that all members should get something. "I will give everyone something" or members tell the proposer "you should share it with both of us". Calling for a 3-way equal split, $10 for each, is also to be coded here.
Minimum Winning Coalition (MWC)	whenever a proposer mentions that he will only give money to one of the voters. When a voting member explicitly or implicitly tells the proposer that the other member should get zero. Similar phrasing may be used like: "I'm fine with the other person getting 0"; "I don't care if you give money to the other member"; "Let's split us two in half"; "let's go you and I 50-50". "Me 13, you keep the rest".
Future Coalition	when non-proposers attempt to strike a deal of a future coalition. "We should reject and next round we split it in half between of ourselves". "We should reject and figure it out between both of us next round" "The proposer is trying to give us a small share, lets reject and divide ourselves".
Competition	whenever the proposer tells a voter the amount that the other voter is willing to accept. Also marked if the proposer tells a voter that the other voter is willing to accept less, or that she is looking for the cheapest voter. For voters, whenever he or she asks how much the other one is willing to accept and seeks to undercut or match. "I will take less than the other person" or "I will match the other person".

D LLM Encoding Methods and Prompts

This appendix presents the prompts used for LLM coding across all three studies. For each study we include the *baseline*, *RA consensus*, and *meta-prompt* versions; for *Legitimacy* and *Timing*, we additionally include the *guidance* version.

D.1 Study 1: Legitimacy, Communication, and Leadership

D.1.1 Experimental Instructions (Common to All Versions)

STAGE 2

Parts, Rounds, and Firms: Stage II of the experiment will have two parts. In the first part there are 6 rounds and in the second part there are 12 rounds.

For the remainder of this experiment you will be randomly assigned to a firm consisting of five participants. You will be grouped with the same four other participants for all 18 rounds.

The following instructions are for the first part of Stage II – the first six rounds. You will receive instructions about any changes to the rules prior to the start of the second part of Stage II.

Task: There are five employees in each firm. Each round of the experiment can be thought of as a workweek. Each of the five employees spends 40 hours per week at their firm. In each round, there will be a bonus rate for all employees.

After seeing the bonus rate, each employee has to choose how to allocate his or her time between two activities, Activity A and Activity B. Specifically, each employee will be asked to choose how much time to devote to Activity A. The available choices are 0 hours, 10 hours, 20 hours, 30 hours, and 40 hours. That employee's remaining hours will be put towards Activity B. For example, if an employee devotes 30 hours to Activity A, this means that 10 hours will be put towards Activity B. Weekly payoffs for employees depend on a bonus rate and on the number of hours allocated to Activity A by the employees.

Employee Payoffs: The payoff for an employee of the firm is determined in each round by the bonus rate (B), how many hours that employee spends on activity A, and the minimum number of hours employees in his or her firm spend on Activity A. The employee's payoff is reduced by 5 ECUs per hour that he or she spends on Activity A.

The employee also receives the bonus rate multiplied by the minimum number of hours any employee in his or her firm spends on Activity A. Each employee also automatically gets a flat payoff of 200 ECUs in each round.

For example, suppose an employee spends 10 hours on Activity A. Suppose the other three workers in his or her firm spend 20, 40 and 40 hours and the bonus rate equals 8. The minimum hours spent on Activity A is 10 hours. The employee's payoff equals 200 - 5*10 + 8*10 = 230 ECUs.

Firm Managers: In the second part of Stage II (Rounds 7 - 18), there will be a firm manager. The manager will be selected from among the five employees in the firm. Each firm will have five employees who perform the same task as in the first part of Stage II. However, one employee will also now serve as the firm manager. For the remainder of the experiment, one of the five people in your firm will be the manager. The manager will always be the same person.

At the beginning of each round, the manager will be able to type a message to the other employees in his or her firm. Except for the following restrictions, the manager may type whatever he or she wants.

Restrictions on Messages:
1. Please do not identify yourself or send any information that could be used to identify you (e.g. age, race, gender, etc.).
2. Please refrain from using obscene or offensive language.

D.1.2 Coding Instructions - Baseline

All sessions have 18 periods. Subjects are in fixed groups of five playing a weaklink game in each round. Each player chooses an effort in each period from 5 possible choices (0,10,20,30,40). The group outcome is determined by the minimum effort chosen by a group member. The Pareto dominant equilibrium is for everyone to choose 40, but this is risky. The experiment revolves around seeing what gets them to the efficient outcome.

What we need you to do is code the messages that managers sent. Please mark a 1 for any comment that you think fits the category. You can code more than one category per message. Here are categories:

Suggested effort level:

- cat_1a_suggested_effort_0: Suggests choosing 0 hours

- cat_1b_suggested_effort_10: Suggests choosing 10 hours
- cat_1c_suggested_effort_20: Suggests choosing 20 hours
- cat_1d_suggested_effort_30: Suggests choosing 30 hours
- cat_1e_suggested_effort_40: Suggests choosing 40 hours
- cat_1f_ambiguous_suggestion: Ambiguous suggestion - positive about effort but not specific about a number

cat_2_explanation_for_effort: Provided an explanation for choosing suggested effort

cat_3_trust_statements: Statements about needing to trust each other

cat_4_positive_feedback: Positive feedback about previous outcome

cat_5_negative_feedback: Negative feedback about previous outcome

cat_6_social_banter: Social banter – friendly chatter not directly related to the game

D.1.3 Coding Instructions - RA Consensus

All sessions have 18 periods. Subjects are in fixed groups of five playing a weaklink game in each round. Each player chooses an effort in each period from 5 possible choices (0,10,20,30,40). The group outcome is determined by the minimum effort chosen by a group member. The Pareto dominant equilibrium is for everyone to choose 40, but this is risky. The experiment revolves around seeing what gets them to the efficient outcome.

What we need you to do is code the messages that managers sent. Please mark a 1 for any comment that you think fits the category. You can code more than one category per message. Here are categories:

Suggested effort level:

cat_1a_suggested_effort_0: Suggests choosing 0 hours

- Examples of YES (1): "0..im out"
- Examples of NO (0): "if we all pick 40 we make the most"

cat_1b_suggested_effort_10: Suggests choosing 10 hours

- Examples of YES (1): "Everyone select 10 hours for this one!"
- Examples of NO (0): "Keep picking 40!"

cat_1c_suggested_effort_20: Suggests choosing 20 hours

- Examples of YES (1): "let's all choose 20 this round"
- Examples of NO (0): "40 hrs to A"

cat_1d_suggested_effort_30: Suggests choosing 30 hours

- Examples of YES (1): "ok everyone lets try and get the most points and put in a minimum of 30 hours"
- Examples of NO (0): "pick 40 for 400 ECUs"

cat_1e_suggested_effort_40: Suggests choosing 40 hours

- Examples of YES (1): "Keep picking 40!"
- Examples of NO (0): "good job everyone"

cat_1f_ambiguous_suggestion: Ambiguous suggestion - positive about effort but not specific about a number

- Examples of YES (1): "lets keep up the good work!"
- Examples of NO (0): "pick 40"

cat_2_explanation_for_effort: Provided an explanation for choosing suggested effort

- Examples of YES (1): "if we all pick 40 we make the most"
- Examples of NO (0): "pick 40"

cat_3_trust_statements: Statements about needing to trust each other

- Examples of YES (1): "trust me ill lead you to the promise land"
- Examples of NO (0): "good job everyone"

cat_4_positive_feedback: Positive feedback about previous outcome

- Examples of YES (1): "great job, everyone! keep up the good work!!"
- Examples of NO (0): "pick 40"

cat_5_negative_feedback: Negative feedback about previous outcome

- Examples of YES (1): "some one put 30?!?!?!?!?!?! why?????????"

- Examples of NO (0): "good job everyone"

cat_6_social_banter: Social banter – friendly chatter not directly related to the game

- Examples of YES (1): "ya'll check out the new lil wayne album"
- Examples of NO (0): "pick 40"

D.1.4 Coding Instructions - Guidance

All sessions have 18 periods. Subjects are in fixed groups of five playing a weaklink game in each round. Each player chooses an effort in each period from 5 possible choices (0,10,20,30,40). The group outcome is determined by the minimum effort chosen by a group member. The Pareto dominant equilibrium is for everyone to choose 40, but this is risky. The experiment revolves around seeing what gets them to the efficient outcome.

What we need you to do is code the messages that managers sent. Please mark a 1 for any comment that you think fits the category. You can code more than one category per message. Here are categories:

Suggested effort level:

- cat_1a_suggested_effort_0: Suggests choosing 0 hours
- cat_1b_suggested_effort_10: Suggests choosing 10 hours
- cat_1c_suggested_effort_20: Suggests choosing 20 hours
- cat_1d_suggested_effort_30: Suggests choosing 30 hours
- cat_1e_suggested_effort_40: Suggests choosing 40 hours
- cat_1f_ambiguous_suggestion: Ambiguous suggestion - positive about effort but not specific about a number

cat_2_explanation_for_effort: Provided an explanation for choosing suggested effort

cat_3_trust_statements: Statements about needing to trust each other

cat_4_positive_feedback: Positive feedback about previous outcome

cat_5_negative_feedback: Negative feedback about previous outcome

cat_6_social_banter: Social banter – friendly chatter not directly related to the game

CONTEXT: This is Session {session}, Period {period} of the experiment.

Below are ALL the manager messages from different chatgroups in this period. Some messages may reference previous messages (e.g., "do the same as before", "again", "same thing"). When coding such messages, consider what the previous chatgroup messages suggested.

MESSAGES TO CODE:
{messages_block}

For EACH chatgroup, provide the classification. Respond with a JSON object where keys are chatgroup IDs and values are the category codes:

```
{
    "chatgroup_X": {
        "cat_1a_suggested_effort_0": 0,
        "cat_1b_suggested_effort_10": 0,
        "cat_1c_suggested_effort_20": 0,
        "cat_1d_suggested_effort_30": 0,
        "cat_1e_suggested_effort_40": 0,
        "cat_1f_ambiguous_suggestion": 0,
        "cat_2_explanation_for_effort": 0,
        "cat_3_trust_statements": 0,
        "cat_4_positive_feedback": 0,
        "cat_5_negative_feedback": 0,
        "cat_6_social_banter": 0
    }
}
```

IMPORTANT:

- Code EACH chatgroup separately.
- If a message says "again", "same as before", "do the same", etc., look at what previous chatgroup messages in this period suggested and code accordingly.
- Return labels only for the following chatgroups: {[cg for cg, _ in valid_messages]}.

D.1.5 Coding Instructions - Meta-Prompt

You are a research assistant. Below are the experimental instructions and coding instructions for a weak-link coordination game experiment ("Legitimacy, Communication, and Leadership in the Turnaround Game"). Your task is to read and internalize

these instructions, then generate a single reusable prompt template that will be used to code manager messages from this experiment.

=== EXPERIMENTAL INSTRUCTIONS ===

[Experimental Instructions, as shown above.]

=== CODING INSTRUCTIONS ===

[Coding Instructions – Baseline, as shown above.]

=== YOUR TASK ===

Based on the above instructions, generate a single, self-contained prompt template that can be reused to code any manager message from this experiment. The template must:

1. Contain all necessary context about the experiment and coding categories so that a coder (or AI) can understand the task without seeing the original instructions
2. Include these EXACT placeholders (with curly braces) that will be filled in for each message:
   - {`message`} — the actual manager message text to code
   - {`period_context`} — context about which period this message is from
   - {`json_template`} — the JSON structure to fill in with 0/1 values
3. Instruct the coder to return ONLY a JSON object with binary 0 or 1 values for each category
4. Clearly explain ALL 11 coding categories (suggested effort levels 0–40, ambiguous suggestion, explanation for effort, trust statements, positive feedback, negative feedback, social banter)
5. Explain that multiple categories can be coded 1 for a single message
6. Explain the experimental setup: weak-link game, 5 employees per firm, effort choices 0/10/20/30/40, payoffs determined by minimum effort

Return ONLY the prompt template text, nothing else. Do not wrap it in quotes or code blocks.

D.2 Study 2: Promise and Partnership

D.2.1 Experimental Instructions (Common to All Versions)

Promise and Partnership

Thank you for participating in this session. The purpose of this experiment is to study how people make decisions in a particular situation. Feel free to ask us questions as they arise, by raising your hand. Please do not speak to other participants during the experiment.

You will receive $5 for participating in this session. You may also receive additional money, depending on the decisions made (as described below). Upon completion of the session, this additional amount will be paid to you individually and privately.

During the session, you will be paired with another person. However, no participant will ever know the identity of the person with whom he or she is paired.

Decision Tasks

In each pair, one person will have the role of A, and the other will have the role of B. The amount of money you earn depends on the decisions made in your pair.

On the designated decision sheet, each person A will indicate whether he or she wishes to choose IN or OUT.

- If A chooses OUT, A and B each receive $5.

We will collect these sheets after the choices have been indicated. Next, each person B will indicate whether he or she wishes to choose ROLL or DON'T ROLL (a die). Note that B will not know whether A has chosen IN or OUT; however, since B's decision will only make a difference when A has chosen IN, we ask B's to presume (for the purpose of making this decision) that A has chosen IN.

Payoffs Summary

Decision	A Receives	B Receives
A chooses OUT	$5	$5
A chooses IN, B chooses DON'T ROLL	$0	$14
A chooses IN, B chooses ROLL, die = 1	$0	$10
A chooses IN, B chooses ROLL, die = 2, 3, 4, 5, or 6	$12	$10

(All of these amounts are in addition to the $5 show-up fee.)

Message Instructions

*For Message from B Treatment

Prior to the decision by A and B concerning IN or OUT, B has an option to send a message to A. Each B receives a blank sheet, on which a message can be written, if

desired. We will allow time as needed for people to write messages, then these will be collected. Please print clearly if you wish to send a message to A.

Restrictions:

- No one is allowed to identify him or herself by name, number, gender, or appearance.
- The experimenter will monitor the messages. Violations (at experimenter discretion) will result in B receiving only the $5 show-up fee, and the paired A receiving the average amount received by other A's.
- Other than these restrictions, B may say anything that he or she wishes in this message.
- If you wish not to send a message, simply circle the letter B at the top of the sheet.

D.2.2 Coding Instructions - Baseline

General Description of the Task

You will be coding messages sent by participants in the "Promises and Partnership" experiment conducted by Charness & Dufwenberg (2006). The study focuses on the impact of communication on trust and cooperation.

In this experiment, participants played a one-shot trust game where:

- Participants were paired and referred to as Player A (principal) and Player B (agent).
- Player A decides whether to enter a partnership ("In") or opt out ("Out").
- If A chooses "Out", both players A and B receive ($5, $5) or ($7, $7) depending on the treatment.
- If A chooses "In", Player B decides to:
  - "Roll": A six-sided die is rolled with the following payoff possibilities:
    - 5/6 probability → A gets $12, B gets $10
    - 1/6 probability → A gets $0, B gets $10
  - "Don't Roll": A gets $0, B gets $14.

- In some treatments, Player B could send a pre-play message to Player A before decisions were made. These messages were non-binding and free-form.

Your task now is to code each message based on its content to analyze how communication type influences trust and cooperation.

Your Coding Task

You will be shown each message sent by Player B. Classify each message into one of these categories:

1. **Promise (P)**: The message explicitly states an intention to choose "Roll" (i.e. to cooperate) if player A chooses "In". This includes direct promises, commitments, or statements of intended action.
2. **Empty Talk (E)**: The message does not express any promise or intention to Roll. This includes greetings, good luck wishes, jokes, general thoughts, comments irrelevant to the game decision, or messages expressing uncertainty about their intended action.
3. **NO MESSAGE (N)**: No message was sent (blank or opted out). This category applies when Player B had the option to send a message but explicitly declined to do so.

If a message is difficult to classify, use your best judgment based on explicit content.

Overview of The Coding Procedure

Step 1: Read thoroughly the full message (or lack thereof) for each observation.
Step 2: Assign each message to one and only one of the three defined categories (P, E, N).
Step 3: Record the assigned category.

D.2.3 Coding Instructions - RA Consensus

General Description of the Task

You will be coding messages sent by participants in the "Promises and Partnership" experiment conducted by Charness & Dufwenberg (2006). The study focuses on the impact of communication on trust and cooperation.

In this experiment, participants played a one-shot trust game where:

- Participants were paired and referred to as Player A (principal) and Player B (agent).
- Player A decides whether to enter a partnership ("In") or opt out ("Out").
- If A chooses "Out", both players A and B receive ($5, $5) or ($7, $7) depending on the treatment.
- If A chooses "In", Player B decides to:
  - "Roll": A six-sided die is rolled with the following payoff possibilities:
    - 5/6 probability → A gets $12, B gets $10
    - 1/6 probability → A gets $0, B gets $10
  - "Don't Roll": A gets $0, B gets $14.
- In some treatments, Player B could send a pre-play message to Player A before decisions were made. These messages were non-binding and free-form.

Your task now is to code each message based on its content to analyze how communication type influences trust and cooperation.

Your Coding Task

You will be shown each message sent by Player B. Classify each message into one of these categories:

1. **Promise (P)**: The message explicitly states an intention to choose "Roll" (i.e. to cooperate) if player A chooses "In". This includes direct promises, commitments, or statements of intended action.
   - Example of YES (P): "I will choose roll."
   - Example of NO (not P): "Please choose In so we can get paid more."
2. **Empty Talk (E)**: The message does not express any promise or intention to Roll. This includes greetings, good luck wishes, jokes, general thoughts,

comments irrelevant to the game decision, or messages expressing uncertainty about their intended action.

- Example of YES (E): "Please choose In so we can get paid more."
- Example of NO (not E): "I will choose roll."

3. **NO MESSAGE (N)**: No message was sent (blank or opted out). This category applies when Player B had the option to send a message but explicitly declined to do so.
   - Example of YES (N): *BLANK/EMPTY MESSAGE
   - Example of NO (not N): "Hello!"

If a message is difficult to classify, use your best judgment based on explicit content.

Overview of The Coding Procedure

Step 1: Read thoroughly the full message (or lack thereof) for each observation.
Step 2: Assign each message to one and only one of the three defined categories (P, E, N).
Step 3: Record the assigned category.

D.2.4 Coding Instructions - Meta-Prompt

You are a research assistant. Below are the experimental instructions and coding instructions for a trust game experiment ("Promises and Partnerships"). Your task is to read and internalize these instructions, then generate a single reusable prompt template that will be used to classify messages from this experiment.

=== EXPERIMENTAL INSTRUCTIONS ===
[Experimental Instructions, as shown above.]

=== CODING INSTRUCTIONS ===
[Coding Instructions – Baseline, as shown above.]

=== YOUR TASK ===
Based on the above instructions, generate a single, self-contained prompt template that can be reused to classify any message from this experiment. The template must:

1. Contain all necessary context about the experiment and coding categories so that a coder (or AI) can understand the task without seeing the original instructions

2. Include these EXACT placeholders (with curly braces) that will be filled in for each message:
   - {`message`} — the actual message text to classify (or [NO MESSAGE/EMPTY] if blank)
   - {`context`} — additional context about the session/player
3. Instruct the coder to return ONLY a JSON object: {`"classification": "P/E/N"`}
4. Clearly explain the three categories: Promise (P), Empty Talk (E), No Message (N)
5. Explain the experimental setup so the coder understands what "Roll" and "Don't Roll" mean

Return ONLY the prompt template text, nothing else. Do not wrap it in quotes or code blocks.

D.3 Study 3: Timing of Communication

D.3.1 Experimental Instructions (Common to All Versions)

In this experiment you will be part of a group of 3 people. One of you will be asked to propose a distribution of $30 among the members of your group. Group members will be able to communicate with each other through chat screens. Proposals are voted up or down according to the simple majority rule. In case the current proposal is rejected, the members of the same group proceed to another proposal and voting round until one allocation is approved. The details of the experiment follow.

The Details of the Experiment

As expressed above, this experiment involves three main components: (1) chat, (2) proposal, and (3) vote. We proceed to fully explain each of them.

(1) Chat: The computer will randomly choose one of you to be the proposer of a distribution of $30. Before a proposal is made, you will have up to three minutes during which time you can exchange written messages with the other two members of your group. Messages can be sent to each member of your group individually through a private chat screen and also collectively through a public chat screen which the other two members can see. The chat screen will remain

open during both proposing and voting stages of bargaining, explained below. We ask that you please be respectful of others and do not reveal your identity or personal information while chatting.

(2) Proposal: In this stage the proposer submits a division of the $30.

(3) Voting: You will observe how much the proposer assigned to each member of the group. You can then click "accept" or "reject". For approval, the proposal requires a simple majority (at least 2 votes). The proposer will automatically be counted as voting in favor.

If rejected: every member in your group will proceed to stage (2) with a member randomly selected as proposer. Feedback on the previous proposal, the voting result, and who was the proposer will be given to you.

The process repeats itself until an allocation of the total fund is approved.

If approved: the result will be binding. Next, you will then be matched into new groups to repeat the stages (1)-(3). You will participate in a total of 15 periods. In each period, you will be randomly reassigned into a group of 3 people, with your subject number for each period determined randomly as well. Thus, while your subject number will remain the same for all rounds within a given period, it will change across periods: in period 1 you can be subject 3, and in period 2 you can be subject 1.

Your Earnings Only 2 of the 15 periods will be randomly selected to count for payment. Your earnings (E) are then given by the shares you received in those periods plus the show up fee of $10.

Example.

Below, we provide an example for you to understand how the payoffs of the experiment work.

Consider a 3 person group with $30 to divide. The proposer allocates $8 to subject 1, $5 to herself, and $17 to subject 2. If Subject 1 votes in favor and Subject 2 against (the proposer is automatically counted in favor), then the proposal is approved. The payments subjects would receive if this period was selected for payment are the offered shares in the approved proposal. Note however that votes could have been different in which case a new round would take place.

Review of the experiment

1. Everyone is randomly assigned into groups of 3
2. There are $30 to divide.
3. One of you will be randomly chosen as the proposer.
4. You will have up to three minutes to chat while the proposer enters a division of the money and up to three minutes to chat while a voting decision is reached.
5. Be respectful and do not reveal any personal information while chatting.
6. Once a proposal is made, voting will take place. If a majority accepts, the allocation is binding, and you will wait in standby until the other groups in your session decide on an allocation.
7. If a majority rejects, the process repeats itself until a given allocation is accepted.
8. Once an allocation is accepted, you will start a new period with randomly selected members. 2 of the 15 periods of play will be chosen randomly for payment.

D.3.2 Coding Instructions - Baseline

You will be reading over conversations between members of a group that were bargaining to divide a given amount of money among themselves. These group members were participants in an experiment that took place between January and April of 2020 in New York University.

In the experiment, one subject of the group was selected to be the person who would divide $30 among himself and two other members. Subjects had up to 3 minutes to chat with the other two members. Proposers had to choose a distribution of the fund and once they did so, the remaining members would proceed to a vote. In some of our experimental sessions, subjects were allowed to keep chatting during the voting stage, while in other sessions this was not allowed. We also conducted sessions where chat only occurred at the voting stage. If a majority approved the proposal, then the proposal was binding. If not, the process would repeat itself until approval. Thus, potentially a same group could negotiate for several rounds.

A copy of the experimental instructions has been given to you. Please read them thoroughly.

Details of the Coding Task

You will be shown the chat transcripts for different groups. Each chat will have three windows, since each person could communicate privately with another member or publicly with all members.

Under each chat window, there will be several categories that you should mark in case it applies to the conversation. The categories are:

1. **All-way split (All_way_split category)**: whenever a member states that the total fund should be split between all three members, that is, that all members should get something. "I will give everyone something" or members tell the proposer "you should share it with both of us". Calling for a 3-way equal split, $10 for each, is also to be coded here.
2. **Minimum Winning Coalition (MWC category)**: whenever a proposer mentions that he will only give money to one of the voters. When a voting member explicitly or implicitly tells the proposer that the other member should get zero. Similar phrasing may be used like: "I'm fine with the other person getting 0"; "I don't care if you give money to the other member"; "Let's split us two in half"; "let's go you and I 50-50". "Me 13, you keep the rest".
3. **Competition (Compete category)**: whenever the proposer tells a voter the amount that the other voter is willing to accept. Also marked if the proposer tells a voter that the other voter is willing to accept less, or that she is looking for the cheapest voter. For voters, whenever he or she asks how much the other one is willing to accept and seeks to undercut or match. "I will take less than the other person" or "I will match the other person".
4. **Future Coalition (Future_coalition category)**: when non-proposers attempt to strike a deal of a future coalition. "We should reject and next round we split it in half between of ourselves". "We should reject and figure it out between both of us next round" "The proposer is trying to give us a small share, lets reject and divide ourselves".

D.3.3 Coding Instructions - RA Consensus

You will be reading over conversations between members of a group that were bargaining to divide a given amount of money among themselves. These group members were participants in an experiment that took place between January and April of 2020 in New York University.

In the experiment, one subject of the group was selected to be the person who would divides $30 among himself and two other members. Subjects had up to 3 minutes to chat with the other two members. Proposers had to choose a distribution of the fund and once they did so, the remaining members would proceed to a vote. In some of our experimental sessions, subjects were allowed to keep chatting during the voting stage, while in other sessions this was not allowed. We also conducted sessions where chat only occurred at the voting stage. If a majority approved the proposal, then the proposal was binding. If not, the process would repeat itself until approval. Thus, potentially a same group could negotiate for several rounds.

A copy of the experimental instructions has been given to you. Please read them thoroughly.

Details of the Coding Task

You will be shown the chat transcripts for different groups. Each chat will have three windows, since each person could communicate privately with another member or publicly with all members.

Under each chat window, there will be several categories that you should mark in case it applies to the conversation. The categories are:

1. **All-way split (All_way_split category)**: whenever a member states that the total fund should be split between all three members, that is, that all members should get something. "I will give everyone something" or members tell the proposer "you should share it with both of us". Calling for a 3-way equal split, $10 for each, is also to be coded here.
    - Examples of YES (1): [Voter 1] "1/12/17?"
    - Examples of NO (0): [Voter 1] "10 for me, 20 for you"
2. **Minimum Winning Coalition (MWC category)**: whenever a proposer mentions that he will only give money to one of the voters. When a voting member explicitly or implicitly tells the proposer that the other member should get zero. Similar phrasing may be used like: "I'm fine with the other person getting 0"; "I don't care if you give money to the other member"; "Let's split us two in half"; "let's go you and I 50-50". "Me 13, you keep the rest".
    - Examples of YES (1): [Voter 1] "0 - 10 - 20?"
    - Examples of NO (0): [Voter 1] "let's just split equally"

3. **Competition (Compete category)**: whenever the proposer tells a voter the amount that the other voter is willing to accept. Also marked if the proposer tells a voter that the other voter is willing to accept less, or that she is looking for the cheapest voter. For voters, whenever he or she asks how much the other one is willing to accept and seeks to undercut or match. "I will take less than the other person" or "I will match the other person".
    - Examples of YES (1): [Proposer] "the other one is offering 11, can you do 10?"
    - Examples of NO (0): [Proposer] "15 15?"
4. **Future Coalition (Future_coalition category)**: when non-proposers attempt to strike a deal of a future coalition. "We should reject and next round we split it in half between of ourselves". "We should reject and figure it out between both of us next round" "The proposer is trying to give us a small share, lets reject and divide ourselves".
    - Examples of YES (specific dollar amount): [Voter 1] "we should both reject and split the offer between us when its reassigned"
    - Examples of NO (-1): [Voter 1] "10/20/0"

D.3.4 Coding Instructions - Guidance

You will be reading over conversations between members of a group that were bargaining to divide a given amount of money among themselves. These group members were participants in an experiment that took place between January and April of 2020 in New York University.

In the experiment, one subject of the group was selected to be the person who would divides $30 among himself and two other members. Subjects had up to 3 minutes to chat with the other two members. Proposers had to choose a distribution of the fund and once they did so, the remaining members would proceed to a vote. In some of our experimental sessions, subjects were allowed to keep chatting during the voting stage, while in other sessions this was not allowed. We also conducted sessions where chat only occurred at the voting stage. If a majority approved the proposal, then the proposal was binding. If not, the process would repeat itself until approval. Thus, potentially a same group could negotiate for several rounds.

A copy of the experimental instructions has been given to you. Please read them thoroughly.

Details of the Coding Task

You will be shown the chat transcripts for different groups. Each chat will have three windows, since each person could communicate privately with another member or publicly with all members.

Under each chat window, there will be several categories that you should mark in case it applies to the conversation. The categories are:

1. **All-way split (All_way_split category)**: whenever a member states that the total fund should be split between all three members, that is, that all members should get something. "I will give everyone something" or members tell the proposer "you should share it with both of us". Calling for a 3-way equal split, $10 for each, is also to be coded here.
    - Examples of YES (1): [Voter 1] "1/12/17?"
    - Examples of NO (0): [Voter 1] "10 for me, 20 for you"
2. **Minimum Winning Coalition (MWC category)**: whenever a proposer mentions that he will only give money to one of the voters. When a voting member explicitly or implicitly tells the proposer that the other member should get zero. Similar phrasing may be used like: "I'm fine with the other person getting 0"; "I don't care if you give money to the other member"; "Let's split us two in half"; "let's go you and I 50-50". "Me 13, you keep the rest".
    - Examples of YES (1): [Voter 1] "0 - 10 - 20?"
    - Examples of NO (0): [Voter 1] "let's just split equally"
3. **Competition (Compete category)**: whenever the proposer tells a voter the amount that the other voter is willing to accept. Also marked if the proposer tells a voter that the other voter is willing to accept less, or that she is looking for the cheapest voter. For voters, whenever he or she asks how much the other one is willing to accept and seeks to undercut or match. "I will take less than the other person" or "I will match the other person".
    - Examples of YES (1): [Proposer] "the other one is offering 11, can you do 10?"
    - Examples of NO (0): [Proposer] "15 15?"
4. **Future Coalition (Future_coalition category)**: when non-proposers attempt to strike a deal of a future coalition. "We should reject and next round we split it

in half between of ourselves". "We should reject and figure it out between both of us next round" "The proposer is trying to give us a small share, lets reject and divide ourselves".

- Examples of YES (specific dollar amount): [Voter 1] "we should both reject and split the offer between us when its reassigned"
- Examples of NO (-1): [Voter 1] "10/20/0"

How participants write share proposals in chat:

Participants often propose divisions using shorthand numeric formats rather than full sentences. These appear in several common patterns:

- `share1/share2/share3`, e.g., "10/15/5" – three numbers separated by slashes representing a three-way division.
- `share1/share2`, e.g., "12/18" – two numbers separated by a slash representing a two-way split.
- `share1-share2-share3`, e.g., "10-10-10" – three numbers separated by dashes.
- `share1-share2`, e.g., "15-15" – two numbers separated by a dash.
- `share1 share2 share3`, e.g., "5 10 15" – three numbers separated by spaces.
- `share1 share2`, e.g., "15 15" – two numbers separated by spaces.
- "X each" or "X apiece," e.g., "15 each" or "10 apiece" – a single number meaning that amount for each person in the conversation.
- "X for me, Y for you," e.g., "13 for me, 17 for you" – explicit assignment of shares to two people.
- "X/Y/Z?" or "X-Y-Z?" with a question mark — proposing a division as a question.

When you see two numbers that add up to $30, it means only two people are splitting the fund. When you see three numbers that add up to $30, check whether any of them is zero to determine the category.

Clarifications for classifying numeric proposals:

- "$15 each" in any chat box is MWC, because $15 each totals $30 and only two people split the fund.

- "$10 each" in any chat box is All_way_split, because it implies all three members get an equal share: $10 + $10 + $10 = $30.
- Any two numbers that add up to $30 is MWC, (only two people receive money).
- Any three numbers where one is $0 and the rest add up to $30 is MWC, (one member is excluded).
- Any three nonzero numbers that add up to $30 is All_way_split, (all three members get something).
- "Split equally" in a private chatroom is MWC, because only two people are in the private chat, so splitting equally between them excludes the third member.

D.3.5 Coding Instructions - Meta-Prompt

You are a research assistant. Below are the experimental instructions and coding instructions for a bargaining experiment. Your task is to read and internalize these instructions, then generate a single reusable prompt template that will be used to code chat transcripts from this experiment.

=== EXPERIMENTAL INSTRUCTIONS ===

[Experimental Instructions, as shown above.]

=== CODING INSTRUCTIONS ===

[Coding Instructions – Baseline, as shown above.]

=== YOUR TASK ===

Based on the above instructions, generate a single, self-contained prompt template that can be reused to code any chat transcript from this experiment. The template must:

1. Contain all necessary context about the experiment and coding categories so that a coder (or AI) can understand the task without seeing the original instructions
2. Include these EXACT placeholders (with curly braces) that will be filled in for each chat transcript:
   - {`identity_desc`} — description of who is in this chat window
   - {`round_num`} — the negotiation round number
   - {`chat_text`} — the actual chat transcript
   - {`json_template`} — the JSON structure to fill in
3. Instruct the coder to return ONLY a JSON object with binary 0 or 1 values

4. Explain the coding categories clearly: All_way_split, MWC, Compete, Future_coalition
5. Explain the roles: P = Proposer, V1 = Voter 1, V2 = Voter 2
6. Instruct to only code for participants present in the chat window

Return ONLY the prompt template text, nothing else. Do not wrap it in quotes or code blocks.

D.4 System Message (Common to All Studies)

"You are an expert coder for economic experiment messages. Always respond with valid JSON."

D.5 LLM Model Parameters

We used the default inference parameters provided by each model's API.

	GPT-4o	**Gemini 2.0 Flash**	**DeepSeek Chat**
Model ID	`gpt-4o`	`gemini-2.0-flash`	`deepseek-chat`
Temperature	1.0	1.0	1.0
Top-p	1.0	1.0	1.0
Presence penalty	0	0	0
Frequency penalty	0	0	0
Max output tokens	–	–	4096
Context window	∼128k	up to 1M	∼128k

Table D5: Inference parameters for evaluated large language models.

E Alternative Approach to Reliability

Table E6: Cohen's Kappa by Category

Study	Category	1H vs. 1H	1H vs. 1 LLM
Legitimacy	1a	0.938	0.844
Legitimacy	1b	0.925	0.938
Legitimacy	1c	1.000	1.000
Legitimacy	1d	0.665	0.833
Legitimacy	1e	0.823	0.885
Legitimacy	1f	0.138	0.323
Legitimacy	2	0.586	0.660
Legitimacy	3	0.739	0.677
Legitimacy	4	0.656	0.726
Legitimacy	5	0.642	0.623
Legitimacy	6	0.458	0.336
Promises	Empty Talk	0.712	0.767
Promises	No Message	0.964	0.982
Promises	Promise	0.749	0.793
Timing	All-way split	0.667	0.556
Timing	Competition	0.623	0.655
Timing	Future Coalition	0.505	0.639
Timing	MWC	0.575	0.401
*Average**		0.668	0.681

*Promises (Empty Talk and No Message) are excluded from this average.

F Replication of Experimental Behavior Analyses using Different Coding Sources

In this Appendix section, we reproduce in their entirety the main statistical and econometric analyses that the correlate or use communication content as an explanatory variable for behavior in the underlying experimental game of each paper. The table and figure notes provide a detailed explanation of which analyses are being performed and to which original tables and figures they correspond to.

For *Legitimacy* and *Promises*, the original RA analysis corresponds exactly to what is being reported in the articles. For *Timing* we are focusing on the treatment with *Proposer-Only Chat* and conduct the same econometric analysis as the original article which included data from additional treatments.

By and large, we see that directional effects remain when using LLM coding. Changes in significance or direction do not reflect any systematic deviation when using LLM coding as compared to changes in significance or direction with the new human coding. We, therefore, conclude that our LLM coding meets Criterion 2 for reliability.

Table F7: Legitimacy - Original RA

	(1)	(2)	(3)	(4)	(5)
	Model 1	Model 2	Model 3	Model 4	Model 5
IV	No	No	Yes	Yes	Yes
Dependent variable	1e, 2	1e, 2	1e, 2	All relevant	1e, 2
Elected Leader	0.400**	0.361**	0.327***	0.164**	0.341***
	(0.164)	(0.140)	(0.113)	(0.076)	(0.096)
Bonus Increase	0.183	0.141	0.103	0.075	0.091
	(0.164)	(0.134)	(0.113)	(0.080)	(0.104)
Lagged Minimum Effort		0.011**	0.021***	0.017**	0.028**
		(0.005)	(0.006)	(0.008)	(0.014)
Leader's Quiz Score					-0.068
					(0.042)
Leader's Average Effort Stage 2					-0.001
					(0.007)
Observations	120	120	120	120	120

Standard errors in parentheses; values transcribed from Brandts, Cooper & Weber (2014) Table 6.
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table F8: Legitimacy - New RA

	(1)	(2)	(3)	(4)	(5)
	Model 1	Model 2	Model 3	Model 4	Model 5
IV	No	No	Yes	Yes	Yes
Dependent variable	1e, 2	1e, 2	1e, 2	All relevant	1e, 2
Elected Leader	0.333**	0.309**	0.331**	0.166**	0.306**
	(0.151)	(0.137)	(0.152)	(0.079)	(0.121)
Bonus Increase	0.300*	0.273**	0.298**	0.093	0.187
	(0.151)	(0.132)	(0.150)	(0.083)	(0.138)
Lagged Minimum Effort		0.007	0.001	0.019**	0.033
		(0.004)	(0.012)	(0.008)	(0.022)
Leader's Quiz Score					-0.069
					(0.048)
Leader's Average Effort Stage 2					-0.012
					(0.008)
R-squared	0.119	0.133	0.121	0.052	
Observations	120	120	120	120	120

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table F9: Legitimacy - LLM Baseline

	(1)	(2)	(3)	(4)	(5)
	Model 1	Model 2	Model 3	Model 4	Model 5
IV	No	No	Yes	Yes	Yes
Dependent variable	1e, 2	1e, 2	1e, 2	All relevant	1e, 2
Elected Leader	0.350**	0.332**	0.266***	0.131	0.293***
	(0.143)	(0.134)	(0.100)	(0.082)	(0.098)
Bonus Increase	0.317**	0.297**	0.224**	0.058	0.248**
	(0.143)	(0.129)	(0.102)	(0.086)	(0.108)
Lagged Minimum Effort		0.005	0.024***	0.020**	0.022
		(0.004)	(0.008)	(0.008)	(0.015)
Leader's Quiz Score					-0.059
					(0.043)
Leader's Average Effort Stage 2					0.003
					(0.007)
R-squared	0.154	0.163	0.050	0.052	0.078
Observations	120	120	120	120	120

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table F10: Legitimacy - LLM Guidance

	(1)	(2)	(3)	(4)	(5)
	Model 1	Model 2	Model 3	Model 4	Model 5
IV	No	No	Yes	Yes	Yes
Dependent variable	1e, 2	1e, 2	1e, 2	All relevant	1e, 2
Elected Leader	0.400**	0.372**	0.319***	0.152*	0.343***
	(0.160)	(0.144)	(0.116)	(0.080)	(0.122)
Bonus Increase	0.267	0.236*	0.178	0.079	0.216
	(0.160)	(0.138)	(0.118)	(0.084)	(0.140)
Lagged Minimum Effort		0.008	0.023**	0.019**	0.017
		(0.005)	(0.011)	(0.008)	(0.020)
Leader's Quiz Score					-0.044
					(0.052)
Leader's Average Effort Stage 2					0.006
					(0.008)
R-squared	0.148	0.167	0.099	0.078	0.150
Observations	120	120	120	120	120

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table F11: Promises - Original RA

	A's In Rate			B's Roll Rate			(In, Roll)		
Treatment	P	NP	Z Stat	P	NP	Z Stat	P	NP	Z Stat
(5, 5)	22/24	9/18	3.04***	18/24	10/18	1.32*	16/24	5/18	2.49***
B messages	(92%)	(50%)		(75%)	(56%)		(67%)	(28%)	
(7, 7)	16/24	7/25	2.71***	20/24	4/25	4.71***	14/24	1/25	4.13***
B messages	(67%)	(28%)		(83%)	(16%)		(58%)	(4%)	
Pooled	38/48	16/43	4.07***	38/48	14/43	4.49***	30/48	6/43	4.73***
	(79%)	(37%)		(79%)	(33%)		(63%)	(14%)	

P/NP = promise / no promise. Z stat is a one-tailed test of equal proportions. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table F12: Promises - New RA

	A's In Rate			B's Roll Rate			(In, Roll)		
Treatment	P	NP	Z Stat	P	NP	Z Stat	P	NP	Z Stat
(5, 5)	22/26	9/16	2.03**	19/26	9/16	1.12	16/26	5/16	1.91**
B messages	(85%)	(56%)		(73%)	(56%)		(62%)	(31%)	
(7, 7)	17/29	6/20	1.97**	22/29	2/20	4.53***	14/29	1/20	3.23***
B messages	(59%)	(30%)		(76%)	(10%)		(48%)	(5%)	
Pooled	39/55	15/36	2.78***	41/55	11/36	4.15***	30/55	6/36	3.61***
	(71%)	(42%)		(75%)	(31%)		(55%)	(17%)	

P/NP = promise / no promise. Z stat is a one-tailed test of equal proportions. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table F13: Promises - LLM Baseline

	A's In Rate			B's Roll Rate			(In, Roll)		
Treatment	P	NP	Z Stat	P	NP	Z Stat	P	NP	Z Stat
(5, 5)	23/25	8/17	3.25***	19/25	9/17	1.56*	17/25	4/17	2.83***
B messages	(92%)	(47%)		(76%)	(53%)		(68%)	(24%)	
(7, 7)	16/26	7/23	2.18**	21/26	3/23	4.73***	14/26	1/23	3.75***
B messages	(62%)	(30%)		(81%)	(13%)		(54%)	(4%)	
Pooled	39/51	15/40	3.76***	40/51	12/40	4.63***	31/51	5/40	4.68***
	(76%)	(38%)		(78%)	(30%)		(61%)	(13%)	

P/NP = promise / no promise. Z stat is a one-tailed test of equal proportions. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

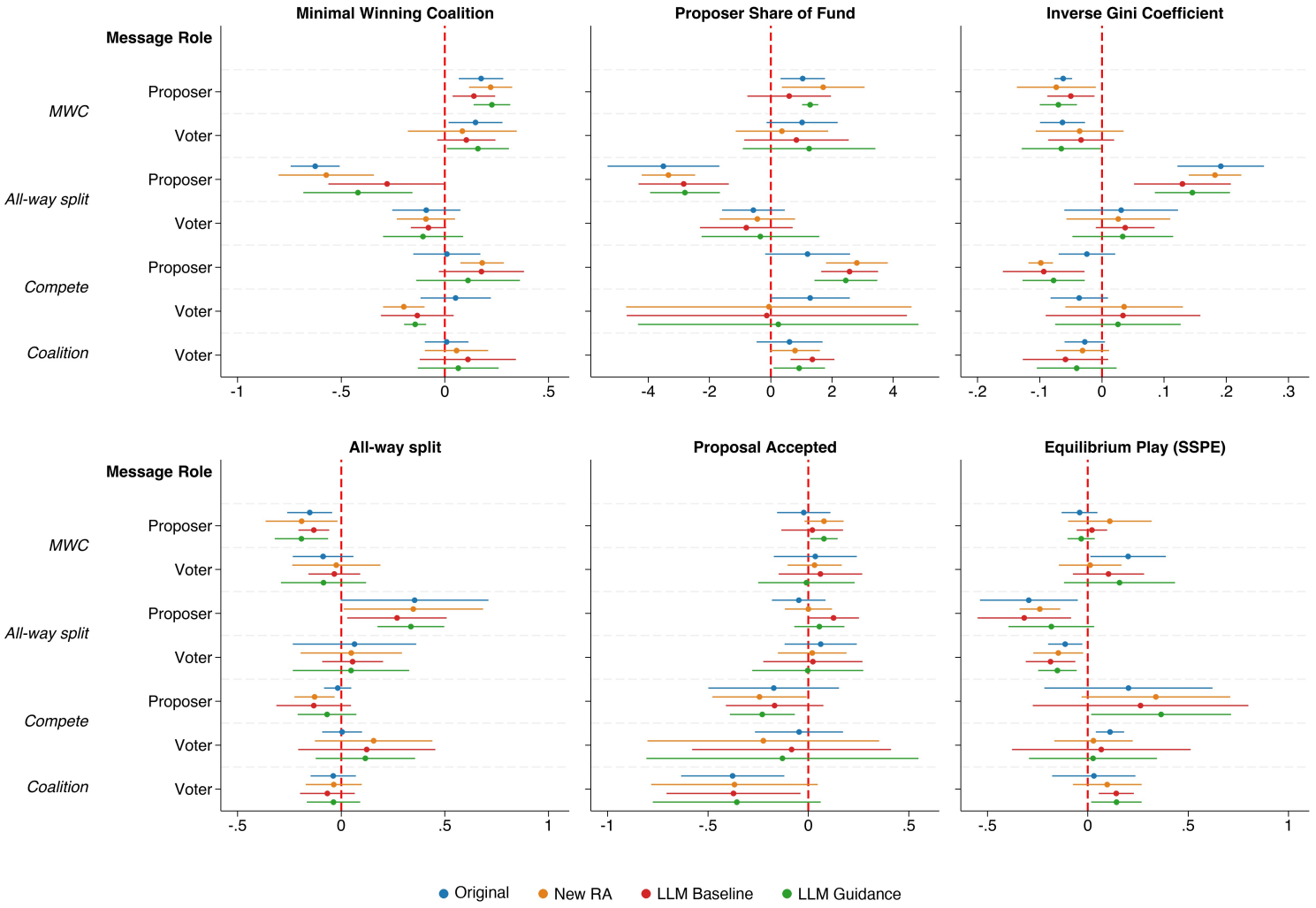

Figure F1: *Notes:* This figure displays the relationship between whether a category of communication content was coded by a majority of coders as having occurred, and bargaining outcomes (MWC, the proposer's share of the fund, equality as measured by the inverse Gini coefficient, occurrence of three-way splits, whether a proposal was accepted, and whether there was a weak measure of equilibrium play) in a given round of play. An MWC is a split where two players share the money and one gets nothing. We distinguish between whether a proposer or voter was coded as having sent a message on a given topic for all topics with the exception of future coalitions, which was only observed for voters. We cluster standard errors at the subject and session levels; communication content refers to the proposal stage only for all outcomes with the exception of future coalitions. We control for treatment assignment. Coefficients are shown for three sets of coders — Original (the paper's original coding), New RA (three newly recruited research assistant coders), and LLM Guidance.

Table F14: Experimental Behavior and Communication Content: Replicating Statistical Analyses with Different Codings —– LLM Coding: Guidance

Outcome	***p*-value**			**Match Sign**		**Match Significance**	
	Original	RA-based	LLM-based	Original–RA	RA–LLM	Original–RA	RA–LLM
Panel A: Legitimacy (Brandts et al. (2015))							
Elected Leader (DV: Use 40)	$p = 0.004$	$p = 0.030$	$p = 0.006$	Yes	Yes	Yes	Yes
Elected Leader (DV: Relevant)	$p = 0.050$	$p = 0.037$	$p = 0.058$	Yes	Yes	Yes	Yes
Panel B: Promises (Charness and Dufwenberg (2006))							
A's In Rate (P vs. NP)	$p<0.001$	$p = 0.003$	$p<0.001$	Yes	Yes	Yes	Yes
B's Roll Rate (P vs. NP)	$p<0.001$	$p<0.001$	$p<0.001$	Yes	Yes	Yes	Yes
(In, Roll) (P vs. NP)	$p<0.001$	$p<0.001$	$p<0.001$	Yes	Yes	Yes	Yes
Panel C: Timing (Baranski and Haas (2023))							
MWC × Proposer	$p = 0.014$	$p = 0.007$	$p = 0.004$	Yes	Yes	Yes	Yes
MWC × Voter	$p = 0.036$	$p = 0.381$	$p = 0.043$	Yes	Yes	No	No
3-Way Split × Proposer	$p<0.001$	$p = 0.004$	$p = 0.015$	Yes	Yes	Yes	Yes
3-Way Split × Voter	$p = 0.182$	$p = 0.131$	$p = 0.181$	Yes	Yes	Yes	Yes
Competition × Proposer	$p = 0.861$	$p = 0.012$	$p = 0.251$	Yes	Yes	No	No
Competition × Voter	$p = 0.403$	$p = 0.008$	$p = 0.003$	No	Yes	No	Yes
Future Coalition × Voter	$p = 0.816$	$p = 0.326$	$p = 0.367$	Yes	Yes	Yes	Yes

Notes: We use the *guidance* LLM prompt for this table.

Panel A (Legitimacy): The p-value is the two-tailed p-value for Elected Leader in the 2SLS regressions of Models 3 and 4 of Table 6.

Panel B (Promises): The p-value comes from a one-tailed two-proportion z-test comparing the rate of each behavioral outcome between Promise-coded and No-Promise-coded episodes. The tests correspond to the third row ("Pooled") of Table III.

Panel C (Timing): The p-value is the two-tailed p-value on the row's independent variable in the OLS regression of the MWC-outcome indicator corresponding to the analysis behind Figure 5.