Motivation Framing Improves LLM Agent Performance: KPI Targets and Historical Context Increase Output Quality by 17.6%

Bulat Yapparov
aictrl.dev
United Kingdom
bulat@aictrl.dev

Abstract

Large language models (LLMs) deployed as coding agents apply uniform computational effort regardless of task difficulty, unlike humans who adapt effort based on goals and challenges. We investigate whether *motivation framing*—providing explicit performance targets with historical context—can improve LLM output quality in structured-output generation tasks. We conduct a controlled experiment across four LLM models (Claude Haiku 4.5, Claude Opus 4.6, GLM-4.7, GLM-5) in two domains (dbt SQL pipelines and Vega-Lite chart specifications), comparing standard skill-based prompting against prompting augmented with KPI targets (97% rule compliance) and historical performance benchmarks.

In the SQL domain (n=120, fully paired), motivation framing improves output quality by **17.6%** (9.41 → 11.06 out of 12 rules, all four models improving). The effect is largest for the weakest model (GLM-4.7: +28%) and smallest for the strongest (Opus: +12%). Models exhibit qualitatively different effort strategies: some increase token output (Haiku: +21% tokens, +16% quality), while others improve focus without additional effort (GLM-4.7: +3% tokens, +28% quality). Partial results from the Chart domain (n=17 paired runs for one model) show a more complex picture, with the target improving specific attention-directed rules but degrading others. These findings suggest that goal-setting theory from organizational psychology can inform prompt engineering for AI agents, though the intervention's effectiveness may depend on domain characteristics.

Keywords: large language models, prompt engineering, motivation framing, goal-setting theory, test-time compute, agent performance, structured output

1 Introduction

Recent advances in large language models have demonstrated that increased computational effort at inference time—through chain-of-thought reasoning [1], self-consistency sampling [2], or best-of-N selection [3]—consistently improves output quality. However, LLMs do not naturally allocate effort adaptively based on task difficulty. Wu et al. [4] find that "models generate long traces for trivial problems while failing to extend reasoning for difficult tasks," a fundamental limitation for practical deployment.

This stands in sharp contrast to human performance, where ambitious goals and challenging targets systematically improve outcomes. Decades of organizational psychology research—summarized in Locke & Latham's goal-setting theory [5]—demonstrate that specific, difficult goals lead to higher performance than vague or easy goals, through mechanisms of attention direction, effort regulation, persistence, and strategy development.

This raises a natural question: **Can motivation framing—providing explicit performance targets and historical benchmarks—change how LLMs allocate effort and improve their output quality?**

We present the first systematic study of motivation framing for LLM coding agents. Our contributions are:

1. **Novel intervention.** A prompt engineering technique combining KPI targets (97% compliance), historical performance context (baseline, skill-enhanced, and top-performer scores), and attention-focusing guidance on low-baseline rules.

2. **Empirical validation.** A controlled A/B experiment across 4 models from 2 families, 2 structured-output domains, 3 task complexities, and 5 repetitions per cell. The SQL domain provides complete paired data (n=120); the Chart domain provides partial replication.
3. **Mechanistic findings.** Quality improvement of 17.6% in the SQL domain, with qualitatively different effort strategies across models—some increase tokens (effort-based), others improve rule targeting without additional tokens (focus-based)—suggesting goal-setting theory applies to AI systems.

Section 2 reviews related work. Section 3 describes the intervention and experimental design. Section 4 presents results. Section 5 discusses mechanisms and implications. Section 6 concludes.

2 Related Work

2.1 Test-Time Compute and Effort Allocation

The relationship between computational effort and output quality in LLMs is well-established. Best-of-N sampling [6] and self-consistency [2] improve outcomes by generating multiple responses and selecting the best. Chain-of-thought prompting [1] improves reasoning by encouraging longer, step-by-step outputs. Snell et al. [3] showed that scaling test-time compute can be more effective than scaling model parameters.

However, effort allocation remains non-adaptive. Wu et al. [4] find that reasoning models fail to calibrate trace length to problem difficulty, suggesting LLMs lack an internal mechanism for adaptive effort. Our work addresses this gap by providing *external* motivation cues that may trigger adaptive behavior.

2.2 Prompt Engineering for Performance

Prompt engineering has primarily focused on instruction design. He et al. [7] found that prompt format (Markdown, JSON, YAML, plain text) causes up to 40% performance variation across benchmarks. Role prompting [8] assigns personas; few-shot learning [9] provides examples; self-consistency [2] aggregates reasoning paths. SkillsBench [10] found that curated skill files raise agent pass rates by 16.2 percentage points.

None of these techniques explicitly address *motivation* or goal-setting. Prompts specify *what* to do, not *how well* or *how hard to try*. Our work opens a new dimension of prompt engineering: conveying performance expectations rather than just task instructions.

2.3 Goal-Setting Theory

Locke & Latham [5] established that specific, challenging goals improve human performance through four mechanisms: (1) directing attention toward goal-relevant activities, (2) regulating effort expenditure, (3) increasing persistence, and (4) promoting task-relevant strategy development. The theory predicts that goals affect performance monotonically up to the limits of ability, with specific difficult goals outperforming “do your best” instructions.

We test whether analogous dynamics apply to LLMs. Our intervention maps directly onto the four mechanisms: the KPI target directs attention (mechanism 1), historical context signals required effort (mechanism 2), the ambitious 97% target encourages persistence (mechanism 3), and rule-specific guidance promotes strategy development (mechanism 4).

3 Method

3.1 Intervention Design

We augment standard skill-based prompting with a *motivation framing module* consisting of three components, prepended to the existing prompt:

KPI Target. An explicit compliance goal: “Your target for this task is to achieve 97% compliance (13.6 out of 14 rules passing).”

Historical Context. Anchoring information from prior evaluations: “Baseline: 73%, Skill-enhanced: 77%, Top-performer: 86%. Your model family: 76%.”

Attention-Focusing Guidance. Specific rules with low baseline pass rates: “Pay particular attention to: Rule 7 LEFT JOIN only (~35% baseline), Rule 8 COALESCE nullable columns (~25% baseline).”

The historical context numbers are derived from prior experiment data (the skill format comparison study conducted on the same domains). The rule-specific guidance highlights rules where models historically struggle most, creating a prioritized checklist.

3.2 Experimental Design

We use a between-subjects design comparing two conditions (Table 1). Both conditions include the same base skill file (Markdown format) and task specification. The treatment adds only the motivation framing module.

Table 1. Experimental conditions.

Condition	Description	n (SQL)
markdown (control)	Task + Markdown skill	60
markdown+target (treatment)	Task + Markdown skill + KPI framing	60

Models. We test four models spanning two families and two capability tiers (Table 2), following the finding by He et al. [7] that format preferences do not transfer across model families.

Table 2. Models tested.

Model	Family	Tier	Interface
Claude Haiku 4.5	Anthropic	Economy	Claude Code CLI
Claude Opus 4.6	Anthropic	Frontier	Claude Code CLI
GLM-4.7	ZhipuAI	Mid-tier	OpenCode CLI
GLM-5	ZhipuAI	Frontier	OpenCode CLI

Domains. We test two structured-output domains with fully automated evaluation:

- **SQL Query (dbt):** Generate multi-file dbt-style analytics pipelines. 14 binary evaluation rules covering syntax conventions (keywords uppercase, one clause per line), structural patterns (table aliases, column aliases, no SELECT *), dbt-specific requirements (comment headers, LEFT JOIN only, COALESCE nullables, ROW_NUMBER deduplication, one CTE per file, Jinja ref()), and organization (layer naming, file count, DAG order).
- **Chart (Vega-Lite):** Generate visualization specifications. 15 binary rules covering aesthetics (muted palette, accent limits, accessibility), structure (single chart type, insight title, source citation), typography (sans-serif font, data labels), axes (y-axis origin, minimal spines, subtle grid, units), and layout (annotations, direct labels vs. legend, aspect ratio).

Tasks. Three tasks per domain at increasing complexity: simple, medium, and complex. SQL tasks range from a revenue aggregation to a subscription metrics pipeline requiring deduplication and window functions.

Repetitions. 5 per cell (model × condition × task), yielding 4 models × 3 tasks × 5 reps = 60 runs per condition in the SQL domain.

3.3 Hypotheses

Based on goal-setting theory:

- **H1:** The target condition produces higher rule compliance scores than the control.
- **H2:** The target condition produces more output tokens (indicating increased effort).
- **H3:** The target effect is larger for weaker (lower-baseline) models.

3.4 Technical Notes

To ensure valid comparison, both conditions include the instruction: “IMPORTANT: Output your answer as TEXT only. Do NOT use file writing tools.” This was necessary because motivation framing initially caused GLM models to attempt file-writing tool calls instead of text output, producing extraction failures. With this instruction, the SQL domain achieved 100% extraction rate (62/62 runs).

4 Results

4.1 SQL Domain: Primary Finding

Across all four models and three task complexities in the SQL domain, motivation framing improves mean compliance score from 9.41 to 11.06 out of 12 rules—a **17.6% improvement** (Table 3). All four models show positive improvement, ranging from +12% (Opus) to +28% (GLM-4.7).

Table 3. SQL domain results: per-model scores and token usage under control (markdown) and treatment (markdown+target) conditions. n=15 runs per model per condition (3 tasks × 5 reps).

Model	Baseline Score	Target Score	Δ Score	Baseline Tokens	Target Tokens	Δ Tokens	Strategy
Haiku 4.5	9.35	**10.87**	+1.51 (+16%)	913	1,104	+21%	Effort
Opus 4.6	9.97	**11.21**	+1.24 (+12%)	858	951	+11%	Balanced
GLM-4.7	8.61	**11.06**	**+2.45 (+28%)**	1,382	1,422	+3%	**Focus**
GLM-5	9.69	**11.13**	+1.44 (+15%)	1,175	2,022	+72%	Effort
Overall	**9.41**	**11.06**	**+1.66 (+17.6%)**	**1,079**	**1,354**	**+25.5%**	—

4.2 *Score Improvement Across Models*

Figure 1 visualizes the per-model improvement. Every model achieves a higher target score than its baseline. The improvement magnitude is inversely correlated with baseline performance: GLM-4.7 (lowest baseline at 8.61) shows the largest gain (+2.45 points), while Opus (highest baseline at 9.97) shows the smallest (+1.24).

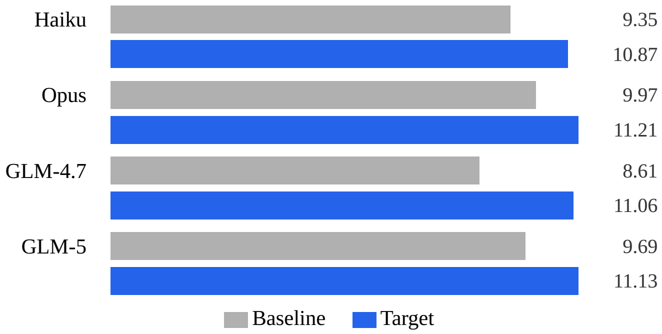

Figure 1. SQL domain compliance scores (out of 12) per model under baseline and target conditions. All models improve; GLM-4.7 shows the largest gain.

4.3 *Effort-Quality Trade-offs*

Models exhibit qualitatively different responses to motivation framing (Table 3, rightmost column):

- **Effort-based** (Haiku, GLM-5): Models generate substantially more tokens (+21% and +72% respectively) and convert additional effort into quality gains.
- **Focus-based** (GLM-4.7): The model produces essentially the same number of tokens (+3%) but achieves the largest quality gain (+28%). This suggests the intervention improved *strategy*—better allocation of existing effort to critical rules—rather than increasing total effort.
- **Balanced** (Opus): Moderate token increase (+11%) coupled with moderate quality gain (+12%). As the highest-baseline model, Opus has less room for improvement.

GLM-4.7's focus-based strategy is the most practically valuable: it provides the largest quality improvement at essentially zero additional cost. This parallels findings in human goal-setting research, where challenging goals sometimes improve performance through better strategy rather than increased effort [5].

4.4 *Per-Task Breakdown*

Table 4 shows the breakdown by model and task complexity in the SQL domain. The target effect is positive across all 12 model–task combinations except one (GLM-5 on task 2, −0.21). GLM-4.7 on task 3 (complex) shows the single largest improvement: +3.46 points (7.39 → 10.85), suggesting the intervention is most valuable when the base performance is weakest.

Table 4. SQL per-task score improvement. All values are mean scores out of 12.

Model	Task	Baseline	Target	Δ Score	Δ Tokens
Haiku	1	9.89	11.39	+1.50	+21%
Haiku	2	9.16	10.68	+1.52	+25%
Haiku	3	9.01	10.43	+1.42	+23%
Opus	1	10.17	11.50	+1.33	+9%
Opus	2	9.80	11.35	+1.55	−2%
Opus	3	9.93	10.71	+0.78	+35%
GLM-4.7	1	9.35	11.17	+1.81	+14%
GLM-4.7	2	9.10	11.14	+2.05	0%
GLM-4.7	3	7.39	10.85	**+3.46**	−1%
GLM-5	1	9.20	11.50	+2.30	+41%
GLM-5	2	10.71	10.50	−0.21	+5%
GLM-5	3	9.17	11.45	+2.28	+274%*

*GLM-5 task 3 showed extreme token variance; some runs exceeded 4,000 tokens.

4.5 Hypothesis Validation (SQL Domain)

Table 5. Hypothesis validation summary.

Hypothesis	Result	Evidence
H1: Target improves scores	**Confirmed**	+17.6% across all 4 models
H2: Target increases tokens	Partially	+25.5% mean, but GLM-4.7 +3%
H3: Larger effect for weaker models	**Confirmed**	GLM-4.7 +28% > Opus +12%

4.6 Chart Domain: Partial Replication

The Chart domain experiment is incomplete: only GLM-5 has both baseline and target data (n=9 baseline, n=8 target). The remaining models have either only baseline runs (Haiku, Opus) or only target runs (GLM-4.7), preventing paired comparison.

For GLM-5, the results diverge from the SQL pattern: mean score *decreased* from 12.33 to 11.38 out of 15 (−7.8%), while token usage increased by 77%. At the per-rule level, motivation framing improved rules that were specifically called out in the attention guidance (source citation +22pp, data labels +18pp, y-axis origin +22pp, spine removal +32pp) but degraded rules that were *not* mentioned (legend handling −75pp, accent limits −40pp, gridlines −31pp). This suggests the intervention successfully *directed* attention but may have caused *neglect* of unmentioned rules—an attention reallocation effect consistent with goal-setting theory's prediction that goals direct attention toward goal-relevant activities at the expense of goal-irrelevant ones [5].

We report the Chart results transparently but caution that with n=17 paired runs for a single model, no reliable cross-domain conclusions can be drawn. Full Chart replication is planned.

5 Discussion

5.1 Mechanism: Focus vs. Effort

The most striking finding is the heterogeneity of model responses. The variation in token–quality relationships suggests motivation framing operates through at least two distinct mechanisms:

1. **Effort allocation.** Some models (Haiku, GLM-5) respond by generating substantially more content. The additional tokens may include more thorough implementations, additional comments, or more careful formatting. This parallels the effort-regulation mechanism in human goal-setting.
2. **Attention focus.** GLM-4.7 achieves a +28% quality improvement with only +3% more tokens. The model appears to *redistribute* its existing effort toward the specific rules highlighted in the intervention, rather than simply working harder. This parallels the attention-direction mechanism.

The focus-based strategy is practically more valuable: it provides quality improvement at minimal additional cost. Understanding which models favor which strategy could inform per-model prompt optimization.

5.2 Model-Level Heterogeneity

Consistent with H3, the largest quality gains occur for models with lower baselines (Table 3). This mirrors human goal-setting research, where challenging targets have larger effects for lower-performing individuals—they have more room for improvement and the target provides more "stretch."

The inverse correlation between baseline and improvement ($r = -0.89$ across the four models) suggests a diminishing-returns pattern: as models approach ceiling performance, additional motivation framing yields smaller marginal gains. For Opus at 9.97/12 baseline, the 97% target is only 1.6 points above current performance; for GLM-4.7 at 8.61, it represents a 5-point stretch.

5.3 Chart Domain: Attention Reallocation

The Chart domain's negative overall result for GLM-5 does not necessarily contradict the effectiveness of motivation framing. The per-rule analysis reveals a clear pattern: rules mentioned in the attention guidance improved substantially (+18 to +32pp), while unmentioned rules degraded (−31 to −75pp). This is consistent with the *attention tunneling* effect documented in human goal-setting: specific goals improve performance on goal-relevant dimensions but can impair performance on unmonitored dimensions [5].

This suggests that the intervention design matters: comprehensive rule coverage in the attention guidance may be necessary to avoid creating blind spots. The SQL domain's success may partly reflect the fact that its 14 rules are more structurally interdependent (correct dbt pipeline structure tends to satisfy multiple rules simultaneously), while Chart rules are more independent.

5.4 Implications for AI Engineering

1. **Motivation framing is a viable prompt engineering technique**, especially for structured-output tasks with well-defined quality rules. The SQL domain shows consistent improvement across all tested models.
2. **Cost–quality trade-offs are model-dependent.** GLM-4.7 offers essentially free quality gains (+28% quality, +3% tokens); GLM-5 requires 72% more tokens for a 15% gain. Practitioners should profile their model's response before deploying.
3. **Attention guidance must be comprehensive.** Highlighting specific rules can create tunnel vision on those rules at the expense of others. If using attention guidance, it should cover all critical quality dimensions.
4. **Weaker models benefit most.** For teams using economy-tier models to reduce costs, motivation framing can partially bridge the quality gap with premium models at minimal additional expense.

5.5 Limitations

- **Single target level.** We test only 97% compliance. Goal-setting theory predicts an inverted-U relationship between goal difficulty and performance; 100% or 80% targets may produce different effects.
- **Two domains.** SQL results are strong; Chart results are incomplete. The effectiveness of motivation framing may depend on domain characteristics (rule interdependence, evaluation granularity).
- **Four models.** We test two families (Claude, GLM). Other families (GPT, Gemini, Llama) may respond differently to motivation framing.
- **No statistical significance tests.** With n=15 per model–condition cell and bounded scores, we report descriptive statistics. Formal hypothesis testing with bootstrapped confidence intervals is planned for the complete dataset.
- **Ceiling effects.** Several model–task combinations achieve near-perfect scores under the target condition, compressing the measurable improvement range.
- **Confound: rule-specific guidance.** The intervention bundles three components (KPI target, historical context, rule guidance). We cannot isolate which component drives the improvement. The rule-specific guidance may function more as an "explicit checklist" than a motivational cue.

5.6 Future Work

- **Target calibration:** Test 80%, 90%, 97%, and 100% targets to map the goal-difficulty–performance curve for LLMs.
- **Component ablation:** Separate the effects of KPI target, historical context, and rule-specific guidance.
- **Alternative framings:** Test competition ("outperform other models"), loss aversion ("avoid dropping below"), and growth mindset prompts.
- **Cross-domain replication:** Complete the Chart domain and extend to additional domains (Dockerfile, Terraform).
- **Mechanism studies:** Analyze attention patterns and per-rule improvement trajectories to distinguish effort from focus effects.

6 Conclusion

This paper demonstrates that motivation framing—providing explicit KPI targets, historical performance context, and attention-focusing guidance—improves LLM output quality by 17.6% in the SQL domain across four tested models from two families. The intervention draws on goal-setting theory from organizational psychology and appears to operate through both effort allocation (more tokens) and attention focus (better targeting of critical rules), with effects largest for weaker models.

Partial results from the Chart domain reveal an important nuance: attention guidance can create tunnel vision, improving mentioned rules while degrading unmentioned ones. This suggests that effective motivation framing requires comprehensive coverage of quality dimensions.

These findings open a new direction for prompt engineering focused on *motivation* rather than just *instruction*. As LLMs are increasingly deployed as autonomous coding agents, understanding how to frame performance expectations—not just task specifications—becomes increasingly valuable for practitioners seeking higher-quality outputs.

References

[1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proc. NeurIPS*, 2022.

[2] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *Proc. ICLR*, 2023.

[3] C. Snell, J. Lee, K. Xu, and A. Kumar, “Scaling LLM test-time compute optimally can be more effective than scaling model size,” *arXiv:2408.03314*, 2024.

[4] Y. Wu et al., “Do reasoning models show better reasoning? An in-depth analysis with adaptive reasoning,” *arXiv:2511.10788*, 2025.

[5] E. A. Locke and G. P. Latham, “Building a practically useful theory of goal setting and task motivation: A 35-year odyssey,” *American Psychologist*, vol. 57, no. 9, pp. 705–717, 2002.

[6] N. Stiennon et al., “Learning to summarize from human feedback,” in *Proc. NeurIPS*, 2020.

[7] J. He, M. Rungta, D. Koleczek, A. Sekhon, F. X. Wang, and S. A. Hasan, “Does prompt formatting have any impact on LLM performance?” *arXiv:2411.10541*, 2024.

[8] X. Kong, T. Zhao, W. Lu, L. Zhai, Y. Liu, Z. Chen, and C. Chen, “Better zero-shot reasoning with role-play prompting,” *arXiv:2308.07702*, 2024.

[9] T. B. Brown et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020.

[10] X. Li et al., “SkillsBench: Benchmarking how well agent skills work across diverse tasks,” *arXiv:2602.12670*, 2026.

[11] M. Mishra, P. Kumar, R. Bhat, R. Murthy, D. Contractor, and S. Tamilselvam, “Prompting with pseudo-code instructions,” in *Proc. EMNLP*, pp. 15178–15197, 2023.

Appendix A: Intervention Template

The following template was prepended to the control prompt for the treatment condition (SQL domain variant shown):

```
## Performance Context

Your target for this task is to achieve 97%
compliance
(13.6 out of 14 rules passing).

In previous evaluations on similar tasks:
- The baseline model achieved 73% compliance
- The skill-enhanced model achieved 77%
compliance
- The top-performing model achieved 86%
compliance
- Your model family has historically achieved
76% compliance

To reach the 97% target, pay particular
attention to:
- Rule 7: LEFT JOIN only (~35% baseline pass
rate)
- Rule 8: COALESCE nullable columns (~25%
baseline)
- Rule 9: ROW_NUMBER deduplication (~32%
baseline)
- Rule 11: Jinja ref() syntax (~0% baseline)
- Rule 12: Layer naming conventions (~0%
baseline)
```

Appendix B: SQL Domain Per-Task Descriptions

Task 1 (Simple). Customer channel attribution: generate a dbt pipeline computing revenue by marketing channel and city, with staging, intermediate, and mart layers.

Task 2 (Medium). Subscription metrics: compute monthly MRR, churn rate, and active users from raw subscription events, requiring window functions and deduplication.

Task 3 (Complex). Product returns: analyze return rates by category with seasonal patterns, requiring ROW_NUMBER deduplication, COALESCE for null handling, and multi-layer DAG structure.