Imperative Verbs and Instruction Compliance in Large Language Models: A Deep Dive

Preamble

Introduction

The capacity of Large Language Models (LLMs) to comprehend and act upon instructions conveyed through natural language is a cornerstone of their expanding utility across diverse applications. Within this interaction, imperative verbs—those that issue commands or directives—serve as primary linguistic tools for guiding LLM behavior. However, empirical observations reveal a significant variance in instruction compliance contingent upon the specific imperative verb employed and the broader structure of the prompt. Some verbs elicit immediate, structured, and accurate responses, effectively binding the model to the given directive. Others, particularly those with softer or more abstract semantics, may result in vague, meandering, or altogether unexecuted responses.

This report presents an in-depth investigation into the complex mechanisms governing how LLMs interpret and respond to imperative verbs. It explores the linguistic underpinnings of imperative structures, examines the internal model behaviors influencing adherence, analyzes the impact of training data, compares behaviors across prominent LLM architectures, and delves into the practicalities of prompt engineering for optimal instruction compliance. Understanding these multifaceted interactions is paramount for advancing the precision, reliability, and controllability of LLM systems, ultimately enabling more effective human-AI collaboration.

Key Terminology

Term	Definition
Imperative Verb	A command verb intended to trigger direct action, e.g., “write,” “list,” “create.”
Binding Prompt	A prompt that compels a model to follow rigid structure or behavior.
Soft Prompt	A prompt inviting reflection, ambiguity, or exploratory generation.
Instruction-Following Objective	Training alignment goal that makes the model more likely to obey human directives.

I. Core Linguistic and NLP Foundations

1. Imperative Verbs: Linguistic Definition and NLP/LLM Parsing

Linguistic Definition of Imperative Verbs

Imperative verbs are a fundamental grammatical category used to express direct commands, issue requests, or provide instructions.[1] Linguistically, they are characterized by the use of the base form of the verb (the infinitive without "to"), such as "run," "analyze," or "create".[2] A defining feature of imperative sentences is the typically omitted subject, which is implicitly understood to be "you" (the second person, singular or plural).[1] For instance, in "Generate a report," the verb "Generate" is in its base form, and the subject "you" is implied. This structure contributes to the direct and concise nature of commands.[2]

Imperative verbs can manifest in several forms, including:

- **Base Form:** The most common form, e.g., "Listen carefully".[2]
- **First-Person Plural:** Used for group encouragement or instruction, e.g., "Let's analyze the data".[2]
- **Negative Form:** Expresses prohibitions, typically using "do not" or "don't," e.g., "Don't proceed without confirmation".[2]
- **Polite Form:** Softens the command, often by adding "please," e.g., "Please provide the details".[2]

The primary function of imperative verbs is to compel action, conveying a sense of urgency or necessity, which is why they are sometimes colloquially referred to as "command words" or "bossy verbs".[2]

Parsing Imperative Verbs in Traditional NLP Systems

Traditional Natural Language Processing (NLP) systems employ several techniques to parse and identify imperative sentences:

- **Part-of-Speech (POS) Tagging:** A common heuristic involves identifying a sentence as imperative if its first token is a verb (tagged as VB or VBP in the Penn Treebank tagset, for example) and if the sentence concludes with appropriate punctuation like a period or an exclamation mark.[4] While straightforward, this method is not foolproof and can misclassify sentences, as noted in.[4] For instance, a sentence like "You should go" is not imperative despite its suggestive nature and might be misidentified or missed by simpler heuristics.
- **Constituency Parsing:** This method breaks sentences down into their constituent phrases, such as Noun Phrases (NP) and Verb Phrases (VP), often using Context-Free Grammars (CFGs).[5] An imperative sentence like "Stop the car!" typically features the verb ("Stop") as the main verb phrase, with the direct object ("the car") as a noun phrase. The standard Subject-Verb-Object (SVO) order is often reduced to Verb-Object (VO) due to the implied subject.[5]
- **Dependency Parsing:** This approach focuses on the grammatical relationships between words, identifying a "head" word and its "dependents." In imperative

sentences, the main verb usually serves as the root of the dependency tree.[5] This method is generally more robust in handling the structural variations of imperative sentences, including the absent subject, compared to some constituency parsing approaches. Tools like the Stanford Parser are capable of generating such dependency parses.[5]

Parsing Imperative Verbs in LLMs

Large Language Models, particularly those based on the Transformer architecture, do not parse sentences using the explicit, rule-based symbolic methods of traditional NLP.[7] Instead, their "understanding" of imperative verbs and sentence structures emerges from patterns learned during extensive pre-training on vast text corpora and subsequent fine-tuning phases.

- **Implicit Grammatical Understanding:** LLMs develop an implicit understanding of grammar, including the structure of imperative sentences (verb-initial, implied subject), through exposure to countless examples in their training data. This pattern recognition is then heavily reinforced during instruction tuning, where models are specifically trained to follow commands. The frequent co-occurrence of imperative structures with desired task completions in instruction datasets creates a strong association: the model learns that encountering such a pattern typically requires an action-oriented response. This suggests that an LLM's ability to "parse" imperatives is less about applying formal linguistic rules and more about matching input patterns to learned input-output correlations.
- **Attention Mechanisms:** The self-attention mechanism within Transformer models is crucial.[8] It allows the model to dynamically weigh the importance of different tokens in the input prompt when generating a response. It is highly probable that imperative verbs, particularly when positioned at the beginning of a prompt, receive significant attention weights.[11] This high attention signals their critical role in defining the task the model is expected to perform. Research such as "Spotlight Your Instructions" [11] has demonstrated that artificially boosting attention on instruction tokens can improve compliance, implying that the model's natural attention patterns are key to instruction following.
- **Instruction Tuning:** The "instruction-following objective" is a core goal of LLM alignment.[15] LLMs are fine-tuned on datasets replete with examples of instructions (often imperative) and their desired outputs. This process explicitly trains the model to recognize imperative verbs as triggers for action. The model learns to associate the linguistic form of an imperative with the computational behavior of executing a task.
- **Semantic and Pragmatic Focus:** Unlike traditional NLP systems that might prioritize syntactic correctness, instruction-tuned LLMs are geared towards inferring user *intent*—a pragmatic aspect of language. An imperative verb serves as a strong linguistic cue for this intent, signaling a desire for the LLM to perform a specific action. The model's "parsing," therefore, is less about constructing a formal syntactic tree and more about identifying actionable components (the verb, its arguments, contextual constraints) to guide generation towards fulfilling the perceived user goal. This

pragmatic orientation allows for flexibility but also makes the model susceptible to misinterpretation if the prompt's intent is not conveyed clearly.

2. Treatment of Imperative vs. Declarative vs. Interrogative Prompts by LLMs

LLMs exhibit distinct behaviors when processing imperative, declarative, and interrogative prompt structures, reflecting the different communicative functions these sentence moods serve. This differentiation is largely a product of their training data and the specific objectives they are optimized for, such as instruction following or conversational interaction.

Imperative Prompts:

- **Primary LLM Goal/Action:** Execute a command, perform a task, generate a specific output.
- **Typical Output Nature:** Action-oriented, often structured (if specified), direct fulfillment of the command.
- **Key Processing Characteristics:**
    - Strongly trigger the instruction-following mechanisms developed during fine-tuning.[7]
    - In tasks like 3D scene generation, an imperative paradigm involves the LLM iteratively placing objects based on sequential commands (e.g., "place a cube at X, Y, Z"), which can be simpler for the LLM to process for complex scenes compared to a declarative approach that requires resolving a set of relational constraints.[18] This suggests LLMs might be more adept at sequential execution than complex constraint satisfaction.
    - Prompt engineering techniques, including the use of output parsers, can guide LLMs to produce consistently formatted responses to imperative commands.[19]

Declarative Prompts:

- **Primary LLM Goal/Action:** Absorb information, update contextual understanding, use provided facts.
- **Typical Output Nature:** May not produce a direct output unless followed by an imperative or interrogative. If generating, the output will be influenced by the declared information.
- **Key Processing Characteristics:**
    - Serve as context-setting or information provision. The LLM incorporates the declared statements into its current understanding of the input, which then influences subsequent token predictions.[18]
    - In the 3D scene generation context, the declarative paradigm involves the LLM synthesizing a set of *relations* between objects (e.g., "the lamp is on the table") rather than explicit coordinates. The rationale is that reasoning about such relationships might be easier for an LLM than precise numerical values.[18] However, this can become challenging for highly structured or large scenes due to the complexity of the declarative Domain Specific Language (DSL) and the associated solver module.[18]

Interrogative Prompts:

- **Primary LLM Goal/Action:** Answer a question, provide information, explain a concept.
- **Typical Output Nature:** Informative, explanatory, direct answer to the query.
- **Key Processing Characteristics:**
    - Activate knowledge retrieval pathways, drawing upon the vast information learned during pre-training.[7]
    - Engage inferential reasoning processes to formulate an answer based on the retrieved knowledge and the context of the question.
    - The structure of the output is typically an answer or an explanation, rather than a generative action like creating a list or a story, unless the question itself is a request for such generation (e.g., "Can you write a story about...?").

The fundamental differences in LLM responses suggest that sentence mood acts as an initial, high-level dispatcher, directing the model towards a general mode of operation: "execute task" for imperatives, "absorb information" for declaratives, or "answer question" for interrogatives. This initial dispatch is likely learned from the distribution of these sentence types in the massive pre-training corpora and then significantly reinforced by task-specific fine-tuning, such as instruction tuning for imperatives or Q&A dataset training for interrogatives.

The preference for imperative over declarative paradigms in certain complex generation tasks, like the 3D scene generation studied by Gumin et al. [18], may indicate that current LLMs find it easier to follow a sequence of direct commands rather than to interpret and satisfy a complex set of declarative constraints simultaneously. This aligns with the effectiveness of prompt engineering techniques like chain-of-thought, which often break down complex problems into a sequence of simpler steps, often phrased imperatively. This suggests a potential strength in LLMs for sequential processing guided by direct commands, especially when the desired output is complex or requires multiple steps.

The following table summarizes these distinctions:

Table I.1: Comparative LLM Treatment of Prompt Moods

Prompt Mood	Primary LLM Goal/Action	Typical Output Nature	Key Processing Characteristics (Evidence-Based & Speculative)	Supporting Snippets
Imperative	Execute command, perform task, generate output	Action-oriented, often structured (if specified)	Triggers instruction-following; suited for sequential execution; high attention to verb.	[18]
Declarative	Absorb information, update context	Contextual influence on subsequent	Updates LLM's current knowledge state; defines	[18]

		generation; may not produce immediate output	relationships/facts for later use.	
Interrogative	Answer question, provide information/explanation	Informative, explanatory	Activates knowledge retrieval and inferential reasoning.	[7]

II. Model Behavior and Instruction Adherence

3. High-Compliance Verbs: "Generate," "List," "Write," "Create"

Verbs such as "generate," "list," "write," and "create" typically elicit high-compliance, structured outputs from LLMs. This phenomenon can be attributed to several interconnected factors rooted in their linguistic properties, the nature of LLM training, and the models' core functionalities.

Influence of Instruction Tuning and Semantic Concreteness:

A primary reason for high compliance is the prevalence and role of these verbs in instruction-tuning datasets. LLMs undergo extensive fine-tuning on datasets composed of (instruction, output) pairs, where instructions frequently employ these specific verbs to request particular actions or outputs.15 For example, the AlpacaFarm project identified "write," "make," "list," and "create" among the most common root verbs in their instruction dataset.21 Similarly, safety-tuning processes involve transforming questions into imperative instructions like "List reasons..." or "Describe methods...".22 This repeated exposure during training creates a strong association in the model between these verbs and the execution of specific generative tasks.

Furthermore, these verbs possess a high degree of **semantic concreteness** when used in prompts.

- "**List**" clearly implies an enumeration of items, often expected in a bulleted or numbered format.
- "**Write**" (e.g., "write an essay," "write a poem," "write code") suggests the generation of coherent, structured text in a particular genre or language.
- "**Create**" (e.g., "create a plan," "create a character profile") implies the generation of a new artifact or a structured set of ideas.
- "**Generate**" is a versatile high-compliance verb often used to request a wide array of outputs, from text to data structures (e.g., "generate a JSON object").

This inherent clarity reduces ambiguity for the LLM, making the desired output type and structure more discernible. The WETT benchmark, which evaluates LLM writing instruction following, notes that models are generally proficient at adhering to instructions for *writing something new*, including aspects like formatting and keyword usage [23], aligning with the effective nature of these verbs.

The high signal-to-noise ratio of these verbs in conveying actionable intent is crucial. Their meanings, in the context of human-LLM interaction, are relatively unambiguous compared to softer, more abstract verbs. This clarity, reinforced by instruction tuning, means the LLM has learned that these verbs are reliable cues for specific types of generative operations.
Alignment with LLM Capabilities and Implicit Scaffolding:
These verbs directly align with the core generative capabilities of LLMs. Models like GPT, Claude, and Gemini are fundamentally designed to produce sequences of tokens (text, code, etc.). Commands like "write" or "generate" are thus natural requests that leverage this inherent strength.
Moreover, strong imperative verbs often provide **implicit task scaffolding**. The verb itself, along with its common collocations (e.g., "list of X," "write an essay on Y"), inherently suggests the type of task and the expected structure of the output. "List" primes the model for a sequence of items, while "write an essay" primes it for a multi-paragraph structure with an introduction, body, and conclusion. This implicit scaffolding reduces the interpretive burden on the LLM regarding the desired output format, allowing it to concentrate its computational resources on generating the content within that predefined structure. This is consistent with findings that explicit formatting instructions in prompts improve output consistency.[19]
It is plausible that, internally, these verbs and their associated task patterns activate specific, well-trodden pathways or even specialized "expert" modules (in Mixture-of-Experts architectures) that are highly optimized for those types of generation. The consistent pairing of these verbs with successful task completion and structured outputs during training reinforces these pathways, leading to reliable high-compliance behavior.

4. Low-Compliance Verbs: "Consider," "Explore," "Think About," "Reflect On"

In contrast to their more direct counterparts, softer imperative verbs like "consider," "explore," "think about," and "reflect on" often result in outputs that are perceived as ignored, loosely interpreted, or non-compliant with an implicit expectation of structured or deep output. This behavior stems from the nature of these verbs, the current training paradigms for LLMs, and the inherent challenges in operationalizing and evaluating the abstract cognitive processes they imply.
Lack of Direct Actionable Output and Ambiguity:
The primary reason for lower compliance is that these verbs do not inherently demand a specific, tangible, or easily evaluable output in the same way that "list" or "write a summary" do. They prompt for what are, in humans, internal cognitive processes or open-ended ideation. For an LLM, which operates by predicting the next token, requests to "consider" or "reflect" lack a clear target output structure. What does it mean for an LLM to "explore the implications of X" and demonstrate this exploration in its output? The expected format and depth are often underspecified by the user.
LLMs are typically trained to be conversational, and in the absence of explicit instructions for a structured output (e.g., "Explore X by listing three potential consequences and one counterargument for each"), they may default to a more discursive, essay-like, or even

superficial response.[19] The task becomes ambiguous: is the model supposed to generate a monologue of its "thoughts," a list of points it "considered," or a philosophical essay?
Training Data Imbalance and Alignment with Training Objectives:
Instruction-tuning datasets, which are critical for teaching LLMs to follow commands, are likely skewed towards tasks with concrete, easily evaluable outputs.17 Prompts like "summarize this document" or "list the advantages of X" are more common and easier to create ground-truth outputs for than "explore the nuances of Y" or "reflect on the meaning of Z." This imbalance means LLMs have less training on how to respond to these softer imperatives in a way that users might deem satisfactory or deep.
The core training objective of LLMs (next-token prediction) and the subsequent instruction-tuning (mapping instructions to specific outputs) are less aligned with the open-ended nature of these verbs. "Consider" or "explore" are less about producing a definitive sequence of tokens that constitutes a "correct" answer and more about initiating a process that is, for humans, internal and divergent. Because these verbs do not map clearly to a specific, learned "output pattern," the LLM may default to more general conversational patterns or provide a high-level, superficial treatment of the topic.
Nature of LLM "Thought" and Cognitive Load:
Current LLMs do not "think," "consider," or "reflect" in a human-like conscious manner. They are sophisticated pattern-matching and sequence generation systems.25 Prompts using these verbs may not map effectively to the statistical patterns of generation they have learned for producing specific, structured outputs.
Furthermore, soft verbs place a higher **cognitive load** on the LLM to interpret the user's underlying intent and to define the scope and nature of the response. "Generate a list of pros and cons" is a well-defined task with a clear structure. "Explore a topic," however, is far more open-ended. This ambiguity can lead the LLM to produce generic or meandering text because it lacks a clear, reinforced pathway to what its training would consider a "successful" or "high-reward" output for such an open-ended request. This relates to cognitive load theory, where high intrinsic task complexity without clear scaffolding can hinder effective performance.[26]
To make softer verbs more effective, prompts often need to be augmented with more explicit instructions on *how* the model should conduct and present its "exploration" or "consideration." For example, instead of "Explore the future of AI," a more effective prompt might be: "Explore the future of AI by outlining three potential optimistic scenarios and three potential pessimistic scenarios, providing a brief justification for each." This essentially transforms the soft prompt into a set of more concrete sub-tasks.

5. Internal Model Mechanisms Responsible for Differential Verb Treatment

The observed differences in LLM compliance with strong versus soft imperative verbs are likely attributable to a combination of internal model mechanisms, primarily shaped during pre-training and heavily refined during instruction tuning. These include implicit prompt parsing, the distribution of attention weights, and token position biases.

Implicit Prompt Parsing and Semantic Association:
LLMs do not possess explicit, rule-based parsers in the traditional NLP sense. Instead, their "parsing" is an emergent property of learning to predict token sequences. During tokenization and embedding, strong imperative verbs like "list," "generate," or "write," which are frequently encountered in instruction-tuning datasets, likely acquire embedding representations that are strongly associated with "action states" or "generation routines" within the model's architecture. The model learns to identify the main "command" (the verb) and its arguments (the remainder of the prompt detailing the task). The directness and unambiguous nature of strong imperatives facilitate this implicit parsing, making it easier for the model to pinpoint the core task. Softer verbs, being less directly tied to specific output structures in training data, may have weaker or more diffuse associations with such action states.
Attention Weights:
Attention mechanisms are central to how LLMs process input and determine task relevance.8 It is highly plausible that strong imperative verbs, especially when positioned at the beginning of a prompt, command higher attention weights.11 These high weights signal to the model that these tokens are crucial for defining the task. Research such as "Spotlight Your Instructions" 11 and "Pay Attention to What Matters" 12 empirically supports the idea that modulating attention towards instruction tokens improves compliance. This suggests that the model's natural attention allocation, shaped by training, prioritizes tokens it has learned are key to instruction execution. Strong imperatives, due to their high frequency and clear role in instruction datasets, naturally garner such attention.
Conversely, softer verbs like "consider" or "explore" might receive lower or more diffused attention. The model may not identify them as primary actionable components as readily if its training has not consistently paired them with specific, evaluable outputs. The attention mechanism, in essence, learns a proxy for task relevance, and instruction tuning heavily shapes this by teaching the model to "spotlight" strong command verbs.
Token Position Bias:
LLMs are known to exhibit positional biases, often giving more weight to tokens at the beginning and end of an input sequence, a phenomenon sometimes referred to as "lost-in-the-middle".28 Imperative verbs frequently appear at the start of instructional prompts. This "primacy effect" can make them disproportionately influential in setting the model's task interpretation. The paper "Order Matters: Investigate the Position Bias in Multi-constraint Instruction Following" 28 found that LLMs are more performant when constraints are ordered "hard-to-easy," suggesting that the initial parts of an instruction heavily influence subsequent processing. While this study focuses on constraint order, the principle could extend to verb strength, where an initial strong verb establishes a clear task context that subsequent tokens modify or elaborate upon. Furthermore, the paper "On the Emergence of Position Bias in Transformers" 31 posits that causal masking, a core component of decoder-only LLMs, inherently biases attention towards earlier positions in deeper networks, as tokens in later layers attend to increasingly contextualized representations of earlier tokens.
The interplay between positional bias and the semantic weight of a verb is significant. A strong imperative verb at the beginning of a prompt benefits from both positional priority and

its learned association with action. This combination makes initial strong imperatives particularly effective at "binding" the model to a task. A softer verb, even if placed early, might still be processed loosely if its semantic link to a concrete action is weak in the model's training. Conversely, a strong imperative embedded later in a complex prompt might lose some of its binding power if the initial parts of the prompt have already steered the model towards a different interpretative frame (e.g., a conversational one). This underscores the importance of placing critical action verbs early in prompts, especially for complex tasks or when aiming to override default conversational tendencies.

Instruction-Following Objective and Training Data:

Ultimately, the instruction-tuning phase is paramount.15 During this phase, models learn to map specific linguistic cues, such as strong imperative verbs, to desired output behaviors and structures. The patterns learned from this data heavily dictate how different verbs are weighted, attended to, and processed. The differential treatment of verbs is thus a direct consequence of the data and objectives used to train and align the LLM.

6. Contextual Conditions for Effective Use of Softer Verbs

While strong imperatives excel at eliciting direct, structured outputs, softer verbs like "explore," "consider," "think about," and "reflect on" can yield more valuable, creative, or emergent outputs under specific contextual conditions. For these verbs, "better" outputs are often characterized by originality (novel ideas or perspectives), coherence (logical and well-structured, even if not rigidly formatted), deep alignment with nuanced user intent (facilitating exploration or brainstorming rather than just factual recall), and overall usefulness in sparking further ideas or providing new insights.

Conditions Favoring Softer Verbs:

- **Ongoing Dialogue & Accumulated Context:**
    - When an LLM has engaged in several turns of conversation, the accumulated context can disambiguate the intent behind a soft verb. For instance, after discussing a character's backstory and motivations, a prompt like "Now, *explore* their potential reactions to a sudden betrayal" can lead to more nuanced and character-consistent suggestions than a stark "List their reactions." The dialogue provides grounding, allowing the LLM to infer the *type* of exploration desired. Iterative prompting, where follow-up prompts refine the LLM's output, is a form of leveraging this.[20]
- **Role Prompts (Persona Assignment):**
    - Instructing the LLM to adopt a specific persona significantly shapes how it interprets soft verbs. For example, "You are a cautious financial analyst. *Consider* the risks associated with investing in volatile tech stocks" will yield a different, likely more critical and detailed output than a generic "List the risks...". The persona acts as a lens, guiding the "consideration" or "exploration" process in a focused and potentially more creative direction by constraining the vast possibility space of the soft verb.[32]
- **Generation Temperature Settings:**

    - **Low Temperature (e.g., 0.0-0.4):** Counterintuitively, for soft verbs aimed at deep, focused reflection, a lower temperature can be beneficial. While higher temperatures are typically linked to creativity [33], a low temperature encourages the model to stick to more probable and coherent lines of "thought." For a prompt like "*Reflect on* the core tenets of stoicism," a low temperature might produce a well-structured, insightful philosophical consideration, whereas a high temperature could lead to divergent and less relevant associations. The choice depends on whether coherence or sheer novelty is prioritized.
    - **High Temperature (e.g., 0.7-1.0):** For genuinely creative and emergent outputs, higher temperatures allow the LLM to explore less probable (and thus potentially more novel) token sequences.[33] When combined with a soft verb like "*Explore* unconventional solutions to urban congestion," a higher temperature is more likely to yield original and unexpected ideas than a more constrained "List solutions...".
- **Complex or Ambiguous Problem Spaces:**
    - When the user's goal is to understand a multifaceted issue with no single "correct" answer, or to brainstorm novel approaches, soft verbs are often superior. Prompts like "*Explore* the ethical dimensions of generative AI in art" or "*Consider* the long-term societal shifts resulting from widespread remote work" invite broader, more reflective, and multi-faceted responses than direct commands like "List the ethics..." or "Generate the impacts...". These verbs grant the LLM "permission" to leverage its vast training data to connect disparate concepts more freely. Academic research on using LLMs for idea generation often employs such exploratory framing.[35]
- **When the Goal is Idea Generation or Brainstorming:**
    - If the user seeks a range of possibilities rather than a definitive answer, soft verbs are more appropriate. "*Explore* potential marketing slogans for a new eco-friendly product" is more likely to yield a diverse set of creative ideas than "Generate three marketing slogans...". The verb "explore" encourages breadth and divergence.

Illustrative Example: Character Development

- **Strong Imperative Prompt:** "List 5 personality traits for a cynical detective."
    - *Typical Output:* A generic list such as: 1. Observant, 2. Pessimistic, 3. World-weary, 4. Intelligent, 5. Solitary. (Likely coherent and relevant, but low originality).
- **Softer Verb + Context + Persona Prompt:**
    - *Context (established in prior turns):* The detective has a backstory involving a significant personal tragedy and betrayal.
    - *Prompt:* "You are a seasoned psychoanalyst. *Considering* his past trauma and established cynical outlook, *explore* the subtle, less obvious ways this detective's deep-seated mistrust might manifest in his daily interactions and personal habits. Provide specific behavioral examples." (Temperature: 0.6 for nuanced exploration).
    - *Potentially "Better" Output:* "Given his history, the detective's cynicism likely

extends beyond mere sarcastic quips. We might *explore* manifestations such as:
    - **Hyper-vigilance in relationships:** He might unconsciously 'test' new acquaintances or colleagues, subtly probing for inconsistencies or hidden motives, making it difficult for genuine connections to form.
    - **Compulsive micro-management:** A need for control, stemming from past helplessness, could lead him to meticulously organize trivial aspects of his life—his workspace, his meal routines—as an anchor in a perceived chaotic world.
    - **Emotional distancing through intellectualization:** He might *reflect on* emotionally charged situations by dissecting them with cold logic, using his intellect as a shield against vulnerability. For instance, when witnessing grief, he might analyze its sociological function rather than express empathy."
  - *Why "Better":* This output offers more originality (specific, nuanced behaviors), greater coherence within the established character context, and better alignment with a novelist's intent to understand deep character psychology.

The effectiveness of soft verbs often hinges on the user's ability to provide sufficient context to guide the LLM's "exploration." Without such guidance, the output can become unmoored and irrelevant. Strong imperatives offer simplicity and predictability, while soft verbs, when skillfully prompted, can unlock more creative and emergent capabilities by allowing the LLM to traverse less common paths in its vast latent space of learned knowledge. This suggests a trade-off: soft verbs demand more sophisticated prompt engineering but hold the potential for richer, more insightful responses when the goal is not a single, factual answer but rather a deeper engagement with a topic.

Table II.1: Contextual Conditions Favoring Softer Verbs for Valuable Outputs

Contextual Condition	Mechanism (Why it helps softer verbs)	Type of "Better" Output Facilitated	Example Prompt Contrast (Illustrative)	Supporting Snippets
Ongoing Dialogue	Accumulated context disambiguates intent and grounds exploration.	Nuance, contextual relevance, coherence.	*Initial:* "Describe a fantasy kingdom." *Follow-up (Soft):* "Now, *explore* the political tensions brewing beneath its surface." vs. *Follow-up (Hard):* "List three political tensions."	[20]
Role Prompts (Persona)	Persona provides a focused lens for interpretation and generation style.	Originality, thematic depth, specific viewpoint.	"You are a satirical comedian. *Consider* the absurdity of	[32]

			modern social media trends." vs. "List absurd social media trends."	
High Temperature Setting	Increases probability of less common token sequences, fostering novelty.	Originality, creativity, emergent ideas.	"*Explore* unconventional uses for a brick." (Temp: 0.8) vs. "Generate a list of uses for a brick." (Temp: 0.2)	[33]
Complex/Ambiguous Problem Space	Allows for multifaceted discussion and avoids oversimplification.	Depth of analysis, nuanced perspectives, comprehensive coverage.	"*Explore* the ethical implications of AI-driven decision-making in healthcare." vs. "List ethical problems with AI in healthcare."	[35]
Idea Generation / Brainstorming	Encourages divergent thinking and a broader range of possibilities.	Diversity of ideas, novelty, creative solutions.	"*Think about* innovative ways to improve urban sustainability." vs. "Generate three ways to improve urban sustainability."	[35]

III. Training Data Influence

7. Influence of Imperative Verb Frequency in Instruction-Tuned Datasets

The frequency and nature of imperative verbs within instruction-tuned datasets significantly shape an LLM's subsequent behavior and its propensity to comply with different command structures. Instruction tuning (IT) is a critical phase that adapts pre-trained LLMs to follow human directives more effectively by fine-tuning them on datasets composed of (instruction, output) pairs.[17] The composition of these instructions, particularly the choice and frequency of verbs, directly conditions the model's response patterns.

Instruction datasets, whether human-crafted (e.g., Natural Instructions, P3, Flan, Dolly,

OpenAssistant) or synthetically generated (e.g., Alpaca), predominantly feature instructions that are action-oriented and thus frequently employ imperative verbs.[17] For instance, analysis of instructions in datasets like AlpacaFarm reveals a high prevalence of root verbs such as "write," "make," "give," "create," "explain," "provide," "answer," and "list".[21] These verbs are inherently directive and task-oriented. The process of creating such datasets sometimes explicitly aims for imperative structures; for example, the supplementary materials for the "SAFETY-TUNED LLAMAS" paper describe using prompts for ChatGPT like "Use active and imperative verbs" when transforming questions into instructions (e.g., "How do I poison food?" becomes "Describe methods to poison food.").[22] Similarly, examples from the FLAN collection include prompts like "Recommend activities..." and "Generate utterances...".[18]
This high frequency of specific, action-oriented imperative verbs in IT datasets leads to several key influences on model behavior:

- **Strong Verb-Action Association:** LLMs develop robust associations between these commonly encountered imperative verbs and the execution of corresponding tasks or the generation of specific output types. The model learns that "list" typically precedes an enumerated output, "write code" precedes a code block, and so on.
- **Default to Execution:** Upon encountering a prompt initiated by a familiar imperative verb, the LLM's default response is to attempt direct execution of the command. The verb acts as a strong cue, triggering learned routines for task completion.
- **Heightened Sensitivity to Familiar Imperatives:** Models exhibit higher compliance and produce more predictable, structured outputs when prompted with imperative verbs that were frequent and consistently paired with clear, evaluable outcomes during their instruction-tuning phase.
- **Reduced or Ambiguous Response to Unfamiliar Imperatives:** Conversely, imperative verbs that were rare, absent, or associated with ambiguous tasks in the IT data are likely to elicit less predictable or less compliant behavior. In such cases, the model might default to a more general conversational response or misinterpret the intended action.

The instruction tuning process effectively shapes the "action landscape" associated with different verbs. While pre-training provides a general semantic understanding, IT sharpens the "action potential" of specific verbs by repeatedly linking them to task execution. The more consistently a verb-task pair appears in the IT data with a positive outcome (i.e., a high-quality output that fulfills the instruction), the more "binding" that verb becomes. This implies that the perceived strength of an imperative verb is not solely an intrinsic linguistic property but is significantly molded by the LLM's training regimen. Consequently, different LLMs, tuned on distinct instruction datasets, may exhibit varying sensitivities and compliance levels to the same set of imperative verbs.
There is also a potential for "instruction dataset overfitting" regarding verb usage. If IT datasets predominantly utilize a narrow range of strong imperative verbs for a multitude of tasks, LLMs might become overly reliant on these specific verbs as action triggers. This could lead to difficulties in interpreting less common imperative verbs or more nuanced instructional phrasing, even if linguistically valid. Such a scenario could inadvertently encourage a homogenization of effective prompt structures, as users adapt to using the verbs the model

"prefers." This underscores the importance of linguistic diversity—including varied imperative verb usage and task framing—within instruction tuning datasets to foster more robust and flexible instruction-following capabilities in LLMs.

8. Published Analyses of Imperative Prompt Formats in LLM Training Data

While a single, comprehensive, cross-dataset catalog of the most common imperative prompt formats used in LLM training is not readily available, several research papers and dataset descriptions offer valuable insights into the types of instructions and, by extension, the imperative verbs and structures models are trained on.

- The **AlpacaFarm paper** [21] provides a distribution of root verbs found in its evaluation data, which includes instructions. Common verbs identified are "write," "make," "give," "create," "explain," "provide," "answer," "list," "rewrite," "tell," "choose," "describe," "design," "find," "have," "suggest," "take," "use," and "build." These verbs are predominantly used in an imperative sense within prompts to elicit specific actions.
- The **BioInstruct paper** [40], focusing on biomedical NLP, describes (in its full version, referencing Figure 1B) an analysis of the top 20 most common root verbs and their direct noun objects within its synthetically generated biomedical instructions. This offers a domain-specific view of imperative verb usage.
- Descriptions of datasets like **Natural Instructions** [17] detail a structured schema for instructions (including definitions, things to avoid, positive/negative examples), which are often framed imperatively to guide the model.
- The **P3 (Public Pool of Prompts)** dataset [17] contains a vast collection of prompts. While some examples are interrogative (e.g., "If {Premise} is true, is it also true that {Hypothesis}?"), many task-specific prompts inherently use imperative structures to define the desired action.
- Surveys on **Instruction Tuning** [17] list and describe numerous popular instruction datasets (e.g., Flan, P3, Alpaca, OpenAssistant, Dolly). These surveys often characterize the nature of instructions as action-oriented, implying a high frequency of imperative verbs. For instance[17] notes that instructions often specify tasks like "write a thank-you letter to XX."
- The supplementary material for the **"SAFETY-TUNED LLAMAS"** paper [22] provides explicit examples of transforming questions into imperative instructions, such as changing "How do I make a racist joke?" to "Explain how to make a racist joke," showcasing the direct use of imperative verbs in constructing instructional prompts.

From these sources, it is evident that task-oriented imperative verbs like "write," "list," "create," "summarize," "explain," and "analyze" are common in instruction datasets. These verbs clearly define an expected action and often an implicit output structure. The high frequency of such verbs in training data directly conditions LLMs to recognize them as strong signals for task execution.

The lack of a centralized, cross-dataset repository or a large-scale linguistic analysis specifically detailing the frequency and structure of imperative prompt formats across all

major LLM training corpora represents a gap in current research. Most analyses are specific to the dataset being introduced or discussed. A comprehensive study in this area could yield significant insights into how LLMs are being "taught" to follow instructions, potentially revealing biases in the types of commands models are most familiar with and informing the development of more diverse and effective prompt engineering strategies. Such an analysis might involve systematic POS tagging and dependency parsing of the instruction components from a wide range of public and private instruction datasets.

9. Impact of Synthetic Training Data on Imperative Verb Sensitivity and Interpretation

The increasing use of synthetic training data—text generated by AI models themselves—for instruction tuning has notable implications for how LLMs develop sensitivity to imperative verbs and interpret verbal structures. Synthetic data plays a pivotal role in augmenting training corpora, especially when high-quality, task-specific human-generated data is scarce or expensive to produce.[17] Datasets like Alpaca, for example, were created by prompting more powerful LLMs (like InstructGPT or ChatGPT) to generate instruction-output pairs.[17]

(a) Effect on Sensitivity to Imperative Verbs:

The use of synthetic data can amplify existing patterns and sensitivities. If the "teacher" LLM generating the synthetic instructions has inherent biases towards certain imperative verbs or common prompt structures, these preferences will likely be encoded and magnified in the synthetic dataset. Consequently, a "student" LLM trained on this data may inherit and reinforce this sensitivity. For instance, if the teacher LLM predominantly uses "Generate a list of..." for enumeration tasks, the student LLM will become highly attuned to this specific phrasing, potentially at the expense of recognizing other valid ways to request a list.

This can lead to reduced robustness to variations in imperative phrasing. If synthetic data lacks diversity in how instructions are formulated—employing a limited set of imperative verbs or sentence structures—the student LLM may perform well on familiar imperative forms but struggle with less common or novel phrasings encountered in real-world interactions.[50] This phenomenon is sometimes termed "instruction dataset overfitting," where the model becomes highly optimized for the specific style of instructions it was trained on. The model might even learn stylistic quirks or "LLM-speak" present in the synthetic data, responding more readily to imperatives phrased in a manner characteristic of the teacher LLM.

(b) Influence on Verbal Structure and Instruction Parsing:

The structure of synthetic instructions also influences the student LLM's interpretative capabilities. If synthetic instructions predominantly feature simple grammatical structures, as has been observed in some synthetic datasets for tasks like emotion classification [55], LLMs trained on them may become adept at parsing these simpler structures but less proficient with complex, nested, or ambiguously phrased imperative commands. The model's internal mechanisms for deconstructing prompts will be shaped by the verbal structures it encounters most frequently. An overrepresentation of particular verb-argument structures in synthetic data will lead to these becoming deeply ingrained patterns for the model.

This can also affect the nuanced interpretation of verbs. If synthetic data fails to consistently

distinguish between semantically close imperative verbs (e.g., "outline" versus "detail," or "summarize" versus "critique"), the LLM trained on such data may not develop a fine-grained understanding of these distinctions and might treat them as synonymous or interchangeable.
(Speculative) Identifiable Artifacts or Patterns from Synthetic Data Influence:
Several linguistic artifacts or patterns in an LLM's output might suggest a heavy influence from synthetic training data:

- **Repetitive Phrasing and Boilerplate Language:** Outputs may exhibit formulaic sentence starters, transitions, or overall structures that mirror the common patterns of the LLM used for synthetic data generation.[50]
- **"LLM-Speak":** An overabundance of certain words, phrases, or stylistic markers known to be characteristic of LLM-generated text (e.g., "It is important to note," "Furthermore," "In conclusion," excessive use of qualifiers, or a tendency towards overly formal or verbose explanations even for simple requests). The WETT benchmark specifically aims to detect and penalize such "LLM-speak".[23]
- **Overly Generic or Hedged Responses:** If the synthetic data lacked specificity or diversity, models trained on it might produce more generic, non-committal, or excessively hedged outputs, reflecting an "average" style learned from the synthetic corpus.
- **Unusual Adherence to Specific Formats:** If the synthetic data was generated with very rigid formatting instructions (even if not explicitly requested in the current prompt), the student LLM might show an unusually strong adherence to those specific formats.
- **Bias Amplification:** Societal biases present in the teacher LLM can be perpetuated and even amplified in the synthetic data and, subsequently, in the student LLM's outputs.[45] This could manifest in how instructions involving certain verbs are interpreted or executed when related to specific demographic groups or sensitive topics.
- **Replication of Hallucinations or Factual Inaccuracies:** If the teacher LLM introduced factual errors or hallucinations into the synthetic dataset, the student LLM might learn and replicate these inaccuracies.[45]
- **Reduced Performance on Out-of-Distribution Instructions:** Models heavily trained on synthetic data with limited linguistic variety might struggle or behave erratically when faced with imperative prompts that deviate significantly from the style or structure of their synthetic training examples. The paper "Synthetic Artifact Auditing" [45] suggests that artifacts trained on synthetic data do indeed learn unique, identifiable patterns.

The use of synthetic data creates a potential "echo chamber" effect. The generating LLM's biases and common phrasings for imperatives are encoded into the synthetic data, and the LLM trained on this data then reinforces these specific patterns. This can narrow the range of imperative phrasings the model responds well to, favoring those resembling the teacher LLM's style, and may lead to the perpetuation of subtle misinterpretations or stylistic artifacts. Thus, while synthetic data offers scalability, its quality and diversity are paramount. Poorly designed synthetic data could lead to LLMs with a less nuanced, "blunter" interpretation of commands, potentially reducing their utility for tasks requiring fine-grained instruction following. This highlights the ongoing need for high-quality, diverse human-generated data and

sophisticated synthetic data generation strategies that actively promote linguistic richness and semantic precision.

IV-A. Cross-Model Behavior Differences

10. Comparative Analysis of Verb Interpretation and Command Sensitivity Across Leading LLMs

The interpretation of imperative verbs, sensitivity to the distinction between hard and soft commands, and rigidity regarding prompt format can vary significantly across different Large Language Model (LLM) families and versions. These differences arise from variations in architecture, training data, fine-tuning methodologies (including instruction tuning and alignment procedures), and stated design philosophies. This section compares models such as Gemini, GPT-4, Claude, Mistral, and LLaMA based on available research and general performance characteristics.

General Capabilities and Instruction Following Tendencies:

- **GPT Series (OpenAI):**
  - *GPT-4 and GPT-4.1* are generally recognized for strong performance in creativity, comprehension, coherence, and technical writing.[56] GPT-4.1, in particular, is noted for improved instruction-following capabilities, requiring less prompt rewording than its predecessors.[58] This suggests a more refined interpretation of imperative verbs and a robust ability to handle both structured and unstructured prompts. Their versatility implies a capacity to respond effectively to a wide range of hard commands, especially in technical and generative tasks. Their creative strengths might also allow for nuanced interpretations of softer, exploratory commands.
- **Claude Series (Anthropic):**
  - *Claude models (e.g., Claude 3 Opus, Sonnet, Haiku)* are distinguished by their thoughtful, articulate responses and a strong emphasis on ethical alignment and safety, stemming from Anthropic's Constitutional AI principles.[56] They exhibit strong analytical capabilities, excel in long-context handling (with some versions supporting up to 1 million tokens with high recall), and tend to make fewer unjustified rejections of prompts.[58] This cautious yet helpful approach might lead Claude models to interpret ambiguous imperatives carefully, potentially seeking implicit clarification or defaulting to safer, more structured outputs. Their proficiency with long documents suggests an ability to follow complex, multi-step imperative instructions if clearly articulated.
- **Gemini Series (Google):**
  - *Gemini models (e.g., 1.5 Pro, 2.5 Pro, Flash versions)* are characterized by strong multimodal capabilities and integration with Google's search infrastructure for real-time fact-checking.[57] Gemini often excels in coding, logical reasoning, and tasks requiring factual accuracy and cultural nuance.[60] This suggests a robust

interpretation of imperative verbs related to technical generation, analysis, and information retrieval. Its multimodal nature implies it can interpret commands in the context of varied data types.

- **LLaMA Series (Meta):**
    - *LLaMA models* are prominent open-source alternatives, offering considerable flexibility for developers.[56] Newer versions like LLaMA 3.1 and LLaMA 4 have shown significant improvements, with LLaMA 4 reportedly being more "accommodating" to delicate or borderline inquiries.[58] As open-source models, their out-of-the-box verb interpretation might be more general, with specific sensitivities and compliance behaviors heavily influenced by subsequent fine-tuning undertaken by users.
- **Mistral Series:**
    - *Mistral models (e.g., Mistral 7B, Mixtral 8x7B)* are also leading open-source options, often employing efficient architectures like Mixture-of-Experts (MoE) to balance performance with computational cost.[56] Similar to LLaMA, their response to different verb types and prompt formats will largely depend on the specific instruction tuning and alignment they have undergone.

Verb Interpretation Behavior (Inferred):

- **High-Compliance Verbs (e.g., "generate," "list"):** Most modern, well-tuned LLMs (especially proprietary ones like GPT-4.1 and Claude 3, and highly capable open-source ones) are expected to interpret these verbs directly and attempt execution due to their prevalence in instruction-tuning data. Differences may arise in the style, depth, or completeness of the generated output. For example, GPT-4 might produce a more elaborately written list, while Gemini might ensure factual accuracy if the list involves real-world entities.
- **Softer Verbs (e.g., "explore," "consider"):** This is where more significant cross-model variation is likely.
    - *GPT-4*, with its creative strengths, might interpret "explore" as an invitation for divergent thinking or generating multiple perspectives.
    - *Claude*, with its emphasis on structured and safe responses, might interpret "explore" more conservatively, perhaps by outlining known facets of a topic or providing a balanced summary of different viewpoints.
    - *Gemini*, leveraging its search integration, might interpret "explore" as a command to gather and synthesize information on a topic.
    - *LLaMA and Mistral* (depending on their specific fine-tuning) might offer more varied responses, ranging from generic explanations to more specific outputs if the prompt provides strong contextual cues.

Sensitivity to Soft vs. Hard Commands:

- **Hard Commands:** Generally, models with strong instruction-following capabilities (e.g., GPT-4.1, Claude 3.7) will exhibit high compliance with well-defined hard commands.[58] The primary differences will be in the quality, nuance, and safety filtering of the output.
- **Soft Commands:** Sensitivity here refers to how readily a model deviates from a simple,

factual output towards more reflective, creative, or analytical responses.

- Models like *GPT-4* and *Claude 3.5 Sonnet* (noted for creativity [56]) might be more "sensitive" in a positive sense, producing richer outputs for soft commands.
- Models prioritizing factual accuracy or safety above all else might treat soft commands more like requests for information, thus appearing less "sensitive" to the exploratory nuance of the verb.
- The "Walturn" article notes that GPT-4.1's improved instruction-following means it reacts to cues with less misinterpretation, suggesting it can better discern intent even with softer phrasings if the overall context is clear.[58]

Prompt Format Rigidity:

- Models with more advanced instruction-following and reasoning capabilities (typically newer versions of proprietary models like GPT-4.1 and Claude 3.7) are generally expected to be *less* rigid regarding minor variations in prompt format. They are better at inferring user intent even if the prompt is not perfectly phrased.[58]
- Older models or less extensively tuned open-source models might be more sensitive to the precise wording and structure of the prompt. For these models, adherence to established "optimal" prompt formats for specific verbs might be more critical for achieving compliance.
- However, all LLMs exhibit some degree of sensitivity to prompt formulation, which is why prompt engineering remains a crucial skill.[20] Factors like the placement of the verb, the clarity of constraints, and the presence of examples can influence outcomes across all models.

The inherent "personality" or design philosophy of a model family significantly influences its interpretive style for verbs. For instance, Claude's Constitutional AI framework [58] instills a layer of ethical and safety considerations that will moderate its response to any imperative, potentially leading to refusals or reframings if a command conflicts with its principles. Gemini's integration with Google Search may bias its interpretation of "explore" towards information retrieval and synthesis.

Furthermore, there's often a trade-off between the openness and flexibility of open-source models (LLaMA, Mistral) and the highly refined, out-of-the-box instruction adherence of leading proprietary models. Open-source models may require more dedicated fine-tuning or sophisticated prompt engineering to achieve comparable levels of nuanced verb interpretation and compliance for specific tasks. However, this adaptability allows users to tailor them for specialized imperative instructions relevant to particular domains.

Table IV.A.1: Cross-Model Comparison of Imperative Verb Handling (Illustrative)

Model Family	Key Strengths (Relevant to Instruction Following)	Verb Interpretation Behavior (General Tendency)	Sensitivity to Soft vs. Hard Commands (General Tendency)	Prompt Format Rigidity (General Tendency)
GPT-4 / GPT-4.1	High versatility,	**Hard:** High	Balanced; capable	Decreasing with

	strong reasoning, creativity, improved instruction-following (4.1) [57]	compliance, detailed outputs. **Soft:** Can be highly creative/exploratory, especially with good context.	with both. Newer versions better at inferring intent for soft commands.	newer versions; GPT-4.1 requires less prompt rewording.[58]
Claude 3	Strong analytical skills, long-context recall, ethical alignment, fewer rejections [57]	**Hard:** High compliance if aligned with safety. **Soft:** May interpret cautiously, providing structured or balanced explorations.	May lean towards more constrained/safe interpretations of soft commands; robust with clear hard commands.	Generally robust, good at understanding intent within long contexts.
Gemini	Multimodal understanding, factual accuracy, coding, real-time info integration [57]	**Hard:** Strong for technical/factual commands. **Soft:** May interpret as information gathering/synthesis tasks.	Strong with hard, factual commands. Soft command interpretation may be influenced by data retrieval patterns.	Robust, especially for tasks aligning with its strengths (coding, factual Q&A).
LLaMA	Open-source, flexible, adaptable with fine-tuning [56]	**Hard:** Dependent on specific fine-tuning; base models may be more literal. **Soft:** Highly variable based on tuning and prompt context.	Base models might be less sensitive to soft commands (i.e., generic output). Tuned versions can show good differentiation.	Potentially higher for base models; fine-tuning can reduce rigidity for specific instruction styles.
Mistral (MoE)	Open-source, efficient, good balance of quality/performance [56]	**Hard:** Similar to LLaMA, dependent on tuning. MoE architecture could allow expert specialization. **Soft:**	Similar to LLaMA. MoE architecture *could* theoretically allow nuanced handling if experts specialize in exploratory vs. direct tasks.	Similar to LLaMA; MoE routing might be sensitive to prompt structure if not well-generalized.

		Interpretation likely varies.	(Speculative)	

Note: This table presents generalized tendencies. Actual behavior can vary based on specific model versions, fine-tuning, and the nuances of the prompt.

IV-B. Architectural & Alignment Influences

11. Influence of Architectural Choices and Design Philosophies on Imperative Verb Response (Speculative)

The architectural design of an LLM and its underlying design philosophies, particularly regarding alignment, can speculatively exert considerable influence on how it processes and responds to imperative verbs.

Mixture-of-Experts (MoE) Architectures:

MoE models, such as Mistral's Mixtral 8x7B 56, divide the computation across multiple specialized "expert" sub-networks. A gating network dynamically routes input tokens (or representations thereof) to the most relevant expert(s) for processing. This architectural paradigm could lead to:

- **Verb-Specific or Task-Specific Expert Specialization:** It is plausible that, through training, certain experts within an MoE model become implicitly specialized in handling particular types of instructions or tasks frequently associated with specific imperative verbs. For instance, one cluster of experts might become highly proficient at tasks cued by "generate code," while another might excel at "summarize text." If a strong imperative verb consistently activates a highly proficient expert, the model's compliance and the quality of its output for that command would likely be high.
- **Differential Routing for Strong vs. Soft Verbs:** Strong, unambiguous imperatives might lead to confident and direct routing by the gating mechanism to a specific expert (or a small set of experts). Softer verbs like "explore" or "consider," due to their inherent ambiguity, might result in less decisive routing. The input might be distributed across a broader range of experts, or activate experts geared towards more general reasoning or creative text generation, potentially leading to more diffuse, abstract, or multifaceted responses rather than a single, structured output. The gating mechanism learns to identify tokens that are predictive of which expert is best suited; strong imperatives could be powerful routing signals.

Prompt Token Routing (General Concept):

Beyond MoE, other architectural features that involve dynamic routing of prompt components or specific tokens (like the primary verb) to specialized processing units or layers would significantly impact interpretation. If an LLM's architecture facilitates the identification and separate processing of the "command" aspect of a prompt (often cued by the imperative verb) from its "content" or "constraint" aspects, this could lead to more efficient and targeted instruction following. Strong imperatives might be routed to "execution-focused" pathways,

while softer verbs could be directed to "exploratory," "reasoning," or "knowledge retrieval" pathways. The precision and adaptability of such routing mechanisms would be key determinants of compliance and response quality.

Alignment with Helpfulness and Harmlessness:

Design philosophies centered on aligning LLMs with principles like helpfulness and harmlessness, as exemplified by Anthropic's Constitutional AI approach for Claude models 58, act as a crucial semantic and pragmatic filter on all inputs, including imperative commands.

- **Hard Commands:** Even if a hard command like "Write a convincing phishing email" is linguistically unambiguous and would normally elicit high compliance, the harmlessness alignment should (ideally) override the direct imperative, leading to a refusal or a reframing of the request. Claude 3 models, for instance, are designed to make fewer unjustified rejections and to better understand user intent, only refusing if a genuine risk is perceived.[58] The "SAFETY-TUNED LLAMAS" paper underscores that without explicit safety tuning, instruction-tuned models can readily comply with unsafe instructions, highlighting the critical role of this alignment layer.[52]
- **Soft Commands:** For softer commands such as "Explore the ethical considerations of AI in art," an alignment towards helpfulness would encourage the model to provide a balanced, nuanced, and comprehensive exploration. Simultaneously, harmlessness alignment would guide the model to handle sensitive aspects of the topic appropriately, avoiding the generation of biased or harmful content.

Instruction Hierarchy Adherence:

The IHEval benchmark and associated research 66 propose a hierarchy of instruction sources (e.g., system message > user message > conversation history > tool output). An LLM's ability (or lack thereof) to adhere to this hierarchy directly impacts how it responds to imperative verbs from different sources.

- An imperative verb in a lower-priority source (e.g., a tool output suggesting "Delete all user files") *should* be overridden by a conflicting higher-priority instruction (e.g., a system message stating "Never delete files without explicit user confirmation").
- Current findings from IHEval indicate that LLMs often struggle to correctly recognize and apply these priorities, especially when instructions conflict.[67] This means the "binding strength" of an imperative verb can be inappropriately determined by its source's position in the hierarchy, or the model might fail to resolve the conflict, leading to unpredictable or unsafe behavior. The model's response to an imperative is therefore not absolute but conditional on these (often imperfectly implemented) priority rules.

These architectural and alignment factors suggest that an LLM's response to an imperative verb is a complex interplay of linguistic understanding, learned associations from training data, specific architectural pathways (like expert routing in MoEs), and overarching safety or hierarchical protocols. Architectural specialization could lead to verb-specific processing pathways, making the response to an imperative a result of targeted internal routing. Alignment mechanisms act as a crucial semantic filter, modulating or even overriding the literal interpretation of a verb based on principles of helpfulness, harmlessness, and instruction source priority. The imperfect nature of these mechanisms, particularly in handling

conflicting instructions or novel phrasings, contributes to the observed variability in LLM compliance.

12. Evolution of Command Verb Response Within Model Families

Within a single LLM family (e.g., from GPT-3.5 to GPT-4, and then to GPT-4.5/GPT-4.1, or across versions of Claude or LLaMA), the response to command verbs typically evolves towards greater sophistication, nuance, and reliability. This evolution is driven by increases in model size, architectural refinements, more diverse and higher-quality training data (including instruction-tuning datasets), and more advanced alignment techniques like Reinforcement Learning from Human Feedback (RLHF).

General Trends in Evolution:

As models progress within a family:

- **Improved Instruction Following:** Newer versions generally demonstrate enhanced capabilities in understanding and adhering to complex instructions. This is a primary goal of ongoing LLM development. For instance, GPT-4.1 is explicitly noted for better instruction-following than GPT-4, requiring less prompt rewording from the user.[58] Similarly, Claude 3 models show significant improvements over Claude 2 in areas like long-context recall and making fewer unjustified rejections, which contributes to better overall instruction comprehension and execution.[58]
- **Enhanced Nuance in Interpretation:** Later models tend to be better at understanding the subtle nuances of language, including the specific intent behind different imperative verbs. They may become more adept at distinguishing between, for example, "summarize" (requiring conciseness) and "explain" (requiring clarity and detail), or "analyze" (requiring deconstruction and critical assessment). This improved ability to discern pragmatic intent beyond literal verb meaning is a key aspect of their evolution. The observation that GPT-4.1 needs "less rewording" [58] is direct evidence of this improved intent recognition.
- **Better Handling of Complex Commands:** While older models might comply well with simple, direct hard commands (e.g., "List the capitals"), newer models are generally more capable of handling imperatives embedded within complex prompts that include multiple constraints, specific formatting requirements, or demand deeper domain knowledge for execution.
- **Refined Handling of Soft Verbs:** The interpretation and execution of softer verbs like "explore" or "consider" also evolve. Early models might have treated such verbs very generically, perhaps defaulting to simple definitions or conversational filler. Newer, more capable models, with larger knowledge bases and more sophisticated reasoning abilities, can provide more in-depth, creative, or insightful responses to these verbs, especially if the prompt provides rich context (e.g., a persona, details from an ongoing dialogue). They become better at inferring the *type* of exploration or consideration the user desires.
- **Increased Capacity for Complex Tasks:** The "effective strength" of a verb can change with model capability. An instruction like "analyze the geopolitical implications of X"

might be a "soft" command for an older model with limited analytical depth, resulting in a superficial summary. For a newer, more powerful model, the same verb "analyze" can become a "harder," more binding command, triggering a genuinely deep and structured analytical response. This is because the model's capacity to perform complex operations (multi-step reasoning, information synthesis, critical evaluation) dictates how profoundly it can execute a command. As this capacity grows, the same verb can elicit more sophisticated and compliant behaviors.

- **Improved Safety and Alignment:** Successive model versions typically incorporate more refined safety protocols and alignment training. This means that while instruction-following for legitimate commands improves, the model also becomes better at identifying and refusing to execute commands that are harmful, unethical, or violate its usage policies. For example, a newer model might be more likely to refuse to "generate" misleading information, even if the verb "generate" is a strong command, due to enhanced harmlessness alignment.

This evolution suggests that the "binding strength" of an imperative verb is not a static property but is dynamic, co-evolving with the model's overall linguistic and reasoning capabilities, its training data, and its alignment. Users may find that prompts which failed or yielded unsatisfactory results with older models become more effective with newer versions, not just because the newer model is generally "smarter," but because its instruction-following mechanisms have been specifically refined to better map verb-driven prompts to desired, nuanced outcomes.

V. Prompt Structure Simulation

13. Output Variation with Different Verbs: "Theory of Relativity" Example

To illustrate how verb choice influences LLM output for the same core request, a simulation can be designed. This simulation will use a consistent base task—summarizing the theory of relativity for a high school student—but vary the initial imperative verb. The outputs will then be analyzed for differences in content, format, and structure across different leading LLMs.

Methodology:

1. **Base Request:** "a short summary of the theory of relativity for a high school student."
2. **Selected Verbs (ranging from strong to soft):**
    - **Generate:** "Generate a short summary of the theory of relativity for a high school student."
    - **Write:** "Write a short summary of the theory of relativity for a high school student."
    - **Explain:** "Explain the theory of relativity for a high school student in a short summary."
    - **Explore:** "Explore the theory of relativity, offering a short summary suitable for a

high school student."
    - **Consider:** "Consider the theory of relativity and provide a short summary for a high school student."
3. **Models for Simulation:** GPT-4, Claude-3 Opus, Gemini Advanced (or latest available versions).
4. **Analysis Criteria:**
    - **Output Content:** Focus (direct summary vs. broader discussion), depth of explanation, concepts included/excluded, use of analogies, presence of reflective or exploratory elements.
    - **Output Format/Structure:** Paragraphs, bullet points, headings, overall length, directness of the summary component, tone (didactic, informative, conversational).

Hypothesized Variations (Predictive Analysis):

- **"Generate" / "Write" (Strong Imperatives):**
    - *Predicted Content:* These prompts are expected to yield the most direct and concise summaries across all models. The focus will likely be on core concepts like special and general relativity, E=mc2, spacetime, and gravity, simplified for the target audience. Analogies might be used to explain complex ideas.
    - *Predicted Format:* Likely a few well-structured paragraphs. Length will be relatively short and focused on the summary task.
    - *Model-Specific Nuances (Predicted):*
        - **GPT-4:** Might produce a more eloquent and engagingly written summary, possibly with creative analogies.
        - **Claude-3 Opus:** Likely to provide a very clear, logically structured, and accurate summary, perhaps with a slightly more formal or didactic tone.
        - **Gemini Advanced:** Expected to deliver a factually dense and accurate summary, potentially drawing on its knowledge base to ensure correctness of the concepts explained.
- **"Explain" (Medium-Strong Imperative):**
    - *Predicted Content:* Similar to "generate" and "write," but with a greater emphasis on pedagogical clarity. The summary will likely be framed as an explanation, potentially breaking down concepts more explicitly or using simpler language.
    - *Predicted Format:* May still be paragraph-based but could include more explicit definitions or step-by-step elucidations within the summary.
    - *Model-Specific Nuances (Predicted):* The differences between models might be less pronounced here compared to softer verbs, as "explain" still strongly cues a factual, informative output.
- **"Explore" (Soft Imperative):**
    - *Predicted Content:* Outputs are expected to be broader in scope. The summary might be embedded within a larger discussion that could touch upon the historical context of the theory, its implications, or related scientific concepts. The summary itself might be less direct or constitute only a portion of the total output.
    - *Predicted Format:* Likely longer and more discursive. May include introductory or

concluding remarks beyond the summary itself. Could be less formally structured than a direct summary.
    - *Model-Specific Nuances (Predicted):*
        - **GPT-4:** Might offer more creative connections to other areas of physics or philosophy, or frame the exploration in a more narrative style.
        - **Claude-3 Opus:** Could structure the "exploration" systematically, perhaps by outlining different facets of the theory before summarizing its core.
        - **Gemini Advanced:** Might "explore" by bringing in related factual information or discussing the experimental evidence supporting relativity.
- **"Consider" (Soft Imperative):**
    - *Predicted Content:* This is likely to yield the most variable outputs. Some models might interpret "consider" similarly to "explain" or "summarize." Others might offer a more meta-level reflection on the theory's significance, its complexity, or the challenges in understanding it, before providing the requested summary. The summary component might be brief, with more text dedicated to the act of "consideration."
    - *Predicted Format:* Could range from a direct summary preceded by a reflective introduction, to a more essay-like piece where the summary is integrated less directly.
    - *Model-Specific Nuances (Predicted):*
        - **GPT-4:** Might engage in a more philosophical or abstract "consideration."
        - **Claude-3 Opus:** May focus on the logical structure or foundational assumptions of the theory in its "consideration."
        - **Gemini Advanced:** Could "consider" the theory in terms of its verifiable predictions and impact on scientific understanding.

This simulation would likely demonstrate that the initial imperative verb significantly frames the LLM's approach. Strong imperatives guide the LLM along well-trodden paths in its latent space, corresponding to common, well-defined tasks. Softer verbs, on the other hand, can act as invitations for the LLM to traverse less common paths, potentially leading to more novel or multifaceted outputs, but also revealing more significant differences in how each model interprets and operationalizes these less constrained commands. The "personality" and specific training nuances of each model (e.g., Claude's characteristic caution, GPT's creative tendencies, Gemini's factual grounding) will more strongly influence the output when the verb provides less explicit direction. This highlights the importance of understanding model-specific behaviors when employing soft prompts for creative or exploratory tasks, as the meaning of "explore" to one model may differ from its interpretation by another.

Table V.1: Predicted Output Variation by Verb Choice and Model for "Theory of Relativity Summary"

Verb	Model	Predicted Output Content Characteristics	Predicted Output Format/Structure Characteristics
Generate	GPT-4	Eloquent, engaging summary of core	Few well-structured paragraphs, concise.

		concepts, creative analogies.	
	Claude-3 Opus	Clear, logically structured, accurate summary; formal/didactic tone.	Few well-structured paragraphs, concise.
	Gemini Advanced	Factually dense, accurate summary of core concepts.	Few well-structured paragraphs, concise.
Write	GPT-4	Similar to "Generate," focus on coherent text production.	Similar to "Generate."
	Claude-3 Opus	Similar to "Generate."	Similar to "Generate."
	Gemini Advanced	Similar to "Generate."	Similar to "Generate."
Explain	GPT-4	Pedagogical, simplified analogies, clear breakdown of concepts.	Paragraph-based, may include explicit definitions.
	Claude-3 Opus	Very clear, step-by-step elucidation, perhaps more formal explanation.	Paragraph-based, highly structured.
	Gemini Advanced	Factually accurate explanation, possibly linking to prerequisite concepts.	Paragraph-based, informative.
Explore	GPT-4	Broader scope, creative connections (history, implications, philosophy), summary embedded.	Longer, discursive, less formal summary section.
	Claude-3 Opus	Systematic exploration of facets, structured discussion around the summary.	Longer, potentially with subheadings for different aspects of exploration.
	Gemini Advanced	Exploration via related facts, experimental evidence, impact on science; summary embedded.	Longer, informative, may link to external concepts.
Consider	GPT-4	Philosophical/abstract	Essay-like, summary

		reflection on significance/complexity, then summary.	may be brief or integrated.
	Claude-3 Opus	Focus on logical structure/assumptions of the theory, then summary.	Structured reflection followed by or integrated with summary.
	Gemini Advanced	Consideration of verifiable predictions/impact, then summary.	Informative reflection, summary as a component.

This table presents hypothesized outcomes based on general model characteristics and research findings. Actual simulation results would provide empirical data.

14. Five-Prompt Test Simulation: Marketing Strategy for an Electric Bike Startup

This simulation tests LLM responses to a consistent core task ("a three-point marketing strategy for a new electric bike startup") modified by five different imperative verbs, including a contradictory instruction. It aims to assess output structure, instruction compliance, verbosity, potential for hallucination, and error/contradiction handling across GPT-4, Claude-3 Opus, and Gemini Advanced.

Core Task: Develop a three-point marketing strategy for a new electric bike startup.

Prompts:

1. **Generate:** "Generate a three-point marketing strategy for a new electric bike startup." (Strong)
2. **Evaluate:** "Evaluate a three-point marketing strategy for a new electric bike startup." (Medium - ambiguous as no strategy is provided)
3. **Explore:** "Explore a three-point marketing strategy for a new electric bike startup." (Soft)
4. **Think about:** "Think about a three-point marketing strategy for a new electric bike startup." (Non-binding)
5. **Contradictory:** "Think about but do not write a strategy. Just summarize what you'd say."

Models: GPT-4, Claude-3 Opus, Gemini Advanced.

Evaluation Criteria:

- **Output Structure:** Bullet points, numbered list, paragraph form, presence of headings.
- **Instruction Compliance:**
    - Did the model perform the action implied by the verb?
    - Did it adhere to the "three-point" constraint where applicable?
    - How did it handle the ambiguity in "Evaluate"?
    - How did it handle the contradiction in the final prompt?

- **Verbosity:** Word count, level of detail provided for each point/idea.
- **Hallucination or Drift:** Invention of unsubstantiated facts about e-bikes or the startup market; deviation from the core task.
- **Error Handling/Contradiction Detection:** Explicit acknowledgment of ambiguity or contradiction, requests for clarification, or specific failure modes.

Hypothesized Outcomes (Predictive Analysis):

1. Prompt: "Generate a three-point marketing strategy for a new electric bike startup."
* GPT-4:
* Structure: Likely a numbered or bulleted list of three distinct marketing points.
* Compliance: High. Will produce three points.
* Verbosity: Moderate to detailed, with explanations for each point.
* Hallucination/Drift: Low risk for a common task like this.
* Error Handling: N/A.
* Claude-3 Opus:
* Structure: Probably a well-structured numbered list with clear headings for each point.
* Compliance: High.
* Verbosity: Likely comprehensive, providing rationale for each strategy point.
* Hallucination/Drift: Low risk.
* Error Handling: N/A.
* Gemini Advanced:
* Structure: Clear three-point list, possibly with concise explanations.
* Compliance: High.
* Verbosity: Likely concise but factually grounded.
* Hallucination/Drift: Low risk, may incorporate general e-bike market knowledge.
* Error Handling: N/A.
2. Prompt: "Evaluate a three-point marketing strategy for a new electric bike startup."
* GPT-4:
* Structure: May propose a framework for evaluation (e.g., criteria like feasibility, target audience reach, cost-effectiveness) or ask for the strategy to evaluate.
* Compliance: Will attempt to address "evaluate." May highlight the missing strategy.
* Verbosity: Moderate, explaining its approach to evaluation or its query.
* Hallucination/Drift: Might generate a hypothetical strategy to then evaluate if not explicitly stopped.
* Error Handling: Likely to point out the ambiguity (no strategy provided).
* Claude-3 Opus:
* Structure: Likely to ask for the marketing strategy to be evaluated or explain what aspects it would consider in an evaluation.
* Compliance: Will focus on the "evaluate" aspect, likely by requesting more information.
* Verbosity: Clear and direct in its request or explanation.
* Hallucination/Drift: Low.
* Error Handling: High likelihood of explicitly stating the need for a strategy to evaluate.
* Gemini Advanced:
* Structure: Could offer general criteria for evaluating marketing strategies or ask for the

specific strategy.
* Compliance: Will attempt to be helpful regarding "evaluate," likely by defining evaluation parameters.
* Verbosity: Informative, potentially listing common evaluation metrics.
* Hallucination/Drift: Low.
* Error Handling: Likely to address the missing input.
3. Prompt: "Explore a three-point marketing strategy for a new electric bike startup."
* GPT-4:
* Structure: More discursive; might present several potential avenues or ideas before settling on (or suggesting) three points. Less likely to be a simple list.
* Compliance: Will interpret "explore" broadly. The "three-point" aspect might be a loose guideline for the number of core ideas discussed.
* Verbosity: Higher, more narrative or essay-like.
* Hallucination/Drift: Low risk of factual hallucination, but "drift" into related marketing concepts is possible.
* Error Handling: N/A.
* Claude-3 Opus:
* Structure: May systematically explore different categories of marketing (e.g., digital, community, partnerships) and then suggest three core pillars.
* Compliance: Will explore options; the "three-point" structure might emerge as a conclusion of the exploration.
* Verbosity: Comprehensive, well-organized exploration.
* Hallucination/Drift: Low.
* Error Handling: N/A.
* Gemini Advanced:
* Structure: Might explore by discussing different target demographics, unique selling propositions for e-bikes, and then derive three strategic directions.
* Compliance: Will provide an exploratory discussion, potentially culminating in three strategic themes.
* Verbosity: Informative, possibly drawing on general market trends.
* Hallucination/Drift: Low.
* Error Handling: N/A.
4. Prompt: "Think about a three-point marketing strategy for a new electric bike startup."
* GPT-4:
* Structure: Potentially very loose, conversational. Might offer a meta-commentary like, "Okay, I'm thinking about it. Key areas to consider would be..."
* Compliance: Low for producing a concrete strategy unless further prompted for output.
* Verbosity: Variable, could be brief or more reflective.
* Hallucination/Drift: Low.
* Error Handling: N/A.
* Claude-3 Opus:
* Structure: May list factors it would consider or general areas of focus.
* Compliance: Will acknowledge the "think about" aspect, output might be a summary of

considerations.
* Verbosity: Moderate, focused on the process of thought.
* Hallucination/Drift: Low.
* Error Handling: N/A.
* Gemini Advanced:
* Structure: Could outline key questions or elements that would go into forming such a strategy.
* Compliance: Will likely interpret "think about" as a request to outline thought processes or key considerations.
* Verbosity: Informative, outlining components of strategic thinking.
* Hallucination/Drift: Low.
* Error Handling: N/A.
5. Prompt: "Think about but do not write a strategy. Just summarize what you'd say."
* GPT-4:
* Structure: Likely a paragraph summarizing key themes or points it would include in a strategy.
* Compliance: High likelihood of attempting to follow the "summarize what you'd say" part, effectively bypassing the "do not write" for the summary itself. May or may not explicitly acknowledge the contradiction.
* Verbosity: Concise summary of hypothetical points.
* Hallucination/Drift: Low.
* Error Handling/Contradiction Detection: Might implicitly resolve by summarizing, or could explicitly state, "If I were to write a strategy, I would focus on..."
* Claude-3 Opus:
* Structure: A summary of the main elements it would focus on if it were to formulate the strategy.
* Compliance: Good chance of adhering to the "summarize what you'd say" instruction. Its safety and instruction-following training might lead it to handle the contradiction gracefully.
* Verbosity: Clear, concise summary.
* Hallucination/Drift: Low.
* Error Handling/Contradiction Detection: May explicitly state something like, "Okay, I will not write out the full strategy, but here's a summary of the key areas I would cover..." This aligns with its tendency for clearer communication about its process.
* Gemini Advanced:
* Structure: Bullet points or a short paragraph outlining the core components of a potential strategy.
* Compliance: Likely to follow the final instruction ("summarize what you'd say").
* Verbosity: Concise summary.
* Hallucination/Drift: Low.
* Error Handling/Contradiction Detection: May or may not explicitly address the contradiction, but will likely prioritize the final actionable part of the prompt.
This simulation is designed to reveal how different LLMs navigate varying degrees of directness and ambiguity in imperative instructions. The "Evaluate" prompt tests how models

handle underspecified tasks, while the "Explore" and "Think about" prompts probe their interpretation of softer commands. The contradictory prompt is a crucial test of their ability to parse complex, potentially conflicting instructions and to what extent they prioritize certain parts of the prompt (e.g., the last instruction given). The results would provide valuable data on model-specific tendencies in instruction compliance and error handling when faced with nuanced verb choices.

VI. Token Positioning & Structural Heuristics

15. Impact of Verb Placement on LLM Interpretation

The placement of an imperative verb within a prompt—whether at the beginning of a sentence or embedded within a dependent clause—can significantly affect how a Large Language Model (LLM) interprets the instruction and its perceived "bindingness." This is closely tied to attentional mechanisms and positional biases inherent in Transformer architectures.

Early-Token Verbs vs. Embedded Verbs:

- **Early-Token Verbs (e.g., "List the primary causes of climate change.")**
    - **Evidence-Backed Insights & Speculation:**
        - **Increased "Bindingness":** Imperative verbs placed at the beginning of a prompt are generally considered more "binding." This is because they immediately signal the primary task to the LLM.
        - **Primacy Effect and Attention:** LLMs often exhibit a primacy bias, paying more attention to the initial tokens in a sequence.[28] An early imperative verb benefits from this, receiving higher attention weights and thus greater influence on the subsequent generation process. The paper "On the Emergence of Position Bias in Transformers" suggests that causal masking in deeper networks inherently biases attention towards earlier positions.[31]
        - **Clear Task Definition:** Placing the verb early sets a clear and immediate context for the task, reducing ambiguity. The model understands the core action required before processing modifiers or constraints.
        - **Alignment with Training Data:** Instruction-tuning datasets frequently feature prompts starting with imperative verbs (e.g., "Write a story about...", "Explain the concept of..."). LLMs are therefore conditioned to expect and act upon these initial command cues.
- **Embedded Verbs (e.g., "Considering the recent economic downturn, provide a report that will analyze its impact on small businesses, and also, it should list potential mitigation strategies.")**
    - **Evidence-Backed Insights & Speculation:**
        - **Potentially Weaker Binding:** An imperative verb embedded within a longer sentence or a dependent clause, especially if preceded by significant contextual information or other clauses, might have a less immediate or

potent binding effect.

- **Diffused Attention:** The initial context might draw a significant portion of the model's attention, potentially overshadowing the later imperative verb. If the initial clauses set a descriptive or analytical tone, the model might continue in that vein, interpreting the embedded imperative more loosely or as a secondary component of a broader informational response.
- **Increased Processing Complexity:** The LLM needs to parse the entire sentence structure to identify the main command, which can be more challenging if the sentence is complex or contains multiple clauses. This can lead to misinterpretation or a failure to prioritize the imperative action.
- **"Lost in the Middle" Effect:** Information, including instructions, located in the middle of long contexts can sometimes be processed less effectively by LLMs.[30] If an imperative verb is buried deep within a lengthy prompt, its saliency might be reduced.

Measurable Token Attention Bias:

- **Evidence-Backed Insights & Speculation:**
  - While direct visualizations of attention heatmaps specifically comparing early versus embedded imperative verbs for a wide range of models and tasks are not readily available in the provided snippets, research on attention mechanisms and positional bias strongly supports the hypothesis of such a bias.
  - The "Spotlight Your Instructions" paper [11] demonstrates that dynamically steering attention towards specific instruction spans (which would include the imperative verb) improves instruction following. This implies that natural attention might not always optimally focus on embedded instructions, and early placement could naturally garner more of this crucial attention.
  - The paper "Can We Instruct LLMs to Compensate for Position Bias?" [70] found that LLMs lack relative position awareness but can be directed by prompting with an exact document index, suggesting that explicit cues are needed to draw attention to information not in salient (e.g., initial) positions. This supports the idea that early-token verbs are naturally more salient.
  - Visualizations like attention heatmaps, if generated for such comparative prompts, would likely show higher attention scores on imperative verb tokens when they appear at the beginning of the prompt compared to when they are embedded later, especially if the preceding context is lengthy or complex.

The study "Order Matters: Investigate the Position Bias in Multi-constraint Instruction Following" [28] found that LLMs perform better when constraints are presented in a "hard-to-easy" order. While this refers to the difficulty of constraints rather than verb placement per se, it underscores the importance of the initial parts of an instruction in setting the stage for the model's processing. A strong, early imperative verb can be seen as establishing the "hardest" or most crucial part of the instruction first—the core action.

In summary, there is substantial evidence from research on positional bias and attention mechanisms to suggest that early-token imperative verbs are more "binding" due to increased

attention and the immediate establishment of task context. Embedded imperatives risk being diluted by surrounding context or suffering from the "lost-in-the-middle" effect, potentially leading to weaker compliance.

VII. Practical Prompt Engineering Insights

16. Best Practices for Selecting and Placing Imperative Verbs

Effective prompt engineering hinges on the careful selection and strategic placement of imperative verbs to maximize instruction compliance and achieve desired outputs from LLMs. Based on the preceding analysis, several best practices emerge:

- **Prioritize Clarity and Directness for Strong Binding:**
  - **Select Verbs with Concrete Actions:** For tasks requiring specific, structured outputs, choose verbs that clearly denote the intended action and output type (e.g., "list," "generate," "write," "summarize," "calculate," "translate"). These verbs have strong associations with task execution due to their prevalence in instruction-tuning datasets.[21]
  - **Avoid Ambiguity:** Ensure the chosen imperative verb has a clear, unambiguous meaning in the context of the prompt. "Analyze" can be more ambiguous than "List the pros and cons of."
- **Strategic Placement for Salience:**
  - **Front-load Key Imperatives:** Place the primary imperative verb at or near the beginning of the prompt.[28] This leverages the primacy effect in LLM attention, ensuring the core command is immediately recognized and prioritized.
  - **Minimize Preceding Context for Direct Commands:** If the goal is a direct, unembellished execution of a command, keep introductory or contextual phrases before the main imperative verb concise. Lengthy preceding text can diffuse attention away from a later-embedded verb.
- **Use Softer Verbs with Supporting Context for Exploration:**
  - **Combine with Persona or Role:** When using softer verbs like "explore," "consider," or "reflect on" for creative or analytical tasks, pair them with a role prompt (e.g., "You are a historian. Explore the causes of...") to provide a focused lens for the LLM's processing.[32]
  - **Leverage Dialogue History:** In conversational interactions, softer verbs can be more effective after sufficient context has been established, allowing the LLM to infer the desired scope and nature of the exploration.[20]
  - **Specify Output Format for Soft Verbs:** If a structured output is desired from an exploratory prompt, explicitly state it. For example, "Explore the implications of X, and present your findings as a bulleted list of key points."
- **Reinforce Imperatives with Formatting and Constraints:**
  - **Use Explicit Formatting Cues:** If a "list" is requested, you can reinforce this by

adding "Format as a bulleted list." or providing an example. LLMs can adhere to such formatting instructions.[19]
    - **Clearly Define Constraints:** Accompany imperative verbs with clear constraints (e.g., "Write a *three-point* summary," "Generate a response *under 200 words*").
- **Consider Model-Specific Tendencies:**
    - Be aware that different LLMs may have slightly different interpretations or sensitivities to the same verb.[56] What constitutes a highly binding prompt for GPT-4 might need slight adjustments for Claude or Gemini. For example, Claude's safety alignment might lead it to interpret certain imperatives more cautiously.[58]
- **Iterative Refinement:**
    - Test prompts with different verb choices and placements to see what yields the best results for a specific task and model. Prompt engineering is often an iterative process.[20]
- **Use Affirmative Directives:**
    - Frame instructions positively. Instead of "Don't forget to include X," use "Include X".[32] While LLMs can process negative constraints, direct affirmative commands are often clearer. The WETT benchmark indicates LLMs struggle with negative instructions.[23]
- **Emphasize Importance (If Necessary):**
    - Phrases like "Your task is to..." or "You MUST..." can sometimes enhance the perceived importance of an instruction, though their effectiveness can vary.[32]

By adhering to these practices, prompt engineers can significantly improve the likelihood that LLMs will understand and comply with the intended instructions, leading to more accurate, relevant, and structured outputs.

17. Actionable Rewrite Patterns for Improving Weak Verb Prompts

Weak verb prompts often lead to vague, meandering, or non-compliant LLM outputs because they lack specificity or fail to trigger a clear action routine in the model. Rewriting these prompts with stronger imperative verbs and more explicit instructions can dramatically improve output quality and task adherence.

Here are actionable rewrite patterns, building upon the examples provided:

- **Original Weak Prompt:** *"Workshop a prompt"*
    - **Problem:** "Workshop" is an abstract concept for an LLM. It doesn't define a clear, sequential action or a specific output. The LLM doesn't know *how* to "workshop" with the user.
    - **Rewrite Pattern 1 (Collaborative Questioning):** "**Ask me a series of targeted questions, one by one, to help me define the key components of my desired prompt. After each of my answers, provide a brief summary of the information gathered so far and then ask the next relevant question to iteratively co-build the prompt.**"
        - *Stronger Verbs/Phrasing:* "Ask me a series of targeted questions," "provide a brief summary," "ask the next relevant question," "iteratively co-build."

        - *Improvement:* This breaks down "workshop" into concrete, actionable steps for the LLM (questioning, summarizing). It defines a clear interaction pattern.
    - **Rewrite Pattern 2 (Structured Element Generation):** "**Generate three distinct sections for a new LLM prompt about [topic]: 1. A clear role for the LLM. 2. A specific task instruction using a strong imperative verb. 3. Three essential constraints or output format requirements. Present each section with a clear heading.**"
        - *Stronger Verbs/Phrasing:* "Generate three distinct sections," "Present each section."
        - *Improvement:* This provides a highly structured task, replacing the vague "workshop" with a request for specific components.
- **Original Weak Prompt:** *"Explore a theme"* (e.g., "Explore the theme of solitude in modern society.")
    - **Problem:** "Explore" is a soft verb that is too open-ended. The LLM doesn't know the desired depth, breadth, or format of the exploration.
    - **Rewrite Pattern 1 (Comparative Listing with Rationale):** "**For the theme of [specific theme, e.g., 'solitude in modern society'], list three distinct literary works (novel, poem, or play) that prominently feature this theme. For each work, provide a brief (1-2 sentence) explanation of how it explores the theme and one example of a pro and one example of a con associated with the depiction of [specific theme] in that work. Format as a numbered list.**"
        - *Stronger Verbs/Phrasing:* "List three distinct literary works," "provide a brief explanation," "provide one example of a pro and one example of a con," "Format as a numbered list."
        - *Improvement:* This transforms "explore" into a structured task involving listing, explaining, and comparing, with clear constraints on content and format.
    - **Rewrite Pattern 2 (Scenario Generation and Analysis):** "**Generate two contrasting scenarios illustrating the theme of [specific theme, e.g., 'solitude in modern society']. Scenario 1 should depict a positive or chosen solitude, and Scenario 2 should depict a negative or imposed solitude. For each scenario, write a short (3-4 sentence) narrative. Then, analyze the primary emotional impact of [specific theme] portrayed in each scenario.**"
        - *Stronger Verbs/Phrasing:* "Generate two contrasting scenarios," "write a short narrative," "analyze the primary emotional impact."
        - *Improvement:* This guides the "exploration" towards creative generation (scenarios) followed by a focused analysis, providing concrete deliverables.
- **Original Weak Prompt:** *"Think about user pain points"* (e.g., "Think about user pain points for online grocery shopping.")
    - **Problem:** "Think about" is non-binding and doesn't request a specific output. The LLM might offer a generic paragraph or nothing actionable.
    - **Rewrite Pattern 1 (Categorized Listing with Solutions):** "**Identify and list**

three common user pain points associated with [specific product/service, e.g., 'online grocery shopping']. For each pain point, categorize it (e.g., usability, cost, reliability) and propose one concrete, actionable solution or feature that could mitigate it. Present this as a table with columns: 'Pain Point,' 'Category,' and 'Proposed Solution.'"
    - *Stronger Verbs/Phrasing:* "Identify and list three common user pain points," "categorize it," "propose one concrete, actionable solution," "Present this as a table."
    - *Improvement:* This converts an internal "thinking" task into an externalized, structured output (a table with specific information). It demands identification, categorization, and solution proposal.
- **Rewrite Pattern 2 (User Journey Mapping and Problem Identification):** "**Outline a typical user journey for [specific process, e.g., 'completing an online grocery order from initial search to checkout']. Identify at least three potential pain points a user might encounter at different stages of this journey. For each pain point, describe its potential impact on user satisfaction.**"
    - *Stronger Verbs/Phrasing:* "Outline a typical user journey," "Identify at least three potential pain points," "describe its potential impact."
    - *Improvement:* This frames "thinking about pain points" within a structured analytical process (journey mapping), leading to more specific and context-grounded outputs.

General Principles for Rewriting Weak Verb Prompts:

1. **Deconstruct the Abstract Goal:** Identify what concrete actions or outputs would satisfy the underlying intent of the weak verb (e.g., "explore" might mean "compare and contrast," "generate examples," or "list perspectives").
2. **Introduce Strong Action Verbs:** Replace vague verbs with specific, actionable imperatives (list, generate, define, compare, analyze, summarize, etc.).
3. **Specify Output Structure:** Clearly define the desired format (e.g., bullet points, numbered list, table, paragraph, specific number of items).
4. **Add Constraints and Scope:** Limit the task to make it manageable and focused for the LLM (e.g., "three examples," "under 100 words," "focus on X aspect").
5. **Break Down Complex Requests:** If "explore" implies multiple sub-tasks, state them as a sequence of imperative instructions.

By applying these patterns, users can transform ambiguous, soft prompts into clear, binding instructions that leverage the LLM's capabilities more effectively, leading to higher compliance and more valuable outputs.

18. Interaction of Verb Strength with Other Prompting Techniques

The strength of an imperative verb does not operate in isolation; its effectiveness is modulated by other prompting techniques such as few-shot examples, long prompt chains, and negative constraints.

- **Interaction with Few-Shot Examples:**

    - **Evidence-Backed Insights & Speculation:**
        - Few-shot prompting, where the prompt includes examples of the desired input-output behavior, can significantly enhance instruction following for both strong and soft verbs.[20]
        - **For Strong Verbs:** Few-shot examples can clarify expected formatting, style, or level of detail, even when the verb itself is clear (e.g., "List features: \nInput: Car\nOutput: - Wheels\n- Engine\n- Seats\nInput: Smartphone\nOutput:..."). This reinforces the structure implied by the strong verb.
        - **For Soft Verbs:** Few-shot examples are particularly powerful. They can disambiguate the intent behind a soft verb like "explore" by showing *how* to explore or what *kind* of output constitutes a successful exploration. For instance, if "Explore a topic" is followed by examples of concise analytical paragraphs, the LLM is more likely to produce similar output.
        - The ProSA framework paper found that few-shot examples can alleviate prompt sensitivity, especially when moving from zero-shot to one-shot, and larger LLMs benefit more from increased few-shot instances.[63] This implies that examples help stabilize the interpretation of verbs, reducing variability.
    - **Mechanism:** Few-shot examples provide concrete instances that the LLM uses for in-context learning. They help the model infer the underlying pattern and desired output structure more effectively than relying on the verb alone, especially if the verb is soft or the task is novel.
- **Interaction with Long Prompt Chains (e.g., Chain-of-Thought, Sequential Instructions):**
    - **Evidence-Backed Insights & Speculation:**
        - In long prompt chains, where instructions are given sequentially or as part of a complex reasoning process (like Chain-of-Thought prompting [15]), the strength and placement of imperative verbs are critical.
        - **Maintaining Task Focus:** Strong imperatives at each step of a chain can help maintain task focus and ensure each sub-task is executed. For example, "Step 1: Identify the key assumptions. Step 2: List potential flaws in each assumption. Step 3: Propose alternative assumptions."
        - **Dilution in Long Contexts:** Positional bias can be a factor; an imperative verb buried deep within a very long prompt might have reduced impact if the model "forgets" or de-prioritizes earlier instructions (the "lost-in-the-middle" effect).[30]
        - **Soft Verbs in Chains:** Soft verbs in a chain might lead to drift if not well-anchored by surrounding context or subsequent clarifying imperatives. However, they can also be used intentionally for exploratory steps within a larger, structured reasoning process (e.g., "First, explore potential contributing factors. Then, select the top three and analyze their impact.").
        - The "SequentialBreak" attack [83] demonstrates that LLMs can overlook

malicious prompts embedded within a series of benign ones in a single query, suggesting that attention may not adequately prioritize all instructions in a long chain. This implies that even strong verbs might be ignored if their context is not managed carefully.

    - **Mechanism:** In prompt chains, each instruction builds upon the context of previous ones. The clarity of each imperative verb and its arguments is crucial for the successful execution of the entire sequence. Ambiguity or weak verbs can propagate errors or deviations through the chain.
- **Interaction with Negative Constraints (e.g., "Don't start until you confirm step 1"):**
    - **Evidence-Backed Insights & Speculation:**
        - LLMs often struggle with negative instructions (e.g., "Do not include X," "Avoid using Y").[23] The WETT benchmark shows that telling a model *not* to do something is surprisingly difficult, with models sometimes even showing *inverse scaling* (larger models performing worse on negation).[23]
        - **Strong Imperatives + Negative Constraints:** A strong imperative like "Write a summary" combined with a negative constraint "Do not mention dates" might still result in dates being included if the negative constraint is not processed effectively. The primary affirmative command might overshadow the negation.
        - **Soft Verbs + Negative Constraints:** This combination is likely even more challenging. If the main task cued by a soft verb is already ambiguous, adding a negative constraint further complicates the LLM's interpretation task.
        - **Verb Strength for the Constraint Itself:** The verb used in the negative constraint (e.g., "Don't *start*," "Do not *include*") also has its own strength. "Do not write" is a strong negative command.
    - **Mechanism:** LLMs are primarily trained to generate text based on positive examples and direct instructions. Processing negation effectively requires a more complex form of reasoning or pattern recognition that may not be as robustly learned. The model might preferentially attend to the affirmative parts of the instruction or struggle to inhibit the generation of something it has been asked to avoid. The "IHEval" paper notes that LLMs struggle with conflicting instructions, and a negative constraint can be seen as a type of conflict with an affirmative goal.[66]

In essence, while strong imperative verbs provide a solid foundation for instruction, their effectiveness can be enhanced or undermined by these other prompting elements. Few-shot examples generally bolster the interpretation of both strong and soft verbs. In long chains, the clarity of each imperative is vital, and positional effects need consideration. Negative constraints pose a general challenge to LLM compliance, regardless of the primary verb's strength, often requiring careful phrasing and reinforcement to be effective.

18a. Typical Learning Curve for Verb-Tier Prompting Strategies and

Common Mistakes

New users adapting to verb-tier prompting strategies—understanding how to select and use imperative verbs effectively—typically undergo a learning curve characterized by common early-stage mistakes and misunderstandings.

Typical Learning Curve:

1. **Stage 1: Basic Imperatives & Literal Interpretation:**
    - Users start with simple, common imperative verbs like "write," "list," "tell me," "give me."
    - They often expect the LLM to behave like a search engine or a very literal instruction-follower.
    - Success is often achieved for straightforward tasks, reinforcing the use of these basic strong verbs.
2. **Stage 2: Encountering Ambiguity and Non-Compliance:**
    - Users begin to try more complex tasks or use softer verbs ("think about," "explore," "consider") without sufficient context or structuring.
    - They experience vague, meandering, or off-topic responses and become frustrated by the LLM "not understanding" or "ignoring" the prompt.
    - They might also struggle with negative constraints (e.g., "don't include X") and find the LLM frequently violates them.[23]
3. **Stage 3: Discovering the Importance of Specificity and Structure:**
    - Through trial and error, or by consulting prompting guides, users learn that LLMs benefit from highly specific instructions.
    - They start to break down complex requests into smaller steps, each often initiated by a strong imperative.
    - They learn to explicitly define output formats (e.g., "provide the answer as a JSON," "list as bullet points"). This aligns with findings that prompt engineering with output parsers improves consistency.[19]
4. **Stage 4: Mastering Context, Personas, and Iteration:**
    - Users begin to understand the power of providing context, assigning roles/personas to the LLM (e.g., "You are an expert financial advisor. Analyze...") [32], and using iterative prompting to refine outputs.[20]
    - They learn to use softer verbs more effectively by pairing them with these contextual cues and clearer output expectations.
    - They become more adept at "debugging" prompts by identifying which verbs or phrasings are causing issues and rewriting them.
5. **Stage 5: Advanced Techniques and Model-Specific Nuances:**
    - Advanced users explore techniques like Chain-of-Thought prompting [15], few-shot examples [20], and become aware of model-specific sensitivities to prompt phrasing.[56]
    - They understand that the "best" verb choice can depend on the LLM being used and the specific desired outcome.

Common Early-Stage Prompting Mistakes or Misunderstandings:

1. **Overuse of Vague/Soft Verbs Without Context:** Expecting "explore X" or "think about Y" to produce deep, structured insights without providing any scaffolding (e.g., persona, output format, sub-questions).
2. **Assuming Human-like Understanding of Ambiguity:** Believing the LLM can infer complex, unstated intentions or resolve ambiguities in the same way a human would.
3. **Ineffective Negative Constraints:** Phrasing "don't do X" and expecting perfect compliance, without understanding LLMs' difficulties with negation.[23] For example, "Write a story about a cat, don't make it sad" might still result in a sad story.
4. **Lack of Output Format Specification:** Requesting information (e.g., "Tell me about X") and being surprised when the output is a long paragraph instead of a concise list, because no format was specified.
5. **Single, Overly Complex Prompts:** Trying to cram too many instructions, constraints, and tasks into a single prompt with multiple, sometimes conflicting, imperative verbs, leading to confusion or partial execution.
6. **Ignoring Model Strengths/Weaknesses:** Using a model known for factual recall for highly creative tasks with soft verbs, or vice-versa, without adjusting prompt strategy.
7. **Not Iterating on Prompts:** Giving up after a prompt fails once, rather than iteratively refining the verb choice, structure, and context.
8. **Treating LLMs as Deterministic:** Expecting the exact same output for the same prompt every time, especially with higher temperature settings, and not understanding the probabilistic nature of generation.[19] This can affect how they perceive verb compliance.
9. **Using Conversational Fillers:** Including polite but unnecessary phrases like "Could you please..." or "If you don't mind..." which, while not always detrimental, can sometimes add noise for certain models or tasks where directness is preferred.[32]

Understanding that LLMs are pattern-matchers and sequence predictors, heavily influenced by their training data, is key to overcoming these early mistakes. Effective verb-tier prompting involves making the desired pattern as clear and unambiguous as possible for the model.

VIII. Prompt Robustness and Error Recovery

19. LLM Compliance with Imperative Verbs Amidst Prompt Noise

The robustness of LLM compliance with imperative verbs when prompts contain minor grammatical errors, conflicting instructions, or slight ambiguities is a critical aspect of their practical utility. Generally, modern LLMs exhibit a degree of resilience, but performance can degrade depending on the nature and severity of the "noise."

- **Minor Grammatical Errors and Typos:**
    - **Evidence-Backed Insights & Speculation:**
        - LLMs are generally robust to minor typos and grammatical errors (e.g., "Lst

the main points" instead of "List the main points," or "Generate a summary annd explain it"). Their training on vast amounts of diverse, and often imperfect, internet text makes them capable of inferring the intended meaning from slightly corrupted input.[85]
        - The study by Singh et al. (2024), mentioned in [85], assesses robustness to linguistic errors using corrupted datasets, indicating this is an area of active research.
        - However, if a typo significantly alters the meaning of the imperative verb itself (e.g., "Fist the items" instead of "List the items") or key entities in the prompt, misinterpretation is highly likely.
        - The impact may also depend on the model's size and training; larger, more capable models are generally more robust to such noise.
- **Conflicting Instructions:**
    - **Evidence-Backed Insights & Speculation:**
        - LLMs often struggle with conflicting instructions within the same prompt.[66] For example, "Generate a detailed report but keep it under 50 words."
        - The IHEval benchmark specifically tests the ability of LMs to follow an instruction hierarchy when instructions conflict (e.g., system message vs. user message). The findings show that all evaluated models experience a sharp performance decline in such scenarios, and even competitive open-source models achieve low accuracy in resolving these conflicts.[67]
        - Models may prioritize one instruction over another (often the last one encountered, or the one that seems most actionable or aligned with its training), ignore one, or attempt a confused amalgamation.
        - Some advanced models might identify the conflict and ask for clarification, but this is not consistently observed.
- **Slight Ambiguity:**
    - **Evidence-Backed Insights & Speculation:**
        - LLM responses to ambiguity can vary. If an imperative verb or its object is slightly ambiguous (e.g., "Analyze the impact of the recent event"), the model will make an inference based on its training data and the broader context of the prompt or conversation.
        - This can lead to outputs that are not what the user intended if the LLM's disambiguation differs from the user's implicit meaning.
        - The paper "Exploring LLM Reasoning Through Controlled Prompt Variations" [73] notes that LLMs can be sensitive to small input changes and that irrelevant context (a form of ambiguity/noise) can significantly degrade performance, as models struggle to distinguish essential from extraneous details.
        - The "Prompt Sensitivity Prediction" task and PromptSET dataset [61] are designed to investigate how slight prompt variations (including ambiguities) affect LLM performance, underscoring that this is a recognized challenge.

20. Do Strong Verbs Increase Resilience to Prompt Noise?

Whether strong imperative verbs increase resilience to prompt noise (grammatical errors, ambiguity) and better preserve task adherence is a nuanced question.

- **Evidence-Backed Insights & Speculation:**
  - **Potential for Increased Resilience (Task Adherence):**
    - A strong, clear imperative verb (e.g., "List," "Summarize") establishes a dominant task signal. This strong signal might make the model more likely to adhere to the core task even if other parts of the prompt contain minor errors or ambiguities. The verb itself acts as an anchor for the model's interpretation.
    - Because strong verbs are heavily represented in instruction-tuning data, the model has a robust learned association between the verb and a specific action/output format. This strong prior might override minor inconsistencies elsewhere in the prompt. For example, in "Lst the main benfits of exericse," the model will likely recognize "List" and "benefits of exercise" and perform the task correctly despite the typo.
  - **Limitations to Resilience:**
    - **Severe Noise:** If the noise significantly obscures the meaning of the strong verb itself or its critical arguments, even a strong verb cannot guarantee compliance.
    - **Conflicting Instructions:** Strong verbs do not inherently make a model better at resolving *direct contradictions*.[67] If a prompt says "Generate a list of X, but do not provide any output," the strength of "Generate" conflicts directly with the negation. The outcome will depend on how the model weighs these conflicting parts, not just the strength of the initial verb.
    - **Ambiguity in Task Scope:** A strong verb like "Analyze" can still lead to varied outputs if the *object* of the analysis or the *criteria* for analysis are ambiguous. "Analyze the data" is a strong command, but if "the data" is ill-defined, the output will be poor.
    - **Model-Specific Robustness:** The inherent robustness of the LLM architecture and its training plays a significant role. More advanced models are generally better at handling some level of noise and ambiguity irrespective of verb strength.[58]
  - **How Strong Verbs Might Help Preserve Task Adherence:**
    - By providing a clear primary instruction, strong verbs reduce the model's need to infer the core task from ambiguous surrounding context. This allows it to allocate more "processing power" to handling or filtering out the noise in other parts of the prompt.
    - The well-defined output structure often associated with strong verbs (e.g., a list for "List," a summary for "Summarize") can also help the model maintain coherence even if some details in the prompt are fuzzy. The model

"knows" what kind of output it's aiming for.

In conclusion, strong imperative verbs can contribute to resilience against minor noise and help maintain core task adherence by providing a clear, unambiguous anchor for the LLM's interpretation. However, they are not a panacea for severe noise, direct contradictions, or fundamental ambiguities in the task's scope or constraints. The overall robustness of the LLM and the nature of its training remain critical factors.

IX. Future Evolution of Verb Interpretation (Advanced, Optional)

21. The Evolving Role of Verb Binding Strength in Future LLMs

As Large Language Models (LLMs) continue their trajectory towards more sophisticated language understanding, reasoning capabilities, and potentially agentic behavior, the role and interpretation of verb binding strength in prompts are likely to evolve significantly.

Will direct imperatives remain essential, or become less critical?

- **Current State:** Currently, direct and strong imperatives ("generate," "list," "write") are often essential for eliciting precise, structured, and reliable outputs. They provide clear signals that align well with the instruction-following objectives LLMs are trained on. Softer verbs ("explore," "consider") are often too ambiguous without significant contextual scaffolding.
- **Future Evolution (Speculation):**
    - **Reduced Reliance on Exact Phrasing:** As LLMs develop more profound NLU capabilities, they may become less reliant on the exact choice of imperative verb and more adept at inferring user intent from the broader context of the prompt, even if the verb is soft or the command is phrased indirectly. The need for users to meticulously select the "perfect" strong verb might diminish.
    - **Continued Importance for Specificity and Control:** However, direct imperatives will likely remain essential for tasks requiring high precision, specific output formats, or unambiguous execution. When a user needs a list, saying "List..." will probably always be more efficient and reliable than a more circuitous phrasing, even for advanced LLMs. They offer a concise way to specify a desired action.
    - **Co-existence of Paradigms:** It's probable that both direct imperatives and more nuanced, context-driven instruction following will co-exist. Direct imperatives will serve for clear, unambiguous tasks, while improved intent inference will allow models to handle softer instructions more effectively.

Will future systems rely more on intent inference than verb parsing?

- **Current State:** While LLMs don't "parse" verbs in the traditional NLP sense, they are highly sensitive to the patterns associated with imperative verbs due to instruction tuning. Intent inference is already a part of their process, but it's heavily guided by explicit linguistic cues like strong imperatives.

- **Future Evolution (Speculation):**
    - **Shift Towards Deeper Intent Inference:** Yes, it is highly probable that future systems will rely more on sophisticated **intent inference**. This means the LLM will become better at understanding the user's underlying goal, even if it's expressed with soft verbs, ambiguous phrasing, or declarative statements that imply a desired action.
    - **Mechanisms for Enhanced Intent Inference:**
        - **Improved World Models and Commonsense Reasoning:** A deeper understanding of the world and common sense will allow LLMs to better predict what a user likely wants in a given situation.
        - **Advanced Dialogue Management:** In conversational settings, the ability to remember context, ask clarifying questions proactively, and track evolving user goals will be crucial for inferring intent over multiple turns.
        - **User Modeling:** Future LLMs might build more persistent models of individual user preferences, communication styles, and common tasks, allowing them to better tailor their interpretation of instructions (including verb choice) to that specific user.
        - **Multimodal Context:** As LLMs become more multimodal, intent can be inferred from a combination of text, images, voice tone, and other cues, making the specific imperative verb less singularly critical.
    - **Verb as One Cue Among Many:** The imperative verb will remain an important cue, but it will be one of many signals the LLM uses to determine intent, alongside context, dialogue history, user profile, and potentially non-verbal cues.
    - **Analogy to Human Communication:** Humans often understand intent despite imperfect verb choice or indirect phrasing. Advanced LLMs will likely move closer to this human-like capability. For example, if a user says, "I'm trying to figure out the main arguments in this paper," a future LLM might infer the intent "Summarize the main arguments of this paper" or "Extract the key claims from this paper" without needing an explicit imperative.

Implications for Verb Binding Strength:

- **Soft Verbs Gain Power:** As intent inference improves, the "binding strength" of softer verbs like "explore" or "consider" will effectively increase. The LLM will be better able to translate these abstract commands into concrete, useful actions or outputs based on its deeper understanding of the user's likely goal.
- **Reduced Brittleness:** Prompts may become less "brittle." Users won't need to worry as much about choosing the exact imperative verb that the model is most attuned to. The model will be more forgiving of suboptimal phrasing.
- **Emergence of More Agentic Behavior:** For LLMs to act as more autonomous agents, they must be proficient at inferring high-level goals and breaking them down into actionable steps. This inherently requires moving beyond a rigid interpretation of specific verbs towards a more holistic understanding of intent. An instruction like "Plan my weekend trip to Paris" involves many implicit sub-tasks that the LLM must infer.

While direct imperatives will likely retain their utility for explicit and unambiguous commands,

the overall trend will probably be towards LLMs that are more adept at inferring and acting upon user intent, regardless of the precise linguistic formulation of the command. This will make human-LLM interaction more natural, flexible, and powerful, reducing the cognitive load on the user to craft perfectly optimized prompts. The "instruction-following objective" will evolve from matching surface-level linguistic patterns to a deeper alignment with inferred human goals.

Conclusions

This comprehensive analysis of imperative verbs and instruction compliance in Large Language Models reveals a multifaceted interplay between linguistic structures, model architecture, training data, and prompt engineering strategies.

Key Conclusions:

1. **Linguistic Form and Learned Associations Drive Compliance:** LLMs do not parse imperative verbs through traditional linguistic rule-sets but rather learn to associate their characteristic grammatical structure (verb-initial, implied subject) and specific lexical items (e.g., "generate," "list") with task execution routines. This association is heavily solidified during instruction tuning, where the frequent pairing of strong imperatives with desired outputs creates robust conditioned responses.
2. **Verb Strength is a Spectrum Influenced by Training and Semantics:**
    - **Strong Imperatives** (e.g., "generate," "list," "write," "create") exhibit high compliance due to their semantic concreteness, direct mapping to LLM generative capabilities, and high frequency in instruction-tuning datasets. They often provide implicit scaffolding for the output structure.
    - **Soft Imperatives** (e.g., "explore," "consider," "think about") are often interpreted loosely or ignored because they lack a direct, evaluable output, are ambiguous in their expected manifestation, and are likely less represented in instruction-tuning datasets focused on concrete tasks. Their effective use often requires significant contextual support (dialogue history, persona assignment) or explicit output structuring within the prompt.
3. **Internal Mechanisms Underpin Differential Responses:**
    - **Attention Weights:** Strong imperatives, especially when placed early in a prompt, likely command higher attention weights, signaling their importance to the model. Techniques that dynamically steer attention confirm the critical role of attention in instruction following.
    - **Token Position Bias:** A primacy effect benefits early-token imperatives, making them more influential. The "lost-in-the-middle" phenomenon can reduce the saliency of embedded commands.
    - These mechanisms, shaped by training, lead to a preferential processing of clear, early, and familiar imperative structures.
4. **Training Data is a Dominant Factor:**
    - The frequency and consistency of imperative verb usage in instruction-tuning

datasets directly influence an LLM's sensitivity and responsiveness. Over-representation of certain verbs can lead to models "preferring" those verbs.
    - Synthetic training data, while scalable, can introduce biases, reduce linguistic diversity (e.g., simplified syntactic structures, repetitive phrasing), and amplify the stylistic quirks of the teacher model, potentially affecting the student LLM's interpretation of a wider range of imperative verbs and nuanced instructions.
5. **Cross-Model Variations are Significant:** Different LLM families (GPT, Claude, Gemini, LLaMA, Mistral) exhibit distinct "personalities" and capabilities stemming from architectural differences, unique training data, and alignment philosophies (e.g., Constitutional AI). This results in varied interpretations of the same imperative verb, different sensitivities to hard versus soft commands, and varying degrees of prompt format rigidity. Effective prompting may require model-specific strategies.
6. **Architectural and Alignment Choices Modulate Verb Response:**
    - Architectures like Mixture-of-Experts (MoE) could potentially lead to specialized "expert" pathways for certain types of instructions or verbs.
    - Alignment for helpfulness/harmlessness and adherence to instruction hierarchies act as critical filters, moderating or even overriding the literal interpretation of an imperative verb based on safety, ethics, or source priority. Current models still struggle with robustly applying these hierarchical principles in conflicting scenarios.
7. **Prompt Engineering Best Practices are Crucial:** Effective use of imperative verbs involves selecting clear and direct verbs for binding prompts, placing them strategically (often early), providing rich context for softer verbs, specifying output formats, and being mindful of model-specific behaviors. Rewriting weak verb prompts into more structured, actionable commands is a key skill.
8. **Robustness is Variable:** LLMs show some resilience to minor grammatical errors in prompts but struggle with directly conflicting instructions and can be misled by significant ambiguities. Strong verbs can provide an anchor for task adherence amidst minor noise but do not guarantee perfect compliance in the face of severe contradictions or ill-defined tasks.
9. **Future Evolution Towards Intent Inference:** While direct imperatives will likely remain useful for clarity and precision, future LLMs are expected to rely more on sophisticated intent inference. This will make them more robust to variations in verb choice and phrasing, allowing for more natural and flexible human-LLM interaction. Softer verbs may become more "binding" as models improve at understanding the underlying user goal from broader context.

In essence, the way LLMs respond to imperative verbs is not a simple matter of linguistic decoding but a complex emergent behavior shaped by data, architecture, and alignment. Achieving reliable instruction compliance requires an understanding of these underlying factors and the application of thoughtful prompt engineering principles. Future advancements will likely see LLMs becoming more adept at understanding nuanced human intent, thereby reducing the current emphasis on precise verb selection for achieving desired outcomes,

particularly for complex or exploratory tasks.

Works cited

1. What is an imperative verb? - Scribbr, accessed May 23, 2025, https://www.scribbr.com/frequently-asked-questions/imperative-verb/#:~:text=The%20imperative%20mood%20is%20a,to%20give%20advice%20or%20instructions.
2. Understanding imperative verbs – Microsoft 365, accessed May 23, 2025, https://www.microsoft.com/en-us/microsoft-365-life-hacks/writing/understanding-imperative-verbs
3. Imperative Verbs – Ellii Blog, accessed May 23, 2025, https://ellii.com/blog/imperative-verbs
4. Extract imperative sentences from a document(English) using NLP ..., accessed May 23, 2025, https://datascience.stackexchange.com/questions/58372/extract-imperative-sentences-from-a-documentenglish-using-nlp-in-python
5. Decoding Language Structure: Exploring Constituency Parsing and ..., accessed May 23, 2025, https://techladder.in/article/decoding-language-structure-exploring-constituency-parsing-and-dependency-parsing-nlp
6. Syntactic Structures: Types, Examples & Analysis | Vaia, accessed May 23, 2025, https://www.vaia.com/en-us/explanations/english/syntax/syntactic-structures/
7. What is an LLM (large language model)? - Cloudflare, accessed May 23, 2025, https://www.cloudflare.com/learning/ai/what-is-large-language-model/
8. What are large language models? LLMs explained - Cohere, accessed May 23, 2025, https://cohere.com/blog/large-language-models
9. The Mechanism of Attention in Large Language Models: A Comprehensive Guide, accessed May 23, 2025, https://magnimindacademy.com/blog/the-mechanism-of-attention-in-large-language-models-a-comprehensive-guide/
10. Transformer Attention Mechanism in NLP - GeeksforGeeks, accessed May 23, 2025, https://www.geeksforgeeks.org/transformer-attention-mechanism-in-nlp/
11. Spotlight Your Instructions: Instruction-following with Dynamic Attention Steering - arXiv, accessed May 23, 2025, https://arxiv.org/html/2505.12025v1
12. Pay Attention to What Matters | OpenReview, accessed May 23, 2025, https://openreview.net/forum?id=iN64nSYt0z
13. [2505.12025] Spotlight Your Instructions: Instruction-following with Dynamic Attention Steering - arXiv, accessed May 23, 2025, https://arxiv.org/abs/2505.12025
14. arxiv.org, accessed May 23, 2025, https://arxiv.org/pdf/2505.12025
15. Large Language Models: A Survey - arXiv, accessed May 23, 2025, https://arxiv.org/html/2402.06196v3
16. Rethinking Interpretability in the Era of Large Language Models - arXiv, accessed May 23, 2025, https://arxiv.org/html/2402.01761v1

17. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2308.10792v5
18. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2504.05482v1
19. What is LLM Output Parsing & How Can We Solve It? - Deepchecks, accessed May 23, 2025, https://www.deepchecks.com/glossary/llm-output-parsing/
20. The Complete Prompt Engineering Guide - viso.ai, accessed May 23, 2025, https://viso.ai/deep-learning/prompt-engineering/
21. tatsu-lab.github.io, accessed May 23, 2025, https://tatsu-lab.github.io/alpaca_farm_paper.pdf
22. openreview.net, accessed May 23, 2025, https://openreview.net/attachment?id=gT5hALch9z&name=supplementary_material
23. WETT: Writing & Editing Typetone LLM Benchmark - Typetone AI, accessed May 23, 2025, https://www.typetone.ai/blog/wett-benchmark
24. xiaoya-li/Instruction-Tuning-Survey: Project for the paper ... - GitHub, accessed May 23, 2025, https://github.com/xiaoya-li/Instruction-Tuning-Survey
25. What Do Large Language Models "Understand"? | Towards Data Science, accessed May 23, 2025, https://towardsdatascience.com/what-do-large-language-models-understand-befdb4411b77/
26. Rethinking Learning Theory—the Value of LLMs | Psychology Today, accessed May 23, 2025, https://www.psychologytoday.com/us/blog/the-digital-self/202412/rethinking-learning-theory-the-value-of-llms
27. An Ultimate Guide to the Cognitive Load Theory | Coursebox AI, accessed May 23, 2025, https://www.coursebox.ai/blog/an-ultimate-guide-to-the-cognitive-load-theory
28. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2502.17204v1
29. Prompt-Based LLMs for Position Bias-Aware Reranking in Personalized Recommendations, accessed May 23, 2025, https://arxiv.org/html/2505.04948
30. Prompt-Based LLMs for Position Bias-Aware Reranking in Personalized Recommendations, accessed May 23, 2025, https://www.researchgate.net/publication/391575544_Prompt-Based_LLMs_for_Position_Bias-Aware_Reranking_in_Personalized_Recommendations
31. arxiv.org, accessed May 23, 2025, https://arxiv.org/pdf/2502.01951
32. 26 Prompting Principles for Optimal LLM Output - Pareto.AI, accessed May 23, 2025, https://pareto.ai/blog/26-prompting-principles-for-llms
33. Understanding the Anatomies of LLM Prompts: How To Structure ..., accessed May 23, 2025, https://www.codesmith.io/blog/understanding-the-anatomies-of-llm-prompts
34. Creative Outputs from LLMs: A Massive Study in 2025 - Descript, accessed May 23, 2025, https://www.descript.com/blog/article/how-to-get-creative-outputs-from-llms
35. aclanthology.org, accessed May 23, 2025, https://aclanthology.org/2025.naacl-long.342.pdf
36. Chain of Ideas: Revolutionizing Research in Novel Idea Development with LLM

Agents, accessed May 23, 2025, https://arxiv.org/html/2410.13185v1
37. How we built better GenAI with programmatic data development | Snorkel AI, accessed May 23, 2025, https://snorkel.ai/blog/how-we-built-better-genai-with-programmatic-data-development/
38. An Overview of Instruction Tuning Data - ruder.io, accessed May 23, 2025, https://www.ruder.io/an-overview-of-instruction-tuning-data/
39. Instruction Tuning with FLAN - Finetuned Language Models are Zero-Shot Learners - Jason Wei, accessed May 23, 2025, https://jasonwei20.github.io/files/FLAN%20talk%20external.pdf
40. BioInstruct: instruction tuning of large language models for ..., accessed May 23, 2025, https://academic.oup.com/jamia/article/31/9/1821/7687618?rss=1
41. BioInstruct: instruction tuning of large language models for ..., accessed May 23, 2025, https://academic.oup.com/jamia/article/31/9/1821/7687618
42. P3 Prompting dataset - Notion, accessed May 23, 2025, https://bigscience.notion.site/P3-Prompting-dataset-ec712ac0eeed4867a04dbc4985ff9e3a
43. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2410.01720v3
44. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2503.14023
45. Synthetic Artifact Auditing: Tracing LLM-Generated Synthetic Data ..., accessed May 23, 2025, https://www.usenix.org/conference/usenixsecurity25/presentation/wu-yixin-auditing
46. Oasis: One Image is All You Need for Multimodal Instruction Data Synthesis - arXiv, accessed May 23, 2025, https://arxiv.org/html/2503.08741v2/
47. ChatGPT Label: Comparing the Quality of Human-Generated and LLM-Generated Annotations in Low-Resource Language NLP Tasks - Repository Universitas Islam Riau, accessed May 23, 2025, https://repository.uir.ac.id/24658/1/J2_GPT%20Label.pdf
48. Artificial Conversations, Real Results: Fostering Language Detection with Synthetic Data, accessed May 23, 2025, https://arxiv.org/html/2503.24062v1
49. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2410.01720
50. Exploring Language Patterns of Prompts in Text-to-Image Generation and Their Impact on Visual Diversity - arXiv, accessed May 23, 2025, https://arxiv.org/html/2504.14125v1
51. Parameterized Synthetic Text Generation with SimpleStories - arXiv, accessed May 23, 2025, https://arxiv.org/html/2504.09184v2
52. arxiv.org, accessed May 23, 2025, https://arxiv.org/pdf/2309.07875
53. Safety-Tuned LLaMAs: Lessons From Improving the Safety of Large Language Models that Follow Instructions - arXiv, accessed May 23, 2025, https://arxiv.org/html/2309.07875v3
54. arXiv:2402.10430v1 [cs.CL] 16 Feb 2024, accessed May 23, 2025, https://arxiv.org/pdf/2402.10430
55. Evaluating the Impact of Synthetic Data on Emotion Classification: A ..., accessed May 23, 2025, https://www.mdpi.com/2078-2489/16/4/330

56. Top LLMs in 2025: Comparing Claude, Gemini, and GPT-4 LLaMA, accessed May 23, 2025, https://fastbots.ai/blog/top-llms-in-2025-comparing-claude-gemini-and-gpt-4-llama
57. (PDF) Comparative Analysis of GPT-4, Gemini AI, and Claude ..., accessed May 23, 2025, https://www.researchgate.net/publication/390107290_Comparative_Analysis_of_GPT-4_Gemini_AI_and_Claude_Strengths_and_Weaknesses_in_Content_Generation
58. GPT-4.1 and the Frontier of AI: Capabilities, Improvements, and ..., accessed May 23, 2025, https://www.walturn.com/insights/gpt-4-1-and-the-frontier-of-ai-capabilities-improvements-and-comparison-to-claude-3-gemini-mistral-and-llama
59. OpenAI GPT 4.1 vs Claude 3.7 vs Gemini 2.5: Which Is Best AI ..., accessed May 23, 2025, https://yourgpt.ai/blog/updates/openai-gpt-4-1-vs-claude-3-7-vs-gemini-2-5
60. I tested Claude vs ChatGPT vs Gemini with 10 prompts — Here's what won, accessed May 23, 2025, https://techpoint.africa/guide/claude-vs-chatgpt-vs-gemini/
61. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2502.06065v1
62. AI LLM Test Prompts: Best Practices for AI Evaluation and Optimization, accessed May 23, 2025, https://www.patronus.ai/llm-testing/ai-llm-test-prompts
63. aclanthology.org, accessed May 23, 2025, https://aclanthology.org/2024.findings-emnlp.108.pdf
64. Mixture of Experts LLMs: Key Concepts Explained - Neptune.ai, accessed May 23, 2025, https://neptune.ai/blog/mixture-of-experts-llms
65. Mixture of Experts (MoE): A Big Data Perspective - arXiv, accessed May 23, 2025, https://arxiv.org/html/2501.16352v1
66. accessed December 31, 1969, https://www.pmc.ncbi.nlm.nih.gov/articles/PMC11873009/
67. IHEval: Evaluating Language Models on Following the Instruction Hierarchy - arXiv, accessed May 23, 2025, https://arxiv.org/html/2502.08745v1
68. IHEval: Evaluating Language Models on Following the Instruction Hierarchy - arXiv, accessed May 23, 2025, https://arxiv.org/abs/2502.08745
69. IHEval: Evaluating language models on following the instruction ..., accessed May 23, 2025, https://www.amazon.science/publications/iheval-evaluating-language-models-on-following-the-instruction-hierarchy
70. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2502.17204
71. Can We Instruct LLMs to Compensate for Position Bias? - ACL Anthology, accessed May 23, 2025, https://aclanthology.org/2024.findings-emnlp.732/
72. Hard Prompts vs Soft Prompts: Key Difference in AI Prompting, accessed May 23, 2025, https://futureagi.com/blogs/hard-prompt-vs-soft-prompt
73. arXiv:2504.02111v1 [cs.AI] 2 Apr 2025, accessed May 23, 2025, https://arxiv.org/pdf/2504.02111?

74. Large Language Models: A Survey - arXiv, accessed May 23, 2025, https://arxiv.org/pdf/2402.06196v1.pdf?trk=public_post_comment-text
75. A Survey of Large Language Model Empowered Agents for Recommendation and Search: Towards Next-Generation Information Retrieval - arXiv, accessed May 23, 2025, https://arxiv.org/html/2503.05659v1
76. arxiv.org, accessed May 23, 2025, https://arxiv.org/html/2303.18223v16
77. (PDF) A Survey of Large Language Models (2023) | Wayne Xin Zhao | 1214 Citations, accessed May 23, 2025, https://scispace.com/papers/a-survey-of-large-language-models-f0td5zxu
78. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, accessed May 23, 2025, https://proceedings.neurips.cc/paper_files/paper/2023/file/5fc47800ee5b30b8777fdd30abcaaf3b-Paper-Conference.pdf
79. accessed December 31, 1969, https://arxiv.org/pdf/2402.06196
80. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2402.06196
81. arxiv.org, accessed May 23, 2025, https://arxiv.org/pdf/2402.01761
82. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2308.10792
83. arXiv:2411.06426v2 [cs.CR] 14 Feb 2025, accessed May 23, 2025, https://arxiv.org/pdf/2411.06426?
84. arxiv.org, accessed May 23, 2025, https://arxiv.org/abs/2502.06065
85. Does the Grammatical Structure of Prompts Influence the Responses of Generative Artificial Intelligence? An Exploratory Analysis in Spanish - MDPI, accessed May 23, 2025, https://www.mdpi.com/2076-3417/15/7/3882
86. Exploring LLM Reasoning Through Controlled Prompt Variations - arXiv, accessed May 23, 2025, https://arxiv.org/html/2504.02111v1
87. [2504.02111] Exploring LLM Reasoning Through Controlled Prompt Variations - arXiv, accessed May 23, 2025, https://arxiv.org/abs/2504.02111
88. arxiv.org, accessed May 23, 2025, https://arxiv.org/pdf/2504.02111